

Міністерство освіти і науки України  
Карпатський національний університет  
імені Василя Стефаника

О. В. Махней

# МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ

Навчальний посібник для студентів  
закладів вищої освіти  
спеціальності «прикладна математика»

*Видання друге, перероблене і доповнене*

Івано-Франківськ  
2026

УДК 004.94:519.24

ББК 22.18

М36

*Рекомендовано Вченою радою факультету математики та інформатики Карпатського національного університету імені Василя Стефаника як навчальний посібник для студентів спеціальності «прикладна математика» (протокол № 1 від 20 січня 2026 р.).*

#### **Рецензенти:**

*Дмитришин Р. І.*, доктор фізико-математичних наук, професор (Карпатський національний університет імені Василя Стефаника);

*П'янило Я. Д.*, доктор технічних наук, професор (Інститут прикладних проблем механіки і математики ім. Я. С. Підстригача).

**М36 Махней О. В. Математичне моделювання** : навчальний посібник. Вид. 2-ге, перероб. і допов. Івано-Франківськ : Карпат. нац. ун-т ім. В. Стефаника, 2026. 383 с.

У посібнику викладено навчальний курс з математичного моделювання. Увага приділена як аналітичним, так і імітаційним методам моделювання. У книзі розглянуто застосування диференціальних рівнянь, клітинних автоматів, мови імітаційного моделювання GPSS World та інших методів моделювання різних систем.

Для студентів спеціальності «прикладна математика». Може бути корисним для студентів інших спеціальностей галузі знань «Інформаційні технології» і студентів фізико-математичних спеціальностей, аспірантів, науково-технічних працівників.

**УДК 004.94:519.24**

**ББК 22.18**

© Махней О. В., 2026.

## Зміст

<b>Передмова</b> . . . . .	<b>9</b>
<b>Розділ 1. Основні поняття про моделювання</b> . . . . .	<b>11</b>
§ 1.1. Поняття системи . . . . .	11
§ 1.2. Поняття моделі . . . . .	13
§ 1.3. Класифікація видів моделювання . . . . .	14
§ 1.4. Поняття математичного моделювання . . . . .	17
§ 1.5. Принципи математичного моделювання . . . . .	20
§ 1.6. Етапи математичного моделювання . . . . .	22
Питання до розділу 1 . . . . .	23
<b>Розділ 2. Диференціальні моделі</b> . . . . .	<b>24</b>
§ 2.1. Побудова диференціальних моделей . . . . .	24
§ 2.2. Падіння тіла . . . . .	26
§ 2.3. Вентиляція приміщення . . . . .	28
§ 2.4. Витікання рідини . . . . .	29
§ 2.5. Потік частинок у трубі . . . . .	31
§ 2.6. Взаємодія в системі «хижак-жертва» . . . . .	34
§ 2.7. Модель розвитку епідемії . . . . .	39
§ 2.8. Прогин балок . . . . .	44
Питання до розділу 2 . . . . .	49
Вправи до розділу 2 . . . . .	49
<b>Розділ 3. Скінченні автомати</b> . . . . .	<b>51</b>
§ 3.1. Скінченні автомати Мілі . . . . .	51
§ 3.2. Скінченні автомати Мура . . . . .	52
§ 3.3. Клітинні автомати . . . . .	54
§ 3.4. Застосування клітинних автоматів до моделювання активних середовищ . . . . .	57
§ 3.5. Модель розвитку організмів на поживних середовищах . . . . .	58
§ 3.6. Модель розвитку організмів на поживних середовищах за наявності хижаків . . . . .	62
Питання до розділу 3 . . . . .	67
Вправи до розділу 3 . . . . .	68

<b>Розділ 4. Мережі Петрі . . . . .</b>	<b>69</b>
§ 4.1. Поняття класичних мереж Петрі . . . . .	69
§ 4.2. Формальне означення мереж Петрі і їхнє матричне подання . . . . .	73
§ 4.3. Властивості мереж Петрі . . . . .	75
§ 4.4. Матричний аналіз мереж Петрі . . . . .	77
§ 4.5. Дерево досяжності . . . . .	82
§ 4.6. Некласичні мережі Петрі . . . . .	88
§ 4.7. Програма HPetriSim для імітаційного моделювання мереж Петрі . . . . .	94
Питання до розділу 4 . . . . .	97
Вправи до розділу 4 . . . . .	97
<b>Розділ 5. Аналітичне моделювання систем масового обслуговування . . . . .</b>	<b>100</b>
§ 5.1. Поняття системи масового обслуговування . . . . .	100
§ 5.2. Характеристики потоку подій і вхідний потік вимог . . . . .	101
§ 5.3. Моделювання стаціонарного пуассонівського потоку . . . . .	103
§ 5.4. Правила обслуговування, організація черги і типи моделей систем масового обслуговування . . . . .	104
§ 5.5. Аналіз часової діаграми для системи масового обслуговування . . . . .	107
§ 5.6. Формула Літгла . . . . .	111
§ 5.7. Багатоканальні системи масового обслуговування з відмовами . . . . .	112
§ 5.8. Багатоканальні системи масового обслуговування з обмеженою чергою . . . . .	120
§ 5.9. Багатоканальні системи масового обслуговування з чергою довільної довжини . . . . .	127
§ 5.10. Основи операційного аналізу мереж систем масового обслуговування . . . . .	132
Питання до розділу 5 . . . . .	137
Вправи до розділу 5 . . . . .	137

**Розділ 6. Основи мови імітаційного моделювання**

<b>GPSS World</b> . . . . .	<b>144</b>
§ 6.1. Основи роботи з середовищем GPSS World . . . . .	144
§ 6.2. Категорії і типи об'єктів мови GPSS . . . . .	147
§ 6.3. Системні числові атрибути . . . . .	149
§ 6.4. Формат рядка . . . . .	150
§ 6.5. Арифметичні і логічні вирази . . . . .	153
§ 6.6. Блоки створення, знищення, затримки, зміни параметрів і пріоритету транзактів . . . . .	155
§ 6.7. Основні блоки і команди опису, пов'язані з пристроями . . . . .	159
§ 6.8. Блоки і команди опису, пов'язані зі статистичною категорією . . . . .	163
§ 6.9. Використання змінних, логічних перемикачів і комірок пам'яті . . . . .	169
§ 6.10. Блок TRANSFER для зміни маршруту руху транзактів . . . . .	172
§ 6.11. Блоки TEST, GATE і LOOP для зміни маршруту руху транзактів . . . . .	176
§ 6.12. Елементи стандартного звіту . . . . .	179
§ 6.13. Функції і генератори випадкових чисел . . . . .	186
§ 6.14. Математичні функції і функції для генерації випадкових величин . . . . .	192
§ 6.15. Приклади побудови моделей . . . . .	194
Питання до розділу 6 . . . . .	205
Вправи до розділу 6 . . . . .	206

**Розділ 7. Додаткові можливості мови і середовища GPSS World**

§ 7.1. Створення копій транзактів і синхронізація їхнього руху . . . . .	213
§ 7.2. Матриці . . . . .	215
§ 7.3. Найважливіші системні числові атрибути . . . . .	216
§ 7.4. Команди керування . . . . .	220
§ 7.5. Інформаційні вікна . . . . .	222

---

§ 7.6. Імітаційне моделювання виходу з ладу обладнання . . . . .	228
§ 7.7. Списки користувача . . . . .	233
§ 7.8. Інтегрування звичайних диференціальних рівнянь . . . . .	238
§ 7.9. PLUS-процедури . . . . .	242
§ 7.10. Опитування неактивних транзактів . . . . .	250
§ 7.11. Виконання блока . . . . .	250
Питання до розділу 7 . . . . .	251
Вправи до розділу 7 . . . . .	251
<b>Розділ 8. Моделювання випадкових величин . . . . .</b>	<b>256</b>
§ 8.1. Поняття про метод Монте-Карло . . . . .	256
§ 8.2. Генератори випадкових чисел . . . . .	259
§ 8.3. Моделювання випадкових подій . . . . .	261
§ 8.4. Метод оберненої функції . . . . .	263
§ 8.5. Моделювання дискретних випадкових величин . . . . .	267
§ 8.6. Моделювання неперервних випадкових величин . . . . .	271
§ 8.7. Приклади моделювання . . . . .	277
Питання до розділу 8 . . . . .	281
Вправи до розділу 8 . . . . .	282
<b>Розділ 9. Вибір розподілів імовірностей . . . . .</b>	<b>287</b>
§ 9.1. Методи використання даних спостережень для моделювання випадкових процесів . . . . .	287
§ 9.2. Емпіричні розподіли неперервних випадкових величин та їхнє моделювання у випадку, коли відомі дані окремих спостережень . . . . .	290
§ 9.3. Емпіричні розподіли дискретних випадкових величин та їхнє моделювання у випадку, коли відомі дані окремих спостережень . . . . .	293
§ 9.4. Емпірично розподілені випадкові величини та їхнє моделювання у випадку згрупованих даних . . . . .	296
§ 9.5. Методи оцінки незалежності вибірки . . . . .	301
§ 9.6. Вибір сім'ї теоретичних розподілів з допомогою гістограми . . . . .	306

§ 9.7. Графіки функцій щільності розподілів імовірностей . . . . .	308
§ 9.8. Аналіз ролі випадкової величини в системі . . . . .	311
§ 9.9. Аналіз підсумкової статистики . . . . .	313
§ 9.10. Визначення параметрів теоретичних розподілів . . . . .	317
§ 9.11. Графічні процедури перевірки придатності розподілу . . . . .	322
§ 9.12. Перевірка гіпотези придатності розподілу з допомогою критерію $\chi^2$ . . . . .	330
§ 9.13. Композиція функцій розподілу . . . . .	332
§ 9.14. Зміщені розподіли . . . . .	334
§ 9.15. Зрізані розподіли . . . . .	337
§ 9.16. Вибір розподілу за відсутності даних спостережень . . . . .	339
§ 9.17. Групове надходження вимог . . . . .	342
§ 9.18. Використання умовних розподілів для моделювання випадкових векторів . . . . .	342
§ 9.19. Двовимірний нормальний розподіл і його моделювання . . . . .	344
§ 9.20. Двовимірний логнормальний розподіл і його моделювання . . . . .	348
§ 9.21. Нестаціонарний пуассонівський потік і його моделювання . . . . .	349
Питання до розділу 9 . . . . .	353
Вправи до розділу 9 . . . . .	354
Критичні точки розподілу $\chi^2$ . . . . .	361
<b>Розділ 10. Планування експерименту . . . . .</b>	<b>362</b>
§ 10.1. Перехідний і стаціонарний режими . . . . .	362
§ 10.2. Визначення кількості реалізацій процесу моделювання для оцінювання ймовірності . . . . .	364
§ 10.3. Визначення кількості реалізацій процесу моделювання для оцінювання середнього значення . . . . .	367
Питання до розділу 10 . . . . .	369
Вправи до розділу 10 . . . . .	369

---

Значення функції $\Phi^{-1}(x)$ . . . . .	369
<b>Розділ 11. Прогнозування систем</b> . . . . .	<b>370</b>
§ 11.1. Основні методи прогнозування систем . . . . .	370
§ 11.2. Точки біфуркації . . . . .	371
§ 11.3. Детермінований хаос . . . . .	372
Питання до розділу 11 . . . . .	375
<b>Список рекомендованої літератури</b> . . . . .	<b>376</b>
<b>Предметний покажчик</b> . . . . .	<b>378</b>

## Передмова

Моделювання — це потужний універсальний метод дослідження й оцінювання ефективності різноманітних систем. Під математичним моделюванням розуміють процес створення для заданого реального об'єкта деякої математичної моделі, якою може бути як система рівнянь, так і комп'ютерна програма. Математичне моделювання широко використовують у різноманітних галузях і розділах сучасної науки й техніки.

Більшість посібників з математичного моделювання присвячено викладенню окремих методів математичного моделювання. Характерною рисою цього посібника є спроба ознайомити студентів з різноманітними способами математичного моделювання. Разом з тим, у зв'язку з бурхливим розвитком комп'ютерної техніки значна увага у посібнику приділена різним методам імітаційного моделювання.

Перший розділ посібника присвячений поняттю моделі і математичного моделювання. Другий розділ стосується аналітичного моделювання систем з допомогою диференціальних рівнянь. Третій і четвертий розділи присвячено відповідно використанню клітинних автоматів і мереж Петрі для моделювання систем. Аналітичне моделювання систем масового обслуговування розглядається у п'ятому розділі. Шостий і сьомий розділи присвячені опису мови імітаційного моделювання GPSS World і її використання для моделювання систем масового обслуговування. Три наступних розділи посібника стосуються імітаційного моделювання різноманітних систем. Останній розділ присвячений особливостям прогнозування складних систем.

Пропонований посібник охоплює основну частину університетської програми з навчальних дисциплін «Математичне моделювання», «Імітаційне моделювання», «Теорія систем і математичне моделювання» для студентів спеціальності «прикладна математика». Посібник може використовуватись та-

кож студентами інших спеціальностей галузі знань «Інформаційні технології» і студентами фізико-математичних спеціальностей закладів вищої освіти.

Методи моделювання ілюструються різноманітними прикладами. Кінець розв'язаних прикладів позначається символом ■, але у тих випадках, де було ймовірним «загубити» відповідь серед тексту, її написано в кінці прикладу.

Кожен розділ супроводжується питаннями для контролю та самоконтролю засвоєння матеріалу. Майже всі розділи містять вправи для самостійного розв'язування, які у поєднанні з іншими збірниками можуть бути основою для проведення практичних і лабораторних занять з певної теми. Посібник може використовуватись і як довідник, чому сприяє детальний предметний покажчик.

У списку літератури читач знайде перелік літературних джерел, у яких питання, висвітлені у цьому посібнику, викладені по-іншому або більш повно.

## Розділ 1. Основні поняття про моделювання

### § 1.1. Поняття системи

Базовими поняттями в теорії і практиці моделювання об'єктів, процесів та явищ є «система» і «модель».

У перекладі з грецької слово «система» буквально означає ціле, яке складається з частин. Можна навести багато сучасних означень системи. Для всіх цих означень спільним є те, що *система* — це сукупність елементів і зв'язків між ними, яка має певну цілісність.

Поняття елемента системи є відносним у тому плані, що елемент — це найменший об'єкт у системі, який не ділять на дрібніші об'єкти, бо це не відповідає меті дослідження. Наприклад, двигун можна вважати елементом автомобіля. У той самий час, автомобіль можна трактувати як елемент транспортної магістралі. Аналогічно поняттю елемента вводиться поняття підсистеми.

*Підсистема* — частина системи, виділена за певною ознакою, яка володіє певною самостійністю і допускає розклад на елементи.

Кожна система характеризується *емерджентністю* (*цілісністю*), згідно з якою система має таку властивість, яка не зводиться до суми властивостей всіх елементів системи.

Під *структурою* системи розуміють сукупність утворюючих систему елементів і зв'язків між ними.

Систему можна подати з допомогою глобальних рівнянь системи:

$$z(t) = f(x(\tau), z(t_0)), \quad \tau \in [t_0, t], \quad (1.1)$$

$$y(t) = g(x(t), z(t)), \quad (1.2)$$

де  $x(t) = (x_1(t), x_2(t), \dots, x_k(t))$  — вхідний сигнал,  $y(t) = (y_1(t), y_2(t), \dots, y_m(t))$  — вихідний сигнал,  $z(t) = (z_1(t),$

$z_2(t), \dots, z_n(t)$  — змінний внутрішній стан системи,  $f$  і  $g$  — деякі функції, які задають поточні значення внутрішнього стану  $z(t)$  і вихідного сигналу  $y(t)$ . Співвідношення (1.1) і (1.2) називають відповідно *рівнянням стану* і *рівнянням спостереження* системи.

У відповідності до типу значень  $x(t)$ ,  $y(t)$ ,  $z(t)$  і  $t$  системи ділять на *неперервні* і *дискретні*.

Для неперервних систем функція станів  $z(t)$  є неперервною функцією (рис. 1.1), можуть існувати також її похідні, наприклад  $\frac{dz}{dt} = F(z, x, t)$ .

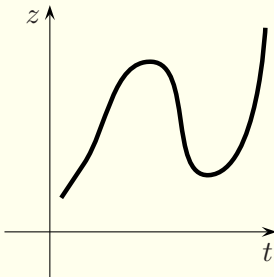


Рис. 1.1

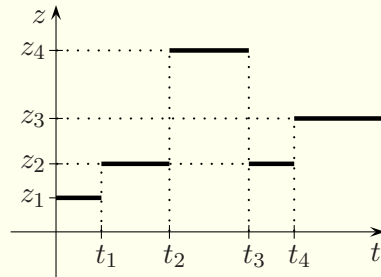


Рис. 1.2

Для дискретних систем функція станів є кусково-сталою, тобто стани набувають значень з дискретної множини. Зміни стану відбуваються не неперервно, а за принципом «від події» (рис. 1.2).

Системи бувають *детермінованими*, *ймовірнісними* (*стохастичними*) і *детерміновано-ймовірнісними*.

У детермінованих системах є однозначна залежність між вхідним і вихідним сигналами. Для ймовірнісних систем характерною є наявність випадкових впливів. Детерміновано-ймовірнісні системи в одних умовах ведуть себе детерміновано, а в інших — імовірно.

Системи ділять на статичні і динамічні. *Статичні* системи не змінюють стан, а *динамічні* можуть перебувати в одно-

му зі скінченної чи нескінченної множини станів. Зміна станів часто розглядається на часовому інтервалі  $[0, \infty)$ .

Системи також можна ділити на прості і складні. *Складні* системи характеризуються відкритістю (наявністю постійної взаємодії з навколишнім середовищем), здатністю до самоорганізації, наявністю зворотного зв'язку, непередбачуваністю. Натомість *прості* системи мають однозначний зв'язок між зовнішньою дією і реакцією системи, передбачуваність, малі зміни зовнішньої дії викликають малі зміни в реакції системи. Прикладом складної системи є будь-який живий організм.

## § 1.2. Поняття моделі

У найзагальнішому формулюванні *модель* — це об'єкт, подібний у деяких відношеннях на прототип, який використовується для опису і (або) пояснення і (або) прогнозування поведінки прототипу.

Найважливішою властивістю моделі є те, що вона дає спрощений образ, який відображає не всі властивості об'єкта, а тільки ті, які є істотними для дослідника.

Модель і реальна система знаходяться між собою у відношенні гомоморфізму (однозначного відображення)  $f : A \rightarrow B$ , де  $A$  — модель, а  $B$  — реальна система. Оскільки модель є простішою за реальну систему, відповідність  $f$  не є взаємно-однозначною.

Із давніх-давен під час вивчення складних процесів чи явищ, конструювання нових споруд люди використовували моделі. Добра модель більш доступна для дослідження, ніж реальна система, вивчення моделі дає нові знання про об'єкт-оригінал.

*Моделювання* — це спосіб дослідження будь-яких об'єктів, процесів чи явищ шляхом побудови та аналізу їхніх моделей.

Типовими цілями моделювання можуть бути пошук оптимальних чи близьких до оптимальних розв'язків, оцінка ефективності роботи системи, визначення властивостей системи,

прогнозування її розвитку.

Можна виділити три основні області використання моделей: навчання, наукові дослідження, керування. При навчанні моделі сприяють наочності відображення різних об'єктів і полегшенню передачі знань про них. У наукових дослідженнях моделі дають нові знання про досліджуваний об'єкт, процес чи явище. При керуванні моделі використовуються для прийняття рішень. Такі моделі мають забезпечувати як опис, так і прогнозування поведінки систем.

### § 1.3. Класифікація видів моделювання

Єдина класифікація видів моделювання неможлива через багатозначність поняття моделі в науці, техніці, суспільстві. Ми розглянемо чотири способи класифікації.

1. Моделювання можна ділити на детерміноване і ймовірнісне. *Детерміноване* моделювання відображає процеси, у яких припускається відсутність випадкових впливів. *Ймовірнісне (стохастичне)* моделювання враховує ймовірнісні чинники.

2. Залежно від того, яким чином відтворюються в часі стани моделі, розрізняють *неперервне, дискретне* і *змішане (дискретно-неперервне)* моделювання. Неперервні моделі логічно використовувати для моделювання неперервних систем, а дискретні — для дискретних. Проте часто бувають випадки, коли дискретні системи успішно моделюють з допомогою неперервних моделей, а неперервні — з допомогою дискретних.

Об'єднуючи попередні класифікації, отримуємо неперервно-детерміноване, дискретно-детерміноване, неперервно-ймовірнісне і дискретно-ймовірнісне моделювання.

3. Моделювання буває статичним і динамічним. *Статичне* моделювання описує стан об'єкта у фіксований момент часу, а *динамічне* — зміну станів об'єкта в часі. Значно частіше виникає потреба у динамічному моделюванні.

4. Моделювання ділять також на *уявне* і *реальне*. Уявне моделювання в свою чергу ділять на наочне, символічне і математичне, а реальне — на натурне і макетне<sup>1</sup> (рис. 1.3).

*Уявне* моделювання (його називають також *абстрактним*) є єдиним способом моделювання систем, які не можуть бути вивчені безпосередньо. Зокрема, нереально поставити експеримент по прямому дослідженню зірок чи планет. У цивілізованих країнах неприпустимо експериментувати з економікою чи здоров'ям людей. Часто безпосередні експерименти неможливі у зв'язку зі складністю, великими матеріальними затратами, унікальністю системи, тривалістю експерименту тощо. У таких випадках уявним моделям немає альтернативи.

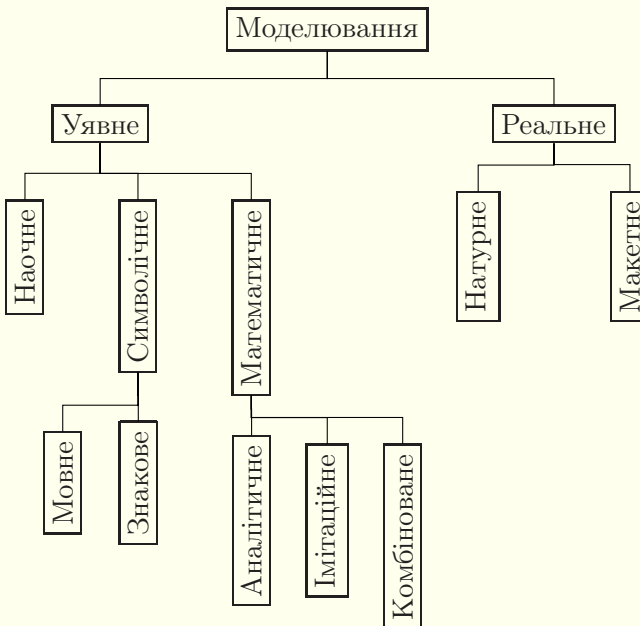


Рис. 1.3

При *наочному* моделюванні на базі уявлень людини про

<sup>1</sup>Макетне моделювання також називають фізичним.

реальні системи створюються наочні моделі, які відображають явища і процеси, що відбуваються в системі. Прикладами таких моделей є навчальні плакати, рисунки, графіки, схеми, діаграми, анімація. З допомогою спеціальних комп'ютерних програм можна створювати тривимірні моделі. До наочного моделювання також відносять створення макетів, які використовуються в навчальному процесі.

*Символічне* моделювання є штучним процесом створення логічного об'єкта, що замінює реальний і виражає його основні властивості з допомогою певної системи слів чи знаків. Символічне моделювання можна поділити на мовне і знакове.

*Мовне* моделювання передбачає опис об'єкта, процесу чи явища словами, причому ці слова мають бути однозначними. Часто до мовної моделі вдаються на початковій стадії моделювання, коли задають опис розглядуваної системи і отримують уявлення про її елементи.

Якщо ввести умовні позначення окремих понять та операцій, тобто знаки, то можна реалізувати *знакове* моделювання. Прикладами знакового моделювання є географічна карта деякої місцевості або запис хімічної реакції.

Поняття *математичного* моделювання розкривається в наступному параграфі.

При *реальному* моделюванні досліджується реальний об'єкт, його частина або подібний до нього об'єкт. Реальне моделювання ділять на натурне і макетне.

*Натурним* моделюванням називають проведення дослідження на реальному об'єкті з наступною обробкою результатів. Методами натурального моделювання є науковий експеримент, комплексні випробування і виробничий експеримент.

*Науковий експеримент* характеризується використанням різноманітних засобів обробки інформації і можливістю втручання людини в процес проведення експерименту. В ході експерименту здійснюються збурювальні впливи на процес функціонування об'єкта, при цьому можуть виявитись окремі кри-

тичні ситуації і визначитись межі стійкості процесу. *Комплексні випробування* здійснюються шляхом багаторазових повторень випробувань об'єктів. У цьому випадку моделювання зводиться до обробки й узагальнення інформації про помічені закономірності. *Виробничий експеримент* полягає в узагальненні досвіду, накопиченого під час виробничого процесу.

Іншим видом реального моделювання є *макетне* моделювання, яке відрізняється від натурного тим, що моделювання здійснюється на об'єктах, які зберігають природу явищ і є подібними до реальних систем. Часто такими об'єктами є макети, які відтворюють модельовану систему в певному масштабі. Найпростіші макетні моделі є зменшеними копіями реальних систем. Наприклад, можна створити макет гідротехнічних споруд і досліджувати його в умовах, близьких до реальних. Інші приклади макетного моделювання — дослідження крила літака або лісу, утвореного зі штучних дерев, в аеродинамічній трубі, дослідження моделі корабля в басейні.

Макетне моделювання може відбуватись у реальному і модельному часі. Наприклад, якщо дослідження поведінки об'єкта в аеродинамічній трубі здійснюються у реальному часі, то моделювання поведінки гірських систем можливе лише в модельному часі.

## § 1.4. Поняття математичного моделювання

*Математичне* моделювання — це створення математичної моделі процесів, які відбуваються у реальному об'єкті. Математичною моделлю може бути як система рівнянь, так і комп'ютерна програма. Кожна математична модель описує реальний об'єкт з деякою мірою наближення. Дослідження моделі дає можливість встановити характеристики реального об'єкта. Математичне моделювання є одним з основних способів моделювання систем.

Математичне моделювання ділять на аналітичне, імітаційне і комбіноване.

При *аналітичному* моделюванні глобальні рівняння системи, які описують її закон функціонування, записуються у вигляді деяких аналітичних співвідношень (алгебричних, диференціальних, інтегральних, інтегро-диференціальних та інших рівнянь чи їхніх систем) і деяких додаткових умов (початкових, крайових, багатоточкових умов чи деяких обмежень). У цьому випадку моделюється переважно функціональний аспект системи. Найчастіше зустрічаються диференціальні моделі, в яких реальні процеси описуються з допомогою звичайних диференціальних рівнянь або рівнянь з частинними похідними (розділ 2).

Аналітична модель досліджується кількома методами: аналітичними, числовими або якісними.

Аналітичні методи дозволяють отримати в загальному вигляді залежності, що пов'язують шукані характеристики з початковими умовами, параметрами і змінними станами моделі.

Числові методи використовують тоді, коли не вдається знайти розв'язок рівняння чи системи рівнянь аналітичними методами. У цьому випадку знаходять наближений розв'язок у числовій формі.

Якісні методи також використовуються у тому випадку, коли, не маючи розв'язку в аналітичному вигляді, можна знайти деякі властивості розв'язку (наприклад, оцінити стійкість чи продовжуваність розв'язку). Якісні методи часто використовують разом з числовими.

Якщо задачу можливо розв'язати і аналітично і чисельно, то віддавати перевагу треба аналітичному методу. Деколи аналітичний розв'язок дозволяє отримати висновки якісного характеру, які неможливо зробити на основі числового розв'язку.

При *імітаційному* моделюванні математичною моделлю є, як правило, комп'ютерна програма, яка відтворює поведінку системи в часі, причому імітуються елементарні явища, які складають процес, зі збереженням їхньої логічної структури

і послідовності протікання. Це дозволяє за вхідними даними отримати інформацію про стани системи в певні моменти часу та оцінити характеристики системи. У наш час імітаційне моделювання є дуже поширеним у зв'язку з активним розвитком комп'ютерної техніки.

Основною перевагою імітаційного моделювання порівняно з аналітичним є можливість розв'язування більш складних задач, для яких не вдається побудувати систему рівнянь, що їх описують, або не вдається її розв'язати. Імітаційні моделі дозволяють досить просто враховувати такі фактори, як наявність дискретних і неперервних елементів, нелінійні характеристики елементів системи, випадкові впливи та інше, що часто створює труднощі при аналітичних дослідженнях.

В рамках імітаційного моделювання виділяють метод Монте-Карло, метод мереж Петрі, метод клітинних автоматів, метод нейронних мереж та інші.

Метод Монте-Карло використовується для моделювання систем з випадковими впливами, ймовірнісні характеристики яких відомі, і полягає в багаторазовому відтворенні моделі з наступною обробкою інформації методами математичної статистики (§ 8.1). Метод мереж Петрі використовують для імітаційного та аналітичного моделювання дискретних технологічних процесів (розділ 4). Клітинні автомати дозволяють моделювати різноманітні процеси у дискретному просторі і часі з випадковими впливами чи без них (розділ 3).

Метод нейронних мереж полягає в тому, що створену нейронну мережу на основі тестових прикладів навчають діяти так, як діє реальна система. Процес цей досить трудомісткий. Навчена нейронна мережа буде працювати так, як система, імітуючи її. До речі, нейронна мережа не містить фізичних нейронів. Чи при її програмній реалізації, чи при апаратній реалізації нейрони моделюються кількома елементарними об'єктами. З'єднання нейронів задається певним чином при створенні мережі.

Імітаційна модель може бути написана на універсальних мовах програмування, таких як Python, C++, Java, але краще використовувати спеціальні мови і середовища моделювання, як-от: GPSS, AnyLogic, Arena, AweSim, PowerSim, Extend тощо. У **шостому** і **сьомому** розділах цього посібника розглядається мова імітаційного моделювання GPSS World.

*Комбіноване (аналітико-імітаційне)* моделювання дозволяє об'єднати переваги аналітичного й імітаційного моделювання. Спочатку здійснюється розбиття процесу функціонування об'єкта на складові підпроцеси. Там, де це можливо, використовуються аналітичні моделі, а для інших підпроцесів будуються імітаційні моделі.

Часто використовується термін «комп'ютерне моделювання». Під *комп'ютерним моделюванням* розуміють програмну реалізацію математичної моделі (аналітичної чи імітаційної), доповнену різними програмними і апаратними засобами.

## § 1.5. Принципи математичного моделювання

При побудові математичних моделей дуже великою є роль досвіду, інтуїції та інших інтелектуальних якостей людини. Тому неможливо скласти формалізовану інструкцію, яка б визначила, як повинна будуватись модель тієї чи іншої системи. А втім, відсутність точної інструкції не заважає досвідченим спеціалістам будувати вдалі моделі. Розглянемо основні *принципи побудови моделей*.

1. *Достатність інформації*. Якщо інформації надто мало, то модель побудувати не вдасться, якщо все відомо, то немає потреби в моделюванні. Існує критичний рівень достатності інформації для моделювання.

2. *Адекватність*. Цей принцип передбачає відповідність моделі меті дослідження і важливим для дослідника аспектам реальної системи.

3. *Відповідність моделі розв'язуваній задачі*. Модель має будуватись для розв'язування певного класу задач чи кон-

кретної задачі дослідження системи. Спроби створення універсальної моделі, спрямованої на розв'язування великої кількості різноманітних задач, призводять до такого ускладнення, що модель виявляється практично непридатною. Досвід показує, що при розв'язуванні кожної конкретної задачі треба мати свою модель, яка відображатиме ті аспекти системи, які є найважливішими в цій задачі. Цей принцип тісно пов'язаний з принципом адекватності.

4. *Спрощення при збереженні істотних властивостей.* Модель має бути в деяких відношеннях простішою за прототип — у цьому сенс моделювання. Чим складнішою є розглядувана система, тим по можливості простішим має бути її опис, який би зумисно підкреслював типові (суттєві в цій задачі) й ігнорував менш суттєві властивості. Цей принцип можна ще назвати абстрагуванням від другорядних деталей.

5. *Відповідність між потрібною точністю результатів моделювання і складністю моделі.* Моделі по своїй природі є наближеними. Для підвищення точності моделі її потрібно деталізувати, що ускладнює модель. Тому потрібно шукати певний компроміс між точністю і складністю моделі, що часто досягається методом спроб і помилок.

6. *Врахування похибок різних видів* (моделювання, вхідних даних і заокруглень). Модель завжди наближена по своїй природі. Крім того, вхідні дані також відомі з деякою точністю. Тому, наприклад, здійснювати обчислення з десятьма значущими цифрами не має сенсу.

7. *Блочна будова.* При виконанні цього принципу полегшується розробка складних моделей і з'являється можливість використання накопиченого досвіду і готових блоків з мінімальними зв'язками між ними.

Практичні рекомендації по *спрощенню моделей.*

1. Зменшення кількості змінних у моделі, тобто деякі змінні параметри розглядаються як сталі.

2. Зміна природи змінних параметрів: дискретні парамет-

ри замінують неперервними або навпаки. Наприклад, кількість живих організмів у популяції можна вважати неперервною функцією часу (§ 2.6) або час вважати дискретним (§§ 3.5, 3.6).

3. Зміна функціональної залежності між змінними: нелінійна залежність часто замінюється лінійною. Наприклад, замість нелінійного диференціального рівняння  $x'' + k \sin x = 0$ , яке описує коливання математичного маятника, розглядають лінійне рівняння  $x'' + kx = 0$ , яке наближено описує малі коливання математичного маятника.

4. Зміна обмежень: додавання, вилучення чи їхня модифікація.

## § 1.6. Етапи математичного моделювання

Виділяють наступні етапи побудови і дослідження математичних моделей.

1. Формулювання проблеми та змістова постановка задачі, визначення мети моделювання.

2. Вивчення системи і процесів, які в ній протікають.

3. Формалізація — побудова рівнянь або складання комп'ютерної програми.

4. Дослідження моделі.

5. Перевірка адекватності моделі.

6. Аналіз результатів моделювання.

7. Оформлення результатів дослідження.

На першому етапі визначаються мета і завдання моделювання, ставиться задача. На другому етапі формується уявлення про систему: здійснюється визначення сукупності елементів, зв'язків між ними, можливих станів кожного елемента тощо. Результатом другого етапу є узагальнена схема моделі, повністю готова до побудови математичної моделі.

На етапі формалізації обирається спосіб математичного моделювання і розробляється сама модель. На цьому етапі будуються диференціальні чи інші рівняння, які описують про-

цес, або складається комп'ютерна програма, створюється мережа Петрі чи використовуються інші способи для моделювання системи. Методи математичного моделювання розглядаються у наступних розділах цього навчального посібника.

Під час дослідження моделі шукають і досліджують аналітичний чи числовий розв'язок задачі при аналітичному моделюванні або планують і проводять комп'ютерний експеримент з моделлю для досягнення поставлених цілей.

На етапі перевірки адекватності моделі здійснюється порівняння отриманої моделі з реальною системою. Адекватність перевіряється порівнянням результатів моделювання з експериментальними даними. Модель завжди лише приблизно відповідає реальній системі, абсолютно адекватних моделей не існує. Важливою є відповідність моделі конкретній меті дослідження. За результатами перевірки адекватності приймається рішення про необхідність коригування або оптимізації моделі.

На наступному етапі аналізують і оцінюють результати моделювання. Наводять результати у вигляді аналітичних функцій, графіків, таблиць. За отриманими результатами формулюють висновки і приймають рішення.

## Питання до розділу 1

1. Що таке система? Що таке емерджентність системи? Як можна класифікувати системи?
2. Що таке модель? В яких областях найчастіше використовують моделі? Назвіть основні види моделювання систем. У чому полягає математичне моделювання?
3. Назвіть основні принципи побудови математичних моделей.
4. Які способи спрощення моделей найчастіше використовують?
5. Назвіть етапи створення математичних моделей.

## Розділ 2. Диференціальні моделі

### § 2.1. Побудова диференціальних моделей

*Диференціальними* називають моделі, які задаються з допомогою диференціальних рівнянь. Цей спосіб моделювання відносять до неперервного моделювання, хоча в деяких випадках його використовують для моделювання дискретних систем. Найпоширеніший метод побудови диференціальних рівнянь полягає у використанні фундаментальних законів природи у конкретній ситуації. Наведемо декілька таких законів:

– другий закон Ньютона ( $F = ma$ , де  $m$  — маса тіла,  $a$  — прискорення руху,  $F$  — сума сил, що діють на тіло) і похідні від нього закони збереження енергії, кількості руху, імпульсу тощо;

– закон всесвітнього тяжіння ( $F = k \frac{m_1 m_2}{r^2}$ , де  $m_1, m_2$  — маси двох тіл,  $r$  — відстань між ними,  $k$  — коефіцієнт пропорційності);

– закон розчинення речовини (швидкість розчинення пропорційна наявній кількості нерозчиненої речовини та різниці концентрацій насиченого розчину і розчину у певний момент часу);

– закон Кірхгофа (алгебраїчна сума сил струмів, які протікають у певній точці електричного кола, дорівнює нулю);

– закон Фур'є ( $q = -\lambda(T) \frac{dT}{dx}$ , де  $q$  — питомий потік теплоти,  $\lambda(T)$  — коефіцієнт теплопровідності середовища,  $\frac{dT}{dx}$  — швидкість зміни температури  $T$ ) і аналогічний закон Нернста про дифузію речовини;

– закон Ньютона про охолодження тіла (швидкість охолодження тіла прямо пропорційна різниці температур тіла та оточуючого середовища);

– закон Гука (сила пружності пружини пропорційна її видовженню) тощо.

Можна рекомендувати дотримуватись такої послідовності дій при побудові і дослідженні диференціальних моделей:

1) встановити величини, які змінюються у заданому явищі чи процесі, і виявити закони (формули) відповідної науки, які ці величини пов'язують;

2) вибрати незалежні змінні і функції цих змінних, які потрібно знайти;

3) виходячи з відомих даних, визначити початкові, крайові або інші умови, які накладаються на шукані функції;

4) виразити усі величини через незалежні змінні, шукані функції та їхні похідні;

5) виходячи з закону, який описує задане явище, скласти диференціальне рівняння чи систему таких рівнянь;

6) зінтегрувати одержані диференціальні рівняння;

7) якщо задані початкові чи інші умови, знайти частинний розв'язок;

8) провести дослідження одержаного розв'язку.

Деколи замість пунктів 4) і 5) використовують такий підхід:

4а) виразити прирости функцій через незалежні змінні, шукані функції і прирости незалежних змінних;

5а) перейшовши до границі, коли прирости незалежних змінних прямують до нуля, отримати диференціальні рівняння.

Питання про відповідність диференціальної моделі й реального явища вивчається на основі аналізу результатів дослідження та їхнього порівняння з поведінкою розв'язку одержаного диференціального рівняння.

Якщо диференціальне рівняння можна розв'язати і аналітично і чисельно, то віддавати перевагу треба аналітичному методу, бо аналітичний розв'язок дозволяє зробити висновки якісного характеру, які неможливо зробити на основі числового розв'язку. Наприклад, розв'язавши крайову задачу для рівняння коливань струни  $\frac{\partial^2 u}{\partial t^2} = a^2 \frac{\partial^2 u}{\partial x^2}$  методом відокремлення змінних, можна зробити висновок про накладання хвиль і вплив окремих гармонік на формування розв'язку. Тут про-

слідковується тісний зв'язок з тембром звучання струни, наявністю крім основної допоміжних частот, які забезпечують відмінність звучання різних типів музичних інструментів на тих самих нотах. Числовий розв'язок звичайно можна відобразити у вигляді анімації на площині чи поверхні у просторі, але він не дає змоги зробити згадані висновки, хоча це той самий розв'язок згідно з теоремою існування і єдиності.

У наступних параграфах розглянемо кілька прикладів диференціальних моделей для різних явищ, процесів і об'єктів. Для глибшого ознайомлення з диференціальними моделями пропонується використовувати книги [4, 8, 14, 16].

## § 2.2. Падіння тіла

З деякої висоти вертикально вниз кинули тіло масою  $m$ . Знайдемо закон зміни швидкості  $v$  падіння цього тіла, якщо на нього діє сила ваги і гальмівна сила опору повітря, пропорційна квадрату швидкості. Також знайдемо максимально можливу швидкість падіння тіла у повітрі.

Задача полягає у відшуванні закону зміни швидкості  $v$  з плином часу  $t$ , тобто  $v(t)$ . З другого закону Ньютона маємо:

$$m \frac{dv}{dt} = F.$$

Сила  $F$  складається з двох сил: сили ваги  $F_1 = mg$  і сили опору повітря  $F_2 = -kv^2$ , де  $k$  — коефіцієнт пропорційності. Тоді

$$m \frac{dv}{dt} = mg - kv^2.$$

Відокремлюємо змінні й інтегруємо рівняння:

$$\begin{aligned} \frac{m dv}{mg - kv^2} = dt &\Rightarrow -\frac{m}{k} \int \frac{dv}{v^2 - \frac{mg}{k}} = t + C \Rightarrow \\ -\frac{m}{k} \frac{1}{2\sqrt{\frac{mg}{k}}} \ln \left| \frac{v - \sqrt{\frac{mg}{k}}}{v + \sqrt{\frac{mg}{k}}} \right| &= t + C \Rightarrow \end{aligned}$$

$$\begin{aligned} \frac{v - \sqrt{\frac{mg}{k}}}{v + \sqrt{\frac{mg}{k}}} &= C e^{-2\sqrt{\frac{kg}{m}}t} \Rightarrow \\ v - \sqrt{\frac{mg}{k}} &= v C e^{-2\sqrt{\frac{kg}{m}}t} + \sqrt{\frac{mg}{k}} C e^{-2\sqrt{\frac{kg}{m}}t} \Rightarrow \\ v &= \frac{\sqrt{\frac{mg}{k}} \left(1 + C e^{-2\sqrt{\frac{kg}{m}}t}\right)}{1 - C e^{-2\sqrt{\frac{kg}{m}}t}}. \end{aligned} \quad (2.1)$$

Нехай початкова швидкість (при  $t = 0$ ) дорівнює  $v_0$ . Тоді

$$C = \frac{v_0 - \sqrt{\frac{mg}{k}}}{v_0 + \sqrt{\frac{mg}{k}}},$$

зокрема, якщо  $v_0 = 0$ , то  $C = -1$ .

Для того щоб знайти максимальну можливу швидкість падіння, перейдемо у формулі (2.1) до границі при  $t \rightarrow +\infty$ . Тоді  $v_{max} = \sqrt{\frac{mg}{k}}$ . Можна і іншим шляхом отримати цю швидкість:

$$F = 0 \Rightarrow mg = kv^2 \Rightarrow v_{max} = \sqrt{\frac{mg}{k}}.$$

При падінні тіла опір повітря залежить від величини, форми, ваги тіла, і також від густини повітря. Ця залежність враховується коефіцієнтом  $k$ , який визначається формулою

$$k = \alpha \frac{S\gamma}{g},$$

де  $S$  — площа проєкції тіла на площину, перпендикулярну до напрямку руху (в  $\text{м}^2$ ),  $\gamma$  — вага  $1 \text{ м}^3$  повітря (в Н, в середньому  $\gamma = 12 \text{ Н/м}^3$ , що відповідає вазі  $1 \text{ м}^3$  повітря при тиску  $760 \text{ мм рт. ст.}$  і температурі  $15^\circ\text{C}$ ),  $g$  — прискорення вільного падіння (в  $\text{м/с}^2$ , на рівні поверхні Землі  $g = 9,8 \text{ м/с}^2$ ),  $\alpha$  — безрозмірний коефіцієнт опору, який залежить від форми тіла. Наприклад, для горизонтально падаючої квадратної пластинки  $\alpha = 0,631$ , для півсфери отвором вниз  $\alpha = 0,664$ .

### § 2.3. Вентиляція приміщення

У лекційній аудиторії кубатурою  $200 \text{ м}^3$  повітря після лекції містить  $0,1 \%$  вуглекислоти. Кондиціонер подає свіже повітря, що містить  $0,04 \%$  вуглекислого газу, в кількості  $a \text{ м}^3/\text{хв}$ . Припустивши, що змішування чистого повітря з забрудненим відбувається миттєво, обчислити, якою має бути величина  $a$ , щоб після  $10$  хвилин перерви вміст вуглекислого газу в аудиторії не перевищував  $0,06 \%$ .

Позначимо вміст вуглекислого газу (в  $\%$ ) у повітрі в момент часу  $t$  через  $y(t)$ . Розглянемо деякий проміжок часу  $\Delta t$  і знайдемо зміну концентрації вуглекислого газу в аудиторії за цей проміжок, вважаючи, що процес є рівномірним. За цей час вентилятор подає  $0,0004a\Delta t \text{ м}^3$  вуглекислого газу, а виходить його з приміщення назовні через шпарини  $0,01y(t + \alpha)a\Delta t \text{ м}^3$ . Отже, за  $\Delta t$  хвилин кількість вуглекислоти в повітрі змінюється на  $0,0004a\Delta t - 0,01y(t + \alpha)a\Delta t \text{ м}^3$ . З іншого боку,  $(y(t + \Delta t) - y(t))0,01 \cdot 200$  – це приріст вуглекислоти в приміщенні. Отже,

$$(y(t + \Delta t) - y(t))0,01 \cdot 200 = (0,0004 - 0,01y(t + \alpha))a\Delta t \Rightarrow 200dy = (0,04 - y)a dt.$$

Таким чином, одержали рівняння з відокремленими змінними. Зінтегруємо його:

$$200 \frac{dy}{0,04 - y} = a dt \Rightarrow y = 0,04 + Ce^{-\frac{at}{200}}.$$

Оскільки  $y(0) = 0,1$ , то  $C = 0,06$  і

$$y = 0,04 + 0,06e^{-\frac{at}{200}}.$$

Внаслідок умови  $y(10) = 0,06$  маємо:

$$0,06 = 0,04 + 0,06e^{-\frac{10a}{200}} \Rightarrow a = 20 \ln 3 \approx 22 \text{ м}^3/\text{хв}.$$

### § 2.4. Витікання рідини

Нехай деяка посудина заповнена рідиною до рівня  $H$ . У дні посудини є отвір площею  $\omega$ , через який витікає рідина. Площа поперечного перерізу посудини горизонтальною площиною є відомою функцією  $S(h)$  висоти  $h$  від дна посудини. Потрібно визначити час  $T$  витікання всієї рідини з посудини та час  $t$  зниження рівня рідини від початкового  $H$  до заданого  $h$  (рис. 2.1).

Нехай висота рівня рідини в посудині у деякий момент часу  $t$  дорівнює  $h$ . Досліди показують, що зі зменшенням рівня рідини у посудині швидкість витікання рідини зменшується. Нехай  $v(h)$  — швидкість витікання рідини, залежна від рівня  $h$  рідини в посудині. Об'єм рідини  $\Delta V$ , що витікає з посудини за проміжок часу  $\Delta t$  від моменту  $t$  до моменту  $t + \Delta t$ , можна обчислити як об'єм циліндра з площею основи  $\omega$  і висотою  $v(h)\Delta t$ . Таким чином,

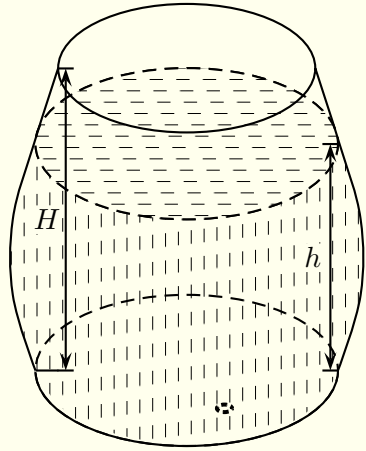


Рис. 2.1

$$\Delta V = \omega v(h)\Delta t.$$

Цей самий об'єм рідини може бути обчислений іншим способом. Внаслідок витікання рідини її рівень  $h$  змінюється на величину  $\Delta h$ , отже,  $\Delta V = -S(h)\Delta h$  (знак мінус береться тому, що  $\Delta h < 0$ ). Прирівнюючи обидва вирази для  $\Delta V$ , маємо рівняння

$$\omega v(h)\Delta t = -S(h)\Delta h,$$

звідки

$$\frac{\Delta t}{\Delta h} = -\frac{S(h)}{\omega v(h)}.$$

Спрямовуючи  $\Delta h$  до нуля, отримуємо диференціальне рівняння

$$\frac{dt}{dh} = -\frac{S(h)}{\omega v(h)}.$$

Інтегруючи це рівняння і враховуючи початкову умову  $t = 0$  для  $h = H$ , приходимо до формул

$$t = \int_h^H \frac{S(h)}{\omega v(h)} dh, \quad T = \int_0^H \frac{S(h)}{\omega v(h)} dh. \quad (2.2)$$

Якщо витікання рідини відбувається через невеликий отвір або коротку трубку, то згідно з законом Торрічеллі  $v = \mu\sqrt{2gh}$ , де  $g$  — прискорення вільного падіння ( $g \approx 9,8 \frac{\text{М}}{\text{с}^2}$ ), а  $\mu$  — коефіцієнт витрат (для води  $\mu = 0,62$ ). У цьому випадку формули (2.2) набувають вигляду

$$t = \frac{1}{\omega\mu\sqrt{2g}} \int_h^H \frac{S(h)}{\sqrt{h}} dh, \quad T = \frac{1}{\omega\mu\sqrt{2g}} \int_0^H \frac{S(h)}{\sqrt{h}} dh. \quad (2.3)$$

**Приклад.** Визначити час витікання всього гасу з заповненої циліндричної залізничної цистерни довжиною  $L = 12$  м і діаметром  $D = 2,6$  м через коротку зливну трубку з площею поперечного перерізу  $\omega = 20$  см<sup>2</sup> у нижній частині цистерни. Коефіцієнт витрат для гасу  $\mu = 0,6$ .

**Розв'язання.** Зрозуміло, що в нормальних умовах вісь циліндра залізничної цистерни розміщена горизонтально. Тому кожен переріз цієї цистерни горизонтальною площиною є прямокутником зі сталою довжиною  $L$  і змінною шириною  $2x$  (рис. 2.2). Знайдемо половину ширини  $x$ . Нехай  $R = \frac{D}{2}$  — радіус циліндра. На рис. 2.3 зображено поперечний переріз цистерни вертикальною площиною, перпендикулярною до осі циліндра. За теоремою Піфагора  $x = \sqrt{R^2 - (h - R)^2}$ . Отже,

$S(h) = 2L\sqrt{(D-h)h}$ , а тому

$$T = \frac{2L}{\omega\mu\sqrt{2g}} \int_0^D \sqrt{D-h} dh = \frac{4LD\sqrt{D}}{3\omega\mu\sqrt{2g}} \approx 12626 \text{ с} \approx 3,5 \text{ год.} \blacksquare$$

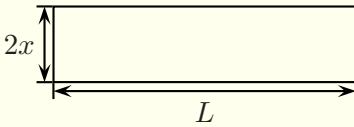


Рис. 2.2

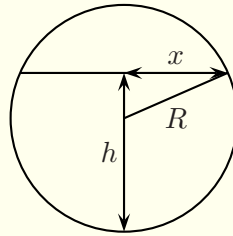


Рис. 2.3

Нехай поряд з витіканням рідини є її постійне надходження у кількості  $q$  одиниць об'єму за одиницю часу. Тоді можна аналогічно вивести формулу для часу витікання частини рідини

$$t = \int_h^H \frac{S(h)}{\omega v(h) - q} dh.$$

## § 2.5. Потік частинок у трубці

Нехай у трубці сталого поперечного перерізу  $S$  рухаються деякі частинки (наприклад, пилюка в трубці порохотяга або електрони в електронній трубці) з відомою швидкістю  $u(t) > 0$ , залежною лише від часу  $t$  і не залежною від координати (рис. 2.4). Припустимо, що густина частинок досить мала і тому частинки між собою не взаємодіють (не зіштовхуються, не притягаються і не здійснюють впливу одна на одну). Потрібно знайти густину частинок  $\rho(x, t)$  у будь-який момент часу, якщо початкова густина задана відомою функцією  $\rho_0(x)$ .

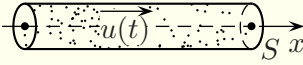


Рис. 2.4

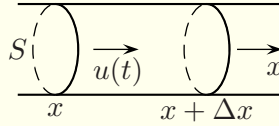


Рис. 2.5

Розглянемо елемент трубки довжиною  $\Delta x$  від точки  $x$  до точки  $x + \Delta x$  (рис. 2.5). Протягом часу  $\Delta t$  через лівий кінець цього елемента проходять частинки загальною масою  $Su(t + \xi_1 \Delta t)\rho(x, t + \xi_2 \Delta t)\Delta t$ , а через правий кінець —  $Su(t + \xi_1 \Delta t)\rho(x + \Delta x, t + \xi_3 \Delta t)\Delta t$ , де  $u(t + \xi_1 \Delta t)$  — середнє значення швидкості, а  $\rho(x, t + \xi_2 \Delta t)$  і  $\rho(x + \Delta x, t + \xi_3 \Delta t)$  — середні густини за час  $\Delta t$ ,  $0 \leq \xi_1, \xi_2, \xi_3 \leq 1$ . Отже, зміна маси частинок в елементі трубки за час  $\Delta t$  становить

$$\Delta m = Su(t + \xi_1 \Delta t)(\rho(x, t + \xi_2 \Delta t) - \rho(x + \Delta x, t + \xi_3 \Delta t))\Delta t.$$

З іншого боку,

$$\Delta m = S(\rho(x + \xi_4 \Delta x, t + \Delta t) - \rho(x + \xi_5 \Delta x, t))\Delta x, \quad 0 \leq \xi_4, \xi_5 \leq 1,$$

де  $\rho(x + \xi_4 \Delta x, t + \Delta t)$  і  $\rho(x + \xi_5 \Delta x, t)$  — середні густини частинок в елементі трубки в кінцевий і початковий моменти часу.

Тоді згідно з законом збереження маси

$$\begin{aligned} u(t + \xi_1 \Delta t)(\rho(x, t + \xi_2 \Delta t) - \rho(x + \Delta x, t + \xi_3 \Delta t))\Delta t = \\ = (\rho(x + \xi_4 \Delta x, t + \Delta t) - \rho(x + \xi_5 \Delta x, t))\Delta x. \end{aligned}$$

Поділивши обидві частини останнього рівняння на  $\Delta x$ ,  $\Delta t$  і спрямувавши їх до нуля, отримуємо диференціальне рівняння з частинними похідними першого порядку:

$$u(t) \frac{\partial \rho}{\partial x} + \frac{\partial \rho}{\partial t} = 0. \quad (2.4)$$

Початкова умова для цього рівняння має вигляд:

$$\rho(x, t)|_{t=0} = \rho_0(x). \quad (2.5)$$

Зінтегруємо лінійне однорідне рівняння (2.4) методом характеристик:

$$\frac{dx}{u(t)} = \frac{dt}{1} \Rightarrow dx = u(t)dt \Rightarrow x - \int_0^t u(\tau)d\tau = C.$$

Тоді загальним розв'язком рівняння (2.4) є функція

$$\rho(x, t) = \varphi\left(x - \int_0^t u(\tau)d\tau\right),$$

де  $\varphi(\xi)$  — довільна неперервно диференційовна функція. З початкової умови (2.5):

$$\rho(x, 0) = \varphi(x) = \rho_0(x).$$

Отже,

$$\rho(x, t) = \rho_0\left(x - \int_0^t u(\tau)d\tau\right)$$

— розв'язок початкової задачі (2.4), (2.5).

Припустимо, що  $u(t) \equiv u_0$  (швидкість стала). Тоді густина

$$\rho(x, t) = \rho_0(x - u_0 t).$$

Профіль густини без змін переноситься вздовж потоку зі сталою швидкістю (рис. 2.6).

Тому має місце біжуча хвиля.

У випадку, коли швидкість  $u(t)$  не є сталою, профіль густини переноситься за рівні проміжки часу на різні відстані.

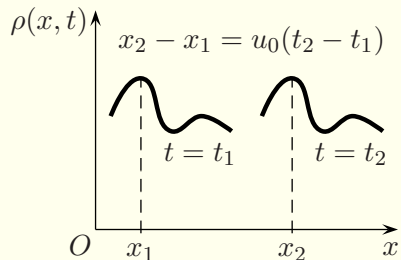


Рис. 2.6

## § 2.6. Взаємодія в системі «хижак-жертва»

Під час першої світової війни суттєво (від 12 % до 36 %) зросли відсотки хрящових риб (акули, скати і т. п.) у промислових виловах у Середземному морі. Хрящові риби не мають промислової цінності, їхнє м'ясо люди в їжу зазвичай не вживають. Зрозуміло, що під час війни вилов риби в Середземному морі зменшився, але екологи і біологи не змогли пояснити, чому зросла кількість хрящових риб, які є хижаками. Один з біологів звернувся до відомого італійського математика Віто Вольтерри з проханням пояснити, у чому річ. Вольтерра склав математичну модель співіснування хижих і нехижих риб. Складемо її і ми.

Нехай  $x(t)$  — кількість нехижих риб у момент часу  $t$ , а  $y(t)$  — кількість хижих риб у момент часу  $t$ . Якщо кількість риб достатньо велика, то ми можемо вважати, що зміна чисельності на одну рибину є незначною, а функція кількості риб є неперервною і диференційовною. Приріст числа нехижих риб пропорційний їхній кількості з коефіцієнтом пропорційності  $\alpha$  (закон Мальтуса). У той самий час, число контактів між хижими і нехижими рибами пропорційно добутку кількостей цих риб. Таким чином,  $\frac{dx}{dt} = \alpha x - \beta xy$ , де  $\beta$  — коефіцієнт смертності нехижих риб. Аналогічно,  $\frac{dy}{dt} = -\gamma y + \delta xy$ , де  $\gamma$  — коефіцієнт смертності хижаків, а  $\delta$  — їхнього розмноження (чим більше їжі та хижаків, тим більше стає хижаків). Коефіцієнти  $\alpha$ ,  $\beta$ ,  $\gamma$  і  $\delta$  є додатними. Отже, маємо систему нелінійних диференціальних рівнянь першого порядку

$$\begin{cases} \frac{dx}{dt} = \alpha x - \beta xy, \\ \frac{dy}{dt} = -\gamma y + \delta xy, \end{cases} \quad (2.6)$$

яку називають системою диференціальних рівнянь Лотки–Вольтерри (Альфред Лотка практично одночасно отримав цю систему для динаміки популяцій). Нехай  $x_0$  і  $y_0$  — початкові кількості нехижих і хижих риб, тоді отримуємо початкові умови  $x(0) = x_0$ ,  $y(0) = y_0$ .

Для того щоб динамічна система мала положення рівноваги, необхідно і достатньо, щоб для нього праві частини системи перетворювались у нуль. Отже, система (2.6) має два положення рівноваги:  $x \equiv 0$ ,  $y \equiv 0$  і  $\bar{x} \equiv \frac{\gamma}{\delta}$ ,  $\bar{y} \equiv \frac{\alpha}{\beta}$ . Перший розв'язок нецікавий, бо він означає відсутність всіх риб. Другий розв'язок означає, що чисельність риб з часом не змінюється.

Поділимо перше рівняння системи (2.6) на друге і зінтегруємо результат:

$$\begin{aligned} \frac{dx}{dy} &= \frac{(\alpha - \beta y)x}{(-\gamma + \delta x)} \Rightarrow \frac{(-\gamma + \delta x)dx}{x} = \frac{\alpha - \beta y}{y} dy \Rightarrow \\ &-\gamma \ln x + \delta x = \alpha \ln y - \beta y + C \Rightarrow \\ &y^\alpha e^{-\beta y} x^\gamma e^{-\delta x} = C \end{aligned} \quad (2.7)$$

(тут стали  $C$  перепозначено, а модулів під знаком логарифма немає, бо  $x > 0$ ,  $y > 0$ ).

**Теорема.** Всі фазові траєкторії системи (2.6) є замкненими.

**Доведення.** Спочатку визначимо поведінку функцій  $f(y) = y^\alpha e^{-\beta y}$  і  $g(x) = x^\gamma e^{-\delta x}$  для додатних  $x$  і  $y$ . Бачимо, що  $f(0) = 0$ ,  $f(\infty) = 0$  і  $f(y) > 0$  для всіх  $y > 0$ . Знайшовши похідну

$$f'(y) = \frac{\alpha y^{\alpha-1} - \beta y^\alpha}{e^{\beta y}} = \frac{y^{\alpha-1}(\alpha - \beta y)}{e^{\beta y}},$$

помічаємо, що функція  $f(y)$  має одну критичну точку  $y = \frac{\alpha}{\beta}$ . Отже,  $f(y)$  досягає свого максимального значення  $M_y = \left(\frac{\alpha}{\beta}\right)^\alpha e^{-\alpha}$  при  $y = \frac{\alpha}{\beta}$ , а графік функції  $f(y)$  має вигляд, показаний на рис. 2.7. Аналогічно, функція  $g(x)$  досягає свого максимального значення  $M_x = \left(\frac{\gamma}{\delta}\right)^\gamma e^{-\gamma}$  при  $x = \frac{\gamma}{\delta}$ , а графік функції  $g(x)$  має вигляд, поданий на рис. 2.8.

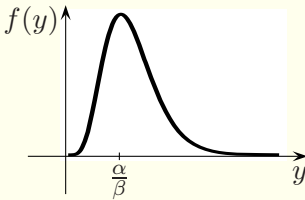


Рис. 2.7.

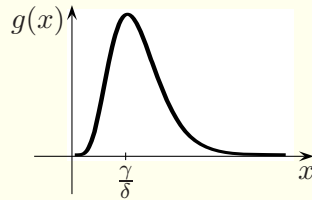


Рис. 2.8.

З попереднього аналізу робимо висновок про те, що рівняння (2.7) не має розв'язків  $x, y > 0$  для  $C > M_x M_y$  і має тільки єдиний розв'язок  $x = \frac{\gamma}{\delta}$ ,  $y = \frac{\alpha}{\beta}$  для  $C = M_x M_y$ . Таким чином, нам залишається розглянути випадок  $C = \lambda M_y$ , де  $\lambda$  є додатним числом, меншим за  $M_x$ . Зазначимо, що рівняння  $x^\gamma e^{-\delta x} = \lambda$  має один розв'язок  $x = x_m < \frac{\gamma}{\delta}$  і один розв'язок  $x = x_M > \frac{\gamma}{\delta}$ . Звідси випливає, що рівняння

$$f(y) = y^\alpha e^{-\beta y} = \left( \frac{\lambda}{x^\gamma e^{-\delta x}} \right) M_y$$

не має розв'язку  $y$ , коли  $x < x_m$  або  $x > x_M$ . Воно має єдиний розв'язок  $y = \frac{\alpha}{\beta}$ , коли  $x = x_m$  або  $x = x_M$ , і має два розв'язки  $y_1(x)$  і  $y_2(x)$  для кожного  $x$  між  $x_m$  і  $x_M$ . Менший розв'язок  $y_1(x)$  завжди менший за  $\frac{\alpha}{\beta}$ , тоді як більший розв'язок  $y_2(x)$  завжди більший за  $\frac{\alpha}{\beta}$ . Коли  $x$  прямує до  $x_m$  або  $x_M$ , обидві функції  $y_1(x)$  і  $y_2(x)$  прямують до  $\frac{\alpha}{\beta}$ . Отже, криві, визначені рівнянням (2.7), є замкненими для додатних  $x, y$  і мають вигляд, подібний до наведеного на рис. 2.9. Оскільки жодна з цих замкнених кривих (за винятком  $\bar{x} = \frac{\gamma}{\delta}$ ,  $\bar{y} = \frac{\alpha}{\beta}$ ) не містить точок рівноваги системи (2.6), то всі розв'язки  $x(t)$ ,  $y(t)$  системи (2.6), такі, що  $x(0) > 0$ ,  $y(0) > 0$ , є періодичними функціями часу, тобто для них виконуються рівності  $x(t+T) = x(t)$ ,  $y(t+T) = y(t)$  для деякого додатного числа  $T$ . Теорему доведено.

З попереднього бачимо, що положення рівноваги  $\bar{x}, \bar{y}$  є центром. Коливання чисельності відбуваються не в фазі: максимальному значенню  $x(t)$  відповідає середнє значення  $y(t)$  і

навпаки (на рис. 2.10 наведено графіки одного з розв'язків системи (2.6), побудовані з допомогою математичного пакета для конкретних значень параметрів).

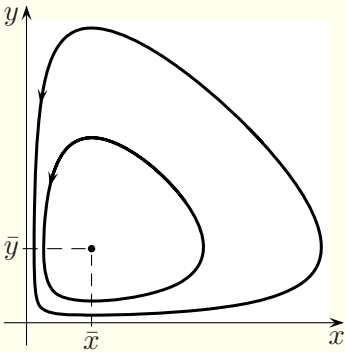


Рис. 2.9.

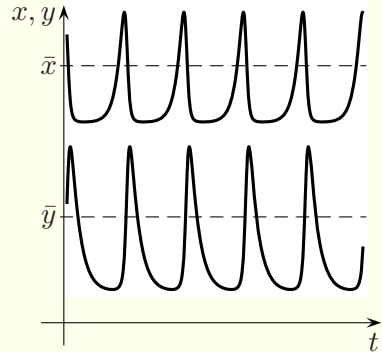


Рис. 2.10.

Якщо тепер припустити, що має місце промисловий вилов риби, то отримаємо систему диференціальних рівнянь

$$\begin{cases} \frac{dx}{dt} = \alpha x - \beta xy - cx, \\ \frac{dy}{dt} = -\gamma y + \delta xy - cy, \end{cases}$$

де  $c$  — додатний коефіцієнт вилову риби. Оскільки ловляться всі риби підряд, то коефіцієнт є тим самим в обидвох рівняннях. Для останньої системи положенням рівноваги буде точка  $\bar{x} = \frac{\gamma+c}{\delta}$ ,  $\bar{y} = \frac{\alpha-c}{\beta}$ , що означає зростання  $\bar{x}$  і зменшення  $\bar{y}$  (бо  $c > 0$ ). Отже, збільшення вилову риби приводить до зростання кількості промислової риби і зменшення кількості хижаків. Цей принцип називають принципом Вольтерри.

Аналогічну модель можна скласти і для інших систем взаємодії між хижаками і жертвами, наприклад, для систем коти — миші, вовки — зайці тощо.

У 1868 році з Австралії в США випадково завезли комаху-шкідника, який мало не знищив виробництво цитрусових. Оскільки в Австралії у нього є природний ворог — комаха,

що його їсть, довелося завести в Америку і цього хижака, внаслідок чого чисельність комах врегулювалась. У ХХ столітті проти шкідників застосували хімічні засоби, наслідком чого стало зростання кількості шкідників згідно з принципом Вольтерри. Тому від застосування отрутохімікатів довелося відмовитись.

Запропонована модель Лотки–Вольтерри лише наближено описує реальну систему. Про це свідчить, зокрема, те, що в реальних системах коливання чисельності хижаків і жертв звичайно не відбувається.

Узагальнена модель Лотки–Вольтерри має вигляд

$$\begin{cases} \frac{dx}{dt} = \alpha x - \beta xy - ax^2, \\ \frac{dy}{dt} = -\gamma y + \delta xy - by^2, \end{cases} \quad (2.8)$$

де  $a$  і  $b$  — коефіцієнти, які відповідають конкурентній боротьбі всередині популяції. Розв'язок системи (2.8) наближається до положення рівноваги по спіралі (рис. 2.11), тобто точка  $\bar{x}, \bar{y}$  — фокус.

Звичайно, і ця модель не описує всі випадки. По-перше, деколи (в унікальних, експериментальних випадках) хижаки можуть з'їсти всіх жертв, а потім померти з голоду. А такого розв'язку у цієї системи рівнянь бути не може. По-друге, якщо жертви можуть десь заховатись, то хижаки їх не з'їдять, поки ті не вийдуть зі свого сховища. Такі системи вже не є детермінованими і велику роль тут відіграє випадковий фактор (наприклад, можливість і вміння заховатись).

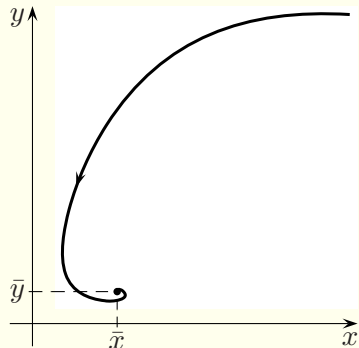


Рис. 2.11.

Для моделювання таких систем потрібно використовувати інші методи. У §§ 3.5, 3.6 розглядається моделювання систем взаємодії між хижа-

ками і жертвами з допомогою клітинних автоматів, причому результати моделювання є подібними до отриманих у цьому параграфі.

### § 2.7. Модель розвитку епідемії

Припустимо, що деяка популяція, яка налічує  $N$  особин, ділиться на три групи. У першу з них включаються здорові особини, що є сприйнятливими до деякої конкретної хвороби. Їхню кількість у момент часу  $t$  позначаємо через  $S(t)$ . У другу групу включаються хворі особини, які є джерелом інфекції для особин з першої групи. Кількість таких особин у популяції в момент часу  $t$  позначаємо через  $I(t)$ . До третьої групи відносять особин, що є здоровими і мають імунітет до хвороби. Їхню чисельність позначаємо через  $R(t)$ . Таким чином,

$$S(t) + I(t) + R(t) = N.$$

Всі хворі особини виліковуються й отримують імунітет до хвороби. Іншим способом отримати імунітет неможливо. Якщо кількість хворих особин не перевищує деякого числа  $I^*$ , то хворих можна ізолювати від здорових і припинити подальший розвиток епідемії. Якщо  $I(t) > I^*$ , то швидкість зміни кількості сприйнятливих до хвороби особин пропорційна їхній чисельності з коефіцієнтом захворюваності  $\alpha > 0$ . Швидкість зміни чисельності особин з імунітетом пропорційна кількості хворих особин з коефіцієнтом одужання  $\beta > 0$ . Початкову кількість хворих (в момент часу  $t = 0$ ) будемо позначати через  $I_0$ . Крім того, припускаємо, що в початковий момент часу особин з імунітетом не було. Всі ці припущення спрощують реальну ситуацію, але часто вони досить точно відображають сутність справи.

Складемо тепер диференціальну модель. Для  $I(t) \leq I^*$

маємо:

$$\begin{cases} \frac{dS}{dt} = 0, \\ \frac{dI}{dt} = -\beta I, \\ \frac{dR}{dt} = \beta I. \end{cases} \quad (2.9)$$

Для  $I(t) > I^*$  отримуємо:

$$\begin{cases} \frac{dS}{dt} = -\alpha S, \\ \frac{dI}{dt} = \alpha S - \beta I, \\ \frac{dR}{dt} = \beta I. \end{cases} \quad (2.10)$$

Початкові умови:  $S(0) = N - I_0$ ,  $I(0) = I_0$ ,  $R(0) = 0$ .

**Випадок 1.** Число  $I_0 \leq I^*$ . У цьому випадку маємо лише одну систему

$$\begin{cases} \frac{dS}{dt} = 0, \\ \frac{dI}{dt} = -\beta I, \end{cases}$$

що означає, що хворі ізольовані від здорових особин без імунітету до хвороби. Тоді

$$\begin{aligned} S(t) &= S(0) = N - I_0, \\ I(t) &= I_0 e^{-\beta t}, \\ R(t) &= N - S(t) - I(t) = I_0(1 - e^{-\beta t}). \end{aligned}$$

На рис. 2.12 графічно показано зміну кількості особин у кожній з трьох груп зі зростанням часу  $t$ .

**Випадок 2.**  $I_0 > I^*$ ,  $\alpha = \beta$ . Тоді має існувати інтервал  $0 \leq t < T$ , для всіх значень  $t$  якого справджується нерівність  $I(t) > I^*$ , бо за змістом задачі  $I$  як функція змінної  $t$  має бути неперервною. Отже, для всіх  $t \in [0, T)$  хвороба буде поширюватись на сприйнятливих до неї особин. Таким чином, для  $t \in [0, T)$

$$\begin{cases} \frac{dS}{dt} = -\alpha S, \\ \frac{dI}{dt} = \alpha S - \alpha I. \end{cases}$$

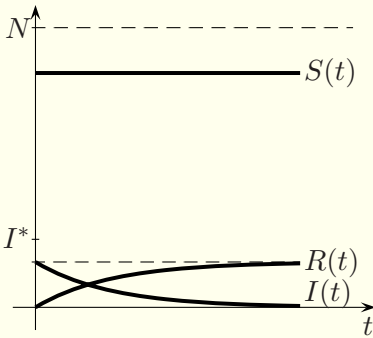


Рис. 2.12.

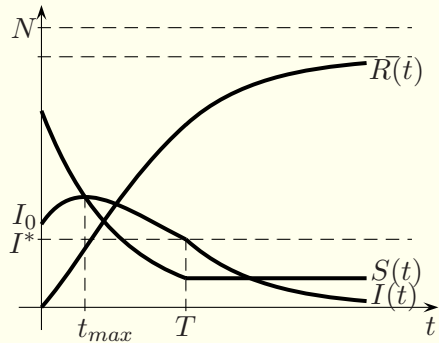


Рис. 2.13.

Тоді

$$S(t) = (N - I_0)e^{-\alpha t},$$

$$\frac{dI}{dt} + \alpha I = \alpha(N - I_0)e^{-\alpha t}. \quad (2.11)$$

Ми отримали лінійне диференціальне рівняння першого порядку. Застосуємо для його інтегрування метод варіації довільної сталої. Відповідне лінійне однорідне рівняння

$$\frac{dI}{dt} + \alpha I = 0$$

має загальний розв'язок  $I(t) = Ce^{-\alpha t}$ . Тоді розв'язок неоднорідного рівняння (2.11) шукаємо у вигляді  $I(t) = C(t)e^{-\alpha t}$ . Після підстановки в (2.11) маємо:

$$C'(t)e^{-\alpha t} = \alpha(N - I_0)e^{-\alpha t} \Rightarrow C(t) = \alpha(N - I_0)t + C,$$

звідки

$$I(t) = e^{-\alpha t}(\alpha(N - I_0)t + C).$$

Враховуючи початкову умову  $I(0) = I_0$ , отримуємо:

$$I(t) = e^{-\alpha t}(\alpha(N - I_0)t + I_0) \quad (2.12)$$

для  $t \in [0, T)$ .

Для того щоб знайти  $T$ , треба підставити в (2.12) замість  $I(t)$  значення  $I^*$ :

$$I^* = e^{-\alpha T} (\alpha(N - I_0)T + I_0).$$

Розв'язавши останнє трансцендентне рівняння при відомих числових значеннях параметрів числовими методами або з допомогою математичного пакета, можна знайти час  $T$ , коли припиниться розповсюдження хвороби.

Знайдемо тепер таке  $t_{\max}$ , при якому кількість хворих особин буде максимальною. Для дослідження функції  $I(t)$  на екстремум прирівнюємо до нуля її похідну

$$\begin{aligned} \frac{dI}{dt} = e^{-\alpha t} (\alpha(N - I_0) - \alpha^2(N - I_0)t - \alpha I_0) = 0 & \Rightarrow \\ t_{\max} = \frac{N - 2I_0}{\alpha(N - I_0)}. \end{aligned}$$

З виразу для похідної видно, що  $\frac{dI}{dt} < 0$  для  $t > t_{\max}$  і  $\frac{dI}{dt} > 0$  для  $t < t_{\max}$ . Тому в точці  $t = t_{\max}$  справді досягається максимум.

Якщо тепер підставити значення  $t_{\max}$  у рівність (2.12), то отримаємо:

$$I(t_{\max}) = e^{\frac{2I_0 - N}{N - I_0}} (N - I_0) = S(t_{\max}).$$

Остання рівність показує, зокрема, що у момент часу  $t_{\max}$  кількість сприйнятливих до хвороби особин збігається з кількістю хворих.

Якщо  $t > T$ , то сприйнятливі до хвороби особини вже не захворіють і  $I(t) = I^* e^{-\alpha(t-T)}$ .

На рис. 2.13 графічно показано, як з плином часу  $t$  змінюватиметься кількість особин у кожній з трьох груп.

**Випадок 3.**  $I_0 > I^*$ ,  $\alpha \neq \beta$ . Цей випадок доводиться розглядати окремо від попереднього, бо тут виходять дещо інші розв'язки диференціальних рівнянь. Нехай  $0 \leq t < T -$

інтервал, для всіх значень  $t$  якого справджується нерівність  $I(t) > I^*$ . Для  $t \in [0, T)$

$$\begin{cases} \frac{dS}{dt} = -\alpha S, \\ \frac{dI}{dt} = \alpha S - \beta I. \end{cases}$$

Тоді

$$\begin{aligned} S(t) &= (N - I_0)e^{-\alpha t}, \\ \frac{dI}{dt} + \beta I &= \alpha(N - I_0)e^{-\alpha t}. \end{aligned} \quad (2.13)$$

Ми отримали лінійне диференціальне рівняння першого порядку. Застосуємо для його інтегрування метод варіації довільної сталої. Відповідне лінійне однорідне рівняння

$$\frac{dI}{dt} + \beta I = 0$$

має загальний розв'язок  $I(t) = Ce^{-\beta t}$ . Тоді розв'язок неоднорідного рівняння (2.13) шукаємо у вигляді  $I(t) = C(t)e^{-\beta t}$ . Після підстановки в (2.13) маємо:

$$C'(t)e^{-\beta t} = \alpha(N - I_0)e^{-\alpha t} \quad \Rightarrow \quad C(t) = \frac{\alpha(N - I_0)}{\beta - \alpha}e^{(\beta - \alpha)t} + C,$$

звідки

$$I(t) = \frac{\alpha(N - I_0)}{\beta - \alpha}e^{-\alpha t} + Ce^{-\beta t}.$$

Враховуючи початкову умову  $I(0) = I_0$ , отримуємо:

$$I(t) = \frac{\alpha(N - I_0)}{\beta - \alpha}e^{-\alpha t} + \frac{\beta I_0 - \alpha N}{\beta - \alpha}e^{-\beta t} \quad (2.14)$$

для  $t \in [0, T)$ .

Для того щоб знайти  $T$ , треба підставити в (2.14) замість  $I(t)$  значення  $I^*$ :

$$I^* = \frac{\alpha(N - I_0)}{\beta - \alpha}e^{-\alpha T} + \frac{\beta I_0 - \alpha N}{\beta - \alpha}e^{-\beta T}.$$

Розв'язавши останнє трансцендентне рівняння при відомих числових значеннях параметрів числовими методами або з допомогою математичного пакета, можна знайти час  $T$ , коли припиниться розповсюдження хвороби.

Як і у випадку 2 можна було б знайти точку максимуму  $t_{\max}$  функції  $I(t)$ , проте для різних чисел  $\alpha$  і  $\beta$  не буде виконуватись рівність  $I(t_{\max}) = S(t_{\max})$ .

Якщо  $t > T$ , то сприйнятливі до хвороби особи вже не захворіють і  $I(t) = I^*e^{-\beta(t-T)}$ .

На рис. 2.14 для  $\alpha < \beta$  графічно показано зміну кількості особин у кожній з трьох груп зі зростанням часу  $t$ .

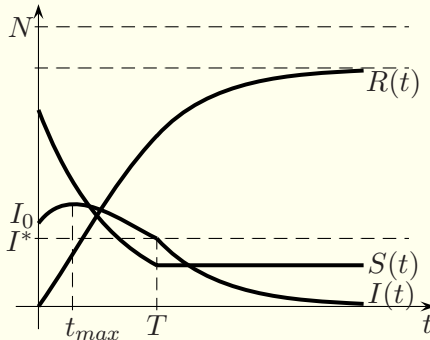


Рис. 2.14.

## § 2.8. Прогин балок

Розглянемо горизонтально розташовану балку сталого поперечного перерізу, зроблену з однорідного матеріалу (рис. 2.15). Її вісь симетрії позначено пунктиром. Ця балка може прогинатись під дією різних сил, що діють у вертикальній площині, а також ваги самої балки (рис. 2.16). Зрозуміло, що під дією цих сил вісь симетрії балки буде викривлятись. Викривлену вісь симетрії називають *пружною лінією* балки. Визначення форми цієї лінії відіграє велику роль у теорії пружності.

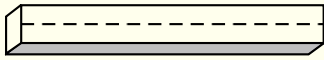


Рис. 2.15.



Рис. 2.16.

*Нейтральним шаром* балки називають той шар, в якому волокна балки не розтягаються і не стискаються під дією тих сил, що зумовлюють прогин балки. Вісь симетрії і пружна лінія балки проходять через нейтральний шар балки.

Існують різні типи балок в залежності від способів їхнього кріплення або опори. Балка, в якій один кінець жорстко закріплений, а інший — вільний, називається *консольною балкою* (рис. 2.17). Балка з рис. 2.18 вільно лежить на двох опорах. Бувають також інші типи балок (рис. 2.19).

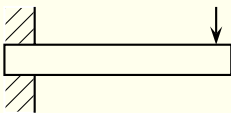


Рис. 2.17.

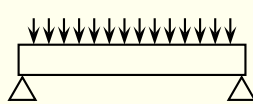


Рис. 2.18.

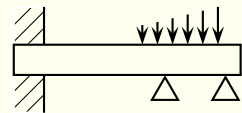


Рис. 2.19.

Існують також різні способи прикладання сил до балки. Сила може бути, наприклад, зосередженою в точці (рис. 2.17), рівномірно розподіленою (рис. 2.18), змінною (рис. 2.19).

Для горизонтальної балки вісь абсцис проведемо горизонтально через невикривлену вісь симетрії балки, а вісь ординат спрямуємо вертикально вниз так, щоб лівий кінець балки розміщувався у початку координат (рис. 2.20).

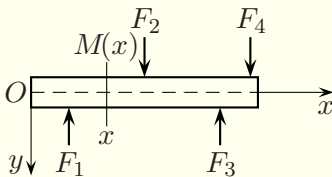


Рис. 2.20.

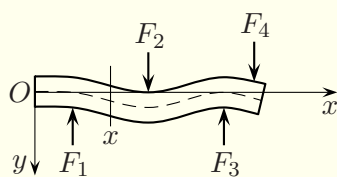


Рис. 2.21.

Під дією зовнішніх сил  $F_1, F_2, F_3, F_4$  та інших вісь симетрії викривляється в пружну лінію. Зміщення пружної лінії

від осі абсцис називають прогином балки в точці  $x$ . Таким чином, якщо відомим є рівняння пружної лінії балки, то завжди можна знайти прогин балки у кожній точці. Розглянемо тепер відшукання рівняння пружної лінії балки.

Позначимо через  $M(x)$  згинальний момент у вертикальному перерізі балки з координатою  $x$ . *Згинальний момент* є алгебричною сумою моментів всіх сил, що діють на балку з одного боку (справа або зліва) від точки  $x$ . При цьому моменти сил, які діють зверху вниз, беруться зі знаком «плюс», а моменти сил, які діють знизу вверх, беруться зі знаком «мінус». *Момент сили* — це добуток сили на відстань від точки прикладання сили до точки  $x$ .

У курсі опору матеріалів доводиться, що кривизна  $K(x)$  пружної лінії пов'язана зі згинальним моментом  $M(x)$  співвідношенням

$$K(x) = \frac{M(x)}{E(x)J(x)},$$

де  $E(x)$  — модуль пружності Юнга, який залежить від матеріалу балки, а  $J(x)$  — геометричний момент інерції поперечного перерізу балки відносно *нейтральної осі* — прямої перетину нейтрального шару балки з площиною цього перерізу. Нейтральна вісь є горизонтальною прямою, що проходить через центр ваги поперечного перерізу балки (рис. 2.22). Якщо поперечний переріз балки є однаковим у різних точках, а балка складається з одного матеріалу, то величини  $E$  і  $J$  є сталими.

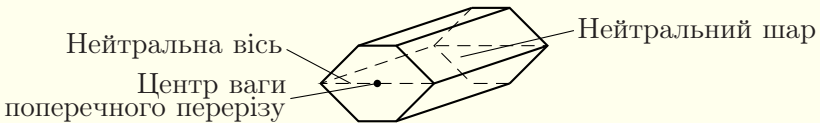


Рис. 2.22.

З математичного аналізу відомо, що кривизна  $K(x)$  подається формулою

$$K(x) = \frac{y''}{(1 + y'^2)^{\frac{3}{2}}},$$

де  $y(x)$  — відхилення пружної лінії від прямої в точці  $x$ . Якщо балка лише злегка прогинається, що часто буває на практиці, то кутовий коефіцієнт  $y'$  пружної лінії є дуже малим і ним можна знехтувати. Тоді отримуємо диференціальне рівняння пружної лінії балки:

$$y'' = \frac{M(x)}{EJ}. \quad (2.15)$$

З математичного аналізу відомо, що момент інерції  $J$  обчислюється за формулою

$$J = I_x = \iint_{(D)} y^2 dx dy, \quad (2.16)$$

де  $(D)$  — поперечний переріз балки, центр ваги якого знаходиться у початку координат. Момент інерції залежить лише від поперечного перерізу, тобто форми балки і не залежить від матеріалу.

**Приклад.** Вільний кінець консольної сталеві балки довжиною  $l = 4$  м навантажений зосередженою силою  $P = 20000$  Н. Знайти рівняння пружної лінії і визначити величину прогину вільного кінця балки, якщо балка має сталий поперечний переріз, який є прямокутником з шириною 3 см і висотою 20 см. Вагою самої балки знехтувати. Модуль пружності Юнга для сталі  $E = 2,1 \cdot 10^{11}$  Н/м<sup>2</sup>.

**Розв'язання.** Спочатку знаходимо момент інерції  $J$ . Поперечний переріз з центром системи координат у центрі його ваги зображено на рис. 2.23. Користуючись формулою (2.16), маємо:

$$J = \int_{-0,1}^{0,1} dy \int_{-0,015}^{0,015} y^2 dx = \frac{4}{3} \cdot 0,1^3 \cdot 0,015 = 2 \cdot 10^{-5} \text{ м}^4.$$

На рис. 2.24 схематично зображено консольну балку з прикладеною до неї силою  $P$ . Визначаємо згинальний момент

$M(x)$  для довільної точки  $x \in (0, l)$ . У даному випадку  $M(x)$  дорівнює моменту сили  $P$  відносно точки  $x$ , взятому зі знаком «плюс», тобто

$$M(x) = P(l - x).$$

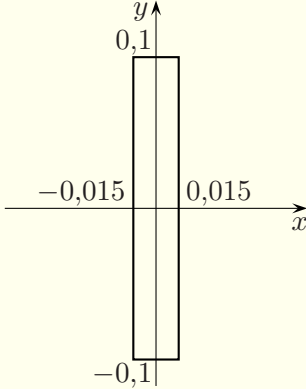


Рис. 2.23.

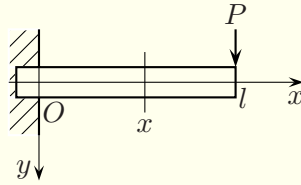


Рис. 2.24.

Отже, згідно з формулою (2.15) отримуємо диференціальне рівняння

$$y'' = \frac{P}{EJ}(l - x). \quad (2.17)$$

Умова закріплення лівого кінця консольної балки дає початкові умови

$$y(0) = 0, \quad y'(0) = 0. \quad (2.18)$$

Безпосередньо інтегруючи диференціальне рівняння (2.17), знаходимо його загальний розв'язок:

$$\begin{aligned} y' &= -\frac{P}{EJ} \frac{(l-x)^2}{2} + C_1, \\ y &= \frac{P}{EJ} \frac{(l-x)^3}{6} + C_1 x + C_2. \end{aligned}$$

Враховуючи початкові умови (2.18), отримуємо:

$$\begin{cases} \frac{P}{EJ} \frac{l^3}{6} + C_2 = 0, \\ -\frac{P}{EJ} \frac{l^2}{2} + C_1 = 0 \end{cases} \Rightarrow C_1 = \frac{Pl^2}{2EJ}, \quad C_2 = -\frac{Pl^3}{6EJ}.$$

Тому рівняння пружної лінії має вигляд

$$y = \frac{P}{2EJ} \left( lx^2 - \frac{x^3}{3} \right).$$

Величина прогину  $h$  в кінці балки отримується з останньої рівності при  $x = l$ :

$$h = \frac{P}{2EJ} \left( l^3 - \frac{l^3}{3} \right) = \frac{Pl^3}{3EJ} = \frac{20000 \cdot 4^3}{3 \cdot 2,1 \cdot 10^{11} \cdot 2 \cdot 10^{-5}} \approx 0,1 \text{ м.} \blacksquare$$

## Питання до розділу 2

1. Що таке диференціальна модель? Якої послідовності дій дотримуються при її побудові?
2. Як визначити час витікання рідини з посудини через невеликий отвір у дні?
3. Наведіть диференціальну модель руху частинок у трубці.
4. У чому полягає диференціальна модель Лотки-Вольтерри? Що вона описує? Наведіть її узагальнення.
5. Наведіть диференціальну модель розвитку епідемії при можливості ізоляції хворих.
6. Як скласти диференціальне рівняння пружної лінії балки? Чому дорівнює згинальний момент?

## Вправи до розділу 2

1. Круглий циліндричний чан з вертикальною віссю діаметром  $D$  і висотою  $H$  заповнений водою. За який час вся вода витече крізь круглий отвір діаметром  $a$  у дні чана?
2. Посудину, що має форму півкулі радіусом 2 м, заповнено водою. За який час витече вся вода крізь круглий отвір радіусом 0,1 м, вирізаний у дні посудини?
3. Встановлений вертикально чан циліндричної форми має отвір у дні. Половина води з повного чана витікає за 5 хвилин. За який час витече вся вода?
4. Швидкість розмноження деяких бактерій пропорційна кількості бактерій у розглядуваний момент часу. Кількість бактерій подвоюється протягом трьох годин. Знайти: а) залежність кількості

бактерій від часу; б) у скільки разів збільшиться кількість бактерій протягом 9 годин.

5. Швидкість розпаду радію пропорційна наявній його кількості. З досвіду відомо, що протягом року вага одного грама радію меншає на 0,435 мг. Визначити період піврозпаду (час, протягом якого початкова кількість зменшиться вдвоє).

6. Швидкість охолодження тіла в повітрі пропорційна різниці між температурою тіла та температурою повітря. Температура повітря  $20\text{ }^{\circ}\text{C}$ , тіло протягом 20 хвилин охолоджується від  $100\text{ }^{\circ}\text{C}$  до  $60\text{ }^{\circ}\text{C}$ . Знайти залежність температури від часу та через який час температура тіла знизиться до  $30\text{ }^{\circ}\text{C}$ .

7. Кількість світла, що поглинається під час проходження крізь тонкий шар води, пропорційна товщині шару й кількості світла, що падає на його поверхню. Якщо під час проходження крізь шар завтовшки 3 м поглинається половина початкової кількості світла, то яка частина цієї кількості дійде до глибини 30 м?

8. Електровоз рухається по горизонтальній залізничній колії зі швидкістю 72 км/год. Машиніст включає гальма, опір руху після початку гальмування дорівнює 0,2 ваги електровоза. Знайти час від моменту початку гальмування до повної зупинки електровоза і пройдений за цей час шлях.

9. Тіло кинуто під кутом  $\alpha$  до горизонту з початковою швидкістю  $v_0$  (наприклад, здійснено постріл з гармати). Виведіть рівняння руху тіла, нехтуючи опором повітря. З аналізу рівняння руху дайте відповідь на поставлені запитання: а) яким є час польоту тіла; б) як далеко від початкової точки тіло впаде на землю; в) при якому куті  $\alpha$  тіло пролетить найбільшу відстань; г) якою є максимальна висота підйому тіла; д) якою є траєкторія польоту тіла?

10. Деякі бактерії розмножуються пропорційно їхній кількості, але в той самий час продукти їхньої життєдіяльності знищують їх пропорційно кількості бактерій і продуктів їхньої життєдіяльності. Швидкість утворення продуктів життєдіяльності теж є пропорційною кількості бактерій. Довести, що кількість бактерій  $N$  спочатку зростає до деякого максимального значення  $M$ , а потім спадає до нуля; у момент часу  $t$  вона подається формулою  $N = M(1 - th^2kt)$ , де час  $t$  змінюється від моменту, коли  $N = M$ .

## Розділ 3. Скінченні автомати

### § 3.1. Скінченні автомати Мілі

Взагалі кажучи, автоматом називають пристрій для прийому, перетворення чи передачі речовини, інформації або енергії, який працює за певною програмою без безпосередньої участі людини.

*Теорія автоматів* — це розділ математичної кібернетики, в якому вивчаються математичні моделі — автомати.

*Автомат Мілі* для кожного символу  $x$  з вхідного алфавіту  $X$  і кожного стану  $q$  з множини допустимих станів  $Q$  ставить у відповідність деякий символ  $y$  з вихідного алфавіту  $Y$  і деякий стан  $q'$  з множини станів  $Q$ . У випадку скінченних множин  $X$ ,  $Q$  і  $Y$  автомат Мілі називають скінченним.

Таким чином, *скінченний автомат Мілі* — це п'ятірка  $A = (X, Q, Y, q_0, F)$ , де  $X = \{x\}$  — скінченний вхідний алфавіт,  $Y = \{y\}$  — скінченний вихідний алфавіт,  $Q = \{q\}$  — скінченна множина внутрішніх станів,  $q_0$  — початковий стан,  $F : X \times Q \rightarrow Y \times Q$  — функція переходів. Функція переходів може бути наочно подана діаграмою станів. Деколи функцію переходів подають у матричній формі.

Скінченні автомати Мілі є дискретно-детермінованими моделями систем. У випадку автомата Мілі глобальні рівняння системи (1.1), (1.2) з § 1.1 подаються у вигляді:

$$\begin{aligned}z(t+1) &= f(x(t), z(t)), \quad t = 0, 1, 2, \dots, \\y(t) &= g(x(t), z(t)), \quad t = 0, 1, 2, \dots\end{aligned}$$

**Приклад.** Побудувати скінченний автомат Мілі для керування ліфтом у двоповерховому будинку.

**Розв'язання.** Вхідний алфавіт автомата складається з кнопок виклику відповідного поверху:  $X = \{C1, C2\}$ ; вихідний алфавіт складається зі зміщень на один поверх вгору або

вниз, а також зупинки ліфта:  $Y = \{U1, D1, S\}$ ; стан відповідає поверху, на якому знаходиться ліфт:  $Q = \{q_1, q_2\}$ . Якщо спочатку ліфт перебуває на першому поверсі, то початковим станом є стан  $q_1$ . Функція переходів цього автомата зручно подається діаграмою станів (рис. 3.1). Лінії зі стрілками визначають переходи зі стану в стан, кожен з написів біля цих ліній містить кнопку виклику відповідного поверху з вхідного алфавіту і вихідний сигнал для зміщення чи зупинки ліфта, відокремлені вертикальною рискою.

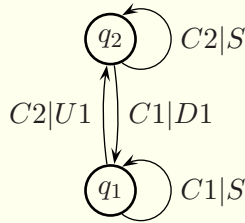


Рис. 3.1

### § 3.2. Скінченні автомати Мура

Є також інше означення автомата, в якому функція переходів задає наступний стан, а вихід автомата залежить лише від його поточного стану. Такі автомати називають автоматами Мура.

*Автомат Мура* для кожного символу  $x$  з вхідного алфавіту  $X$  і кожного стану  $s$  з множини допустимих станів  $S$  ставить у відповідність деякий стан  $s'$  з множини станів  $S$ , який однозначно визначає символ  $y$  вихідного алфавіту  $Y$ . У випадку скінченних множин  $X$ ,  $S$  і  $Y$  автомат Мура називають скінченним.

Таким чином, *скінченний автомат Мура* — це шістка  $B = (X, S, Y, s_0, P, R)$ , де  $X = \{x\}$  — скінченний вхідний алфавіт,  $Y = \{y\}$  — скінченний вихідний алфавіт,  $S = \{s\}$  — скінченна множина внутрішніх станів,  $s_0$  — початковий стан,

$P : X \times S \rightarrow S$  — функція переходів,  $R : S \rightarrow Y$  — функція виходів.

У випадку автомата Мура глобальні рівняння системи (1.1), (1.2) з § 1.1 подаються у вигляді:

$$\begin{aligned}z(t+1) &= f(x(t), z(t)), \quad t = 0, 1, 2, \dots, \\y(t) &= g(z(t)), \quad t = 0, 1, 2, \dots\end{aligned}$$

**Приклад.** Скласти автомат Мура для керування автоматом для продажу кави. Вартість кави — 5 грн. Автомат приймає монети номіналом в одну, дві і п'ять гривень, всі інші монети ігноруються. Монети приймаються по черзі. Після накопичення достатньої суми автомат видає каву. Автомат решту не дає.

**Розв'язання.** Вхідний алфавіт автомата Мура складається з чисел 1, 2 і 5 — номіналів монет, а вихідний — з чисел 0 (сума недостатня) і 1 (сума достатня). Можливими станами є  $s_0, s_1, s_2, s_3, s_4, s_5$ , які відповідають накопиченій сумі. Якщо накопичена сума перевищує 5, наприклад,  $2 + 2 + 2$ , то можна вважати, що досягнутий стан  $s_5$ . Діаграму станів зображено на рис. 3.2. Функція виходу для станів  $s_0, s_1, s_2, s_3, s_4$  набуває значення 0, а для стану  $s_5$  вона дорівнює 1. ■

Автомати Мілі і Мура є еквівалентними у тому сенсі, що кожен автомат Мілі може бути перетворений у відповідний йому автомат Мура і навпаки. При перетворенні автомата Мілі в автомат Мура часто доводиться збільшувати кількість внутрішніх станів системи, тому множини  $Q$  і  $S$ , взагалі кажучи, є різними. Функції переходів  $F$  і  $P$  теж є різними.

Перетворимо автомат Мілі з прикладу в § 3.1 в автомат Мура. Для цього розширимо множину внутрішніх станів до чотирьох. Діаграма станів наведена на рис. 3.3, причому для кожного стану  $s_j$  автомата Мура під ризикою вказується відповідний стан  $q_k$  автомата Мілі і вихідний сигнал.

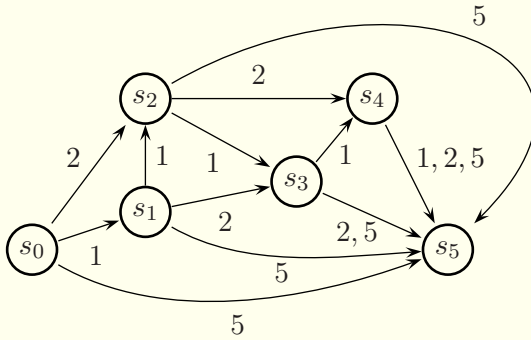


Рис. 3.2

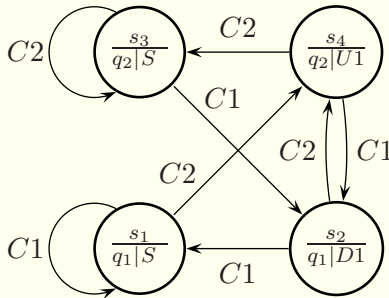


Рис. 3.3

З точки зору зручності побудови простішим є автомат Мілі. Автомат Мура широко використовують у схемотехніці для синтезу реальних обчислювальних і логічних пристроїв. У таких пристроях зберігається і модифікується стан, а вихідний сигнал видається назовні як результат.

### § 3.3. Клітинні автомати

Клітинні автомати є окремим випадком скінченних автоматів. Їх використовують для моделювання динамічної поведінки однорідних двовимірних і тривимірних середовищ у дискретному просторі і часі. При цьому всі фізичні характеристики теж змінюються дискретно. Існує досить розвинена теорія

клітинних автоматів. В її основу лягли праці Джона фон Неймана, який у 1948 році ввів поняття клітинного автомата при розробці комп'ютерної моделі біологічної системи.

При створенні клітинних автоматів весь простір розбивається регулярною ґраткою на таблицю комірок, які називають *клітинками*. Кожна з клітинок може знаходитись в одному зі скінченної кількості станів. Кожній клітинці ставиться у відповідність *клітинний автомат*. Найважливішими особливостями клітинних автоматів є наступні:

- стан кожної клітинки оновлюється в результаті виконання послідовності дискретних кроків у часі (тактів);
- змінні стани у кожній клітинці змінюються одночасно (синхронно) відповідно до значень змінних станів цієї клітинки і клітинок деякого її околу на попередньому кроці.

Клітинки для клітинних автоматів на площині найчастіше мають форму квадрата, рідше — трикутника чи шестикутника. Сукупність клітинок утворює клітинний простір.

За окіл клітинки беруть певну кількість сусідніх клітинок. Зокрема, для квадрата околом можуть бути чотири або вісім сусідніх клітинок (рис. 3.4). Однак можливі ситуації, коли стан клітинки залежить від стану більш віддалених клітинок.

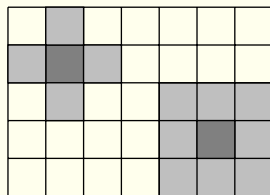


Рис. 3.4

Для того щоб виключити особливий вигляд околу для точок, що лежать на межі області, часто вводять так звані *крайові умови циклічного типу*. Вони означають, що для крайнього лівого стовпця клітинок сусіднім зліва вважається крайній правий стовпець, а для крайнього правого — крайній лівий стовпець. Аналогічно, для крайнього верхнього рядка клітинок сусіднім зверху вважається крайній нижній рядок, і навпаки. Таким чином, можна вважати, що клітинки задаються на поверхні тора.

Функцію переходів для клітинних автоматів називають

*правилами* клітинних автоматів.

Найвідомішим клітинним автоматом є автомат «Життя», розроблений у 1970 році Джоном Конвеєм. Множина правил для цього автомата полягає у наступному:

- клітинка може перебувати у двох станах — активному і пасивному;
- околom є вісім сусідніх клітинок;
- якщо в околі пасивної клітинки є рівно три активних клітинки, то клітинка стає активною («народжується»);
- якщо в околі активної клітинки є дві чи три активних клітинки, то клітинка залишається активною («виживає»);
- якщо в околі активної клітинки є менше двох чи більше трьох активних клітинок, то вона стає пасивною («помирає» від самотності чи перенаселення).

У початковий момент часу є деякий розподіл активних і пасивних клітинок. Використовуючи наявний розподіл клітинок, визначаємо стан клітинок на наступному кроці і т. д. В залежності від вигляду початкового розподілу приходимо до різних еволюцій життя.

Розглянемо приклад. Для початкового розподілу активних клітинок автомата «Життя» з рис. 3.5 на рис. 3.6–3.8 зображено розподіли активних клітинок для четвертого, восьмого і одинадцятого кроків відповідно. На дванадцятому і всіх наступних кроках зберігатиметься розподіл активних клітинок з рис. 3.8.

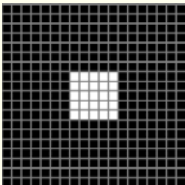


Рис. 3.5

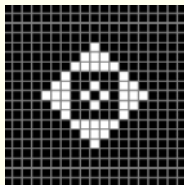


Рис. 3.6

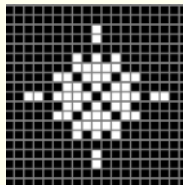


Рис. 3.7

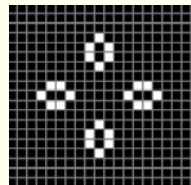


Рис. 3.8

### § 3.4. Застосування клітинних автоматів до моделювання активних середовищ

Клітинні автомати можна застосовувати для імітаційного моделювання поведінки так званих активних середовищ. Активні середовища характеризуються надходженням енергії від зовнішнього джерела і її розсіянням. Завдяки тому, що через кожен фізично малий елемент середовища протікає потік енергії від джерела, цей елемент може виводитись зі стану рівноваги і отримувати здатність виконувати автоколивання, бути бістабільним чи збудливим. Ці елементи можуть утворювати в середовищі різні стаціонарні або залежні від часу просторові структури. Описані процеси лежать в основі явищ самоорганізації, що відбуваються в активних середовищах. Прикладами активних середовищ є деякі розчини зі специфічними хімічними реакціями, нервові тканини, процеси горіння.

Виділяють три типи найпростіших активних елементів: бістабільні, збудливі й автоколивні.

*Бістабільний елемент* має два стійких стани і переходить з одного стану в інший, якщо зовнішні впливи перевищують деяке порогове значення. Одним з варіантів бістабільного елемента можна вважати клітинний автомат «Життя».

*Збудливий елемент* має кілька станів, але стійким до достатньо слабких впливів є лише один з них. Якщо вплив перевищує деякий поріг, то збудливий елемент переходить зі стійкого стану в один з нестійких станів через проміжні нестійкі стани і лише потім поступово повертається до початкового стійкого стану. Приклад середовища зі збудливих елементів — нервова тканина. Якщо подразнення перевищують деякий рівень, то нервова клітина переходить у збуджений стан, створюючи в свою чергу подразнення для з'єднаних з нею клітин і переводячи їх у збуджений стан. Витративши енергію на збудження сусідів, клітина переходить у стан відновлення, коли вона може сприймати нові подразнення і збуджуватись. Потім

клітина повертається до початкового стану спокою, готова до передачі нових імпульсів.

*Автоколивний елемент* працює подібно «вічному двигуну», здійснюючи циклічні переходи з одного стану в інший. Зовнішні впливи не можуть зупинити ці переходи, але можуть їх прискорити чи сповільнити.

Поряд зі скінченими детермінованими автоматами можна розглядати скінченні ймовірнісні автомати, які вводяться подібно до детермінованих. Важливим класом скінчених ймовірнісних автоматів є клітинні автомати з випадковими процесами. Такі клітинні автомати дозволяють моделювати більш складні системи.

Клітинні автомати можуть використовуватись для моделювання поширення епідемій, лісових пожеж, дефектів у металах, біологічних процесів, фазових переходів, динаміки рідин тощо.

### § 3.5. Модель розвитку організмів на поживних середовищах

Одним з найцікавіших застосувань клітинних автоматів є моделювання біологічних систем.

Розглянемо приклад моделювання біологічної системи з допомогою клітинних автоматів. Біологічною системою є двовимірне поживне середовище з одноклітинними чи іншими живими організмами, які на ньому розвиваються. Модель опишемо сукупністю правил:

1. Простір клітинних автоматів є полем  $n \times n$  квадратних клітинок.

2. Окіл клітинки утворюють вісім сусідніх клітинок.

3. Кожній клітинці відповідає значення  $p$  ступеня поживності (енергоємності) розчину, яке може змінюватись від 0 до  $p_{max}$ .

4. Величина поживності (енергоємності) розчину клітинки за такт часу збільшується на величину  $\Delta p$ , але не може

перевищувати  $p_{max}$ .

5. Клітинка може бути вільною або містити один живий організм.

6. Організм бере енергію з поживного розчину клітинки, в якій він знаходиться, зменшуючи її поживність і збільшуючи свій запас енергії на величину  $\Delta E_{pl}$  за такт, але при цьому не може взяти енергії з клітинки більше, ніж її там є.

7. Максимальна можлива кількість енергії, яку може накопичити організм, не перевищує  $E_{max}$ .

8. Наприкінці кожного такту організм намагається перейти в сусідню вільну клітинку, вибираючи її довільним чином. Якщо вільної клітинки в околі немає, то організм залишається в тій самій клітинці.

9. Починаючи з віку  $L_{zr}$  тактів, організм вважається зрілим. Протягом такту до досягнення зрілості організм витрачає на життя  $\Delta E_{mi}$  енергії, а після досягнення зрілості —  $\Delta E_{zrmi}$  енергії.

10. Зрілий організм може створювати собі подібних діленням, витрачаючи додатково  $\Delta E_{di}$  енергії при кожному поділі. При цьому батьківська особина переходить у сусідню вільну клітинку, а нова залишається у попередній клітинці з початковим запасом енергії  $E_0$ . Якщо в околі клітинки вільних клітинок немає або запас енергії організму не перевищує  $\Delta E_{di}$ , то ділення не відбувається.

11. Час життя окремої особини не перевищує  $L_{max}$  тактів. Якщо час життя організму досягає тривалості життя  $L_{max}$  або запас енергії знижується до нуля, то організм помирає.

12. Початкова поживність клітинок задається рівномірним законом розподілу в межах від 0 до  $p_{max}$ . Загальну поживність всіх клітинок на такті  $t$  позначимо через  $P_A(t)$ , тоді відносна поживність  $P(t) = \frac{P_A(t)}{n^2 p_{max}}$ .

13. Початковий закон розподілу організмів у клітинному просторі є рівномірним. Початкова кількість організмів —  $N_0$ . Початковий запас енергії кожної особини дорівнює  $E_0$ , а вік

вважається нульовим. Загальну кількість організмів на такті  $t$  позначимо через  $N_A(t)$ , тоді відносна чисельність організмів  $N(t) = \frac{N_A(t)}{n^2}$ .

Запрограмувати ці правила будь-якою мовою програмування нескладно. У результаті можна отримати залежності  $P$  і  $N$  від часу та фазову траєкторію (залежність  $N$  від  $P$ ). Наприклад, у випадку, коли  $n = 1000$ ,  $p_{max} = 10$ ,  $\Delta p = 1$ ,  $\Delta E_{pl} = 6$ ,  $E_{max} = 30$ ,  $\Delta E_{mi} = 3$ ,  $\Delta E_{zrmi} = 2$ ,  $\Delta E_{di} = 4$ ,  $E_0 = 4$ ,  $L_{max} = 20$ ,  $L_{zr} = 4$ ,  $N_0 = 0,3n^2$ , графіки цих залежностей наведені на рис. 3.9–3.11, а фрагмент можливого розміщення організмів — на рис. 3.12.

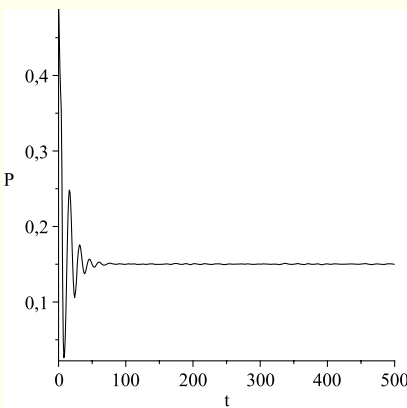


Рис. 3.9

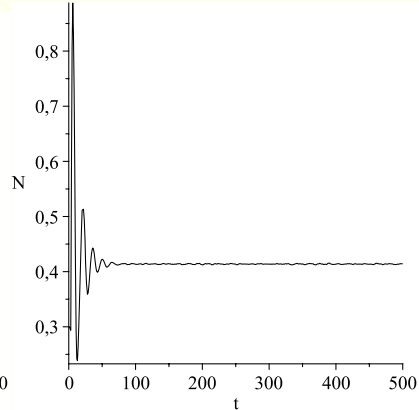


Рис. 3.10

Бачимо, що мають місце згасаючі коливання відносного рівня поживності середовища і відносної чисельності організмів навколо їхніх середніх значень  $\bar{P} \approx 0,15$ ,  $\bar{N} \approx 0,41$ . Крім того, можна встановити, що середня тривалість життя організмів становить  $\bar{L} \approx 4,7$  тактів, хоча є особини, які доживають до глибокої старості (20 тактів). Згасаючі коливання виявляються подібними до тих, що виникають при розв'язуванні модифікованої системи диференціальних рівнянь Лотки–Вольтерри з конкурентною боротьбою (§ 2.6).

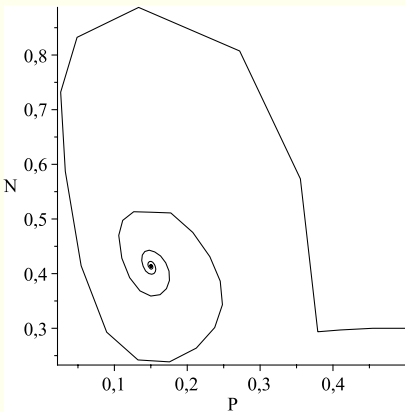


Рис. 3.11

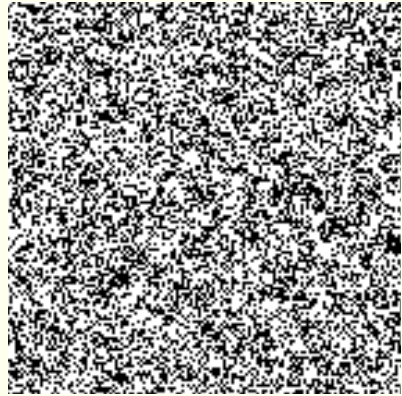


Рис. 3.12

Модель можна модифікувати наступним чином: організм завжди намагається перейти у сусідню вільну клітинку з найбільшим рівнем енергоємності. Якщо клітинки в околі мають менший запас енергії, то особина залишається у тій самій клітинці. Крім того, при діленні організм вибирає сусідню вільну клітинку з найбільшим рівнем енергоємності. У цьому випадку для тих самих числових даних отримуємо графіки, наведені на рис. 3.13–3.15, і розміщення організмів з рис. 3.16.

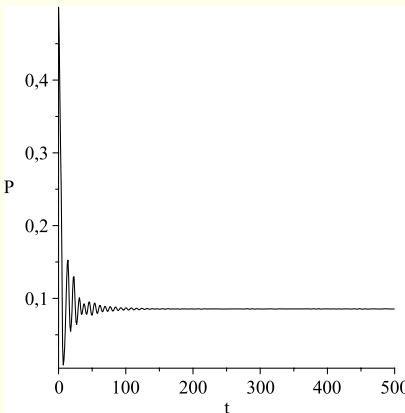


Рис. 3.13

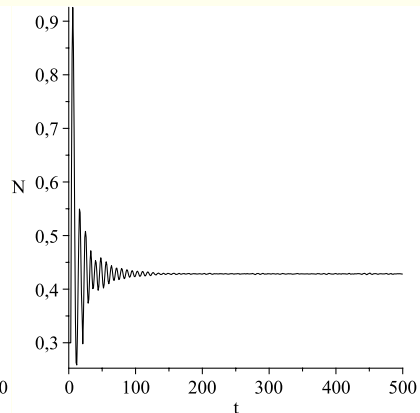


Рис. 3.14

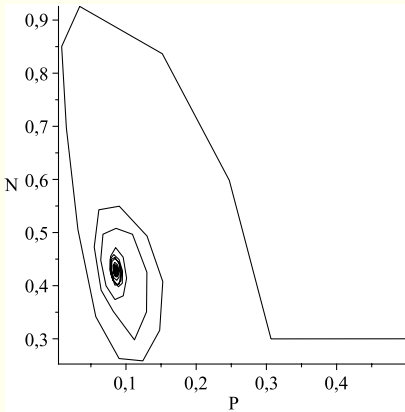


Рис. 3.15

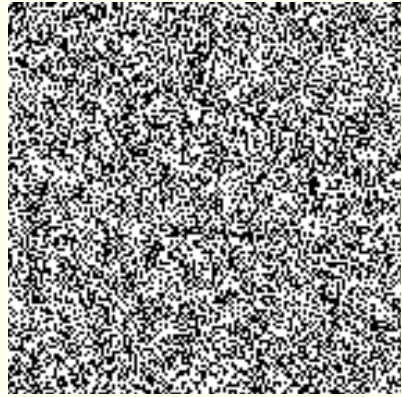


Рис. 3.16

Бачимо, що у цьому випадку спочатку відбуваються згасаючі коливання відносного рівня поживності середовища і відносної чисельності організмів навколо їхніх середніх значень  $\bar{P} \approx 0,085$ ,  $\bar{N} \approx 0,43$ , після чого відносний рівень поживності середовища і відносна чисельність організмів досягають своїх середніх значень. Крім того, середня чисельність організмів у цьому випадку трохи більша, середня тривалість їхнього життя більша ( $\bar{L} \approx 6,6$ ), а середня поживність середовища значно менша, ніж у випадку «нерозумної» поведінки організмів. Таким чином, «розумна» поведінка сприяє виживанню організмів.

### § 3.6. Модель розвитку організмів на поживних середовищах за наявності хижаків

Додамо до попередньої моделі хижаків, життя яких підпорядковане наступним правилам:

1. Клітинка може бути вільною, містити один звичайний організм або одного хижака.

2. Їжею для хижаків є живі організми з попередньої моделі. Хижак може напасти на свою жертву, якщо вона знаходиться в його околі, тобто в одній з восьми сусідніх клітинок.

Якщо є кілька потенційних жертв, то одна з них вибирається довільним чином. При цьому жертва гине з імовірністю  $P_X$ , а хижак переміщується в її клітинку, підвищуючи свій запас енергії на величину енергетичної цінності жертви  $\Delta EX_{pl}$ . Імовірність загибелі жертви не дорівнює одиниці, бо вважається, що вона має деякі засоби захисту. Якщо жертва залишається живою, то гине хижак.

3. Якщо в околі хижака немає потенційних жертв, то хижак переміщується на сусідню вільну клітинку, вибираючи напрям довільним чином.

4. Інші характеристики життя хижаків є аналогічними до відповідних характеристик для звичайних організмів: хижак може накопичити не більше, ніж  $EX_{max}$  енергії, за один такт життя хижак до досягнення зрілості витрачає  $\Delta EX_{mi}$  енергії, а після досягнення зрілості —  $\Delta EX_{zrmi}$  енергії, максимальна тривалість життя хижака становить  $LX_{max}$  тактів, вік зрілості —  $LX_{zr}$  тактів, енергія, що витрачається при діленні, —  $\Delta EX_{di}$ , початковий запас енергії хижака —  $EX_0$ .

5. Якщо час життя хижака перевищує максимально можливий час  $LX_{max}$  або запас енергії зменшується до нуля, то хижак помирає.

6. Початковий розподіл хижаків у просторі є рівномірним. Початкова кількість хижаків —  $X_0$ . Загальну кількість хижаків на такті  $t$  позначимо через  $X_A(t)$ , тоді відносна чисельність хижаків  $X(t) = \frac{X_A(t)}{n^2}$ .

Запрограмувати ці правила будь-якою мовою програмування також нескладно. У результаті можна отримати залежності  $P$ ,  $N$  і  $X$  від часу та залежності  $N$  від  $P$  і  $X$  від  $N$ . Нехай, наприклад,  $n = 1000$ ,  $p_{max} = 10$ ,  $\Delta p = 1$ ,  $\Delta E_{pl} = 6$ ,  $E_{max} = 30$ ,  $\Delta E_{mi} = 3$ ,  $\Delta E_{zrmi} = 2$ ,  $\Delta E_{di} = 4$ ,  $E_0 = 4$ ,  $L_{max} = 20$ ,  $L_{zr} = 4$ ,  $N_0 = 0,3 N$ ,  $EX_{max} = 40$ ,  $\Delta EX_{mi} = 3,5$ ,  $\Delta EX_{zrmi} = 3$ ,  $\Delta EX_{di} = 5$ ,  $LX_{max} = 25$ ,  $LX_{zr} = 6$ ,  $P_X = 0,85$ ,

$$EX_0 = 5, X_0 = 0,05 n^2,$$

$$\Delta EX_{pl} = \begin{cases} 3L + \frac{E}{5}, & L < L_{zr}, \\ 10 + \frac{E}{5}, & L \geq L_{zr}, \end{cases}$$

де  $L$  — вік з'їденої жертви,  $E$  — її запас енергії. У цьому випадку графіки залежностей наведені на рис. 3.17-3.21, а фрагмент можливого розподілу організмів (чорні крапки) і хижаків (сірі крапки) — на рис. 3.22.

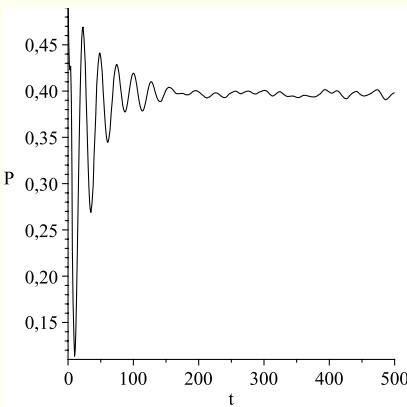


Рис. 3.17

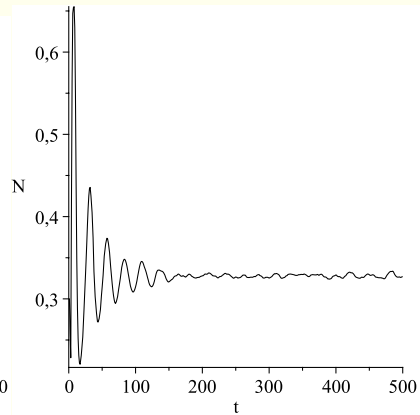


Рис. 3.18

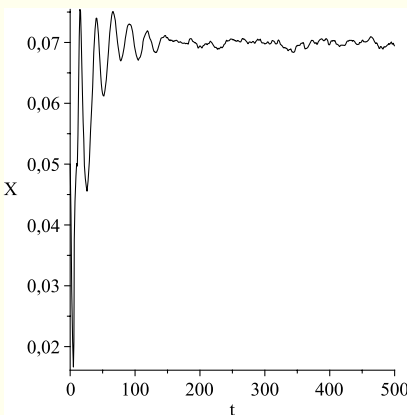


Рис. 3.19

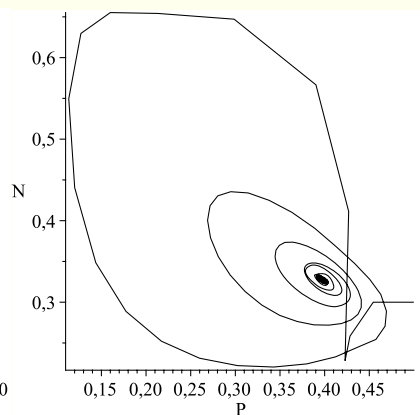


Рис. 3.20

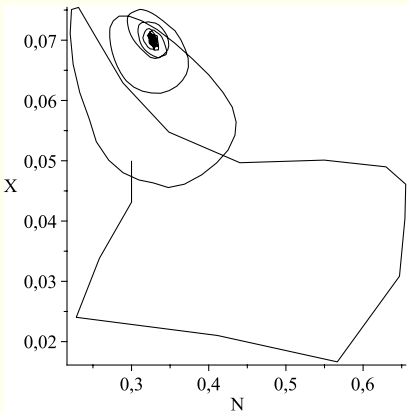


Рис. 3.21

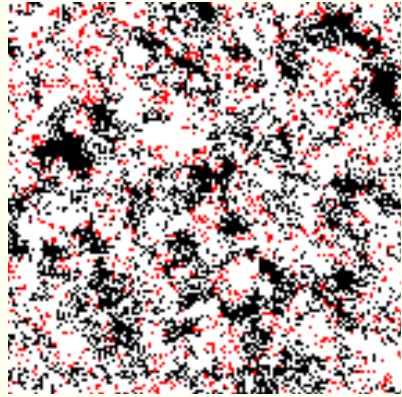


Рис. 3.22

Бачимо, що мають місце згасаючі коливання відносного рівня поживності середовища і відносних чисельностей нехижих і хижих організмів навколо їхніх середніх значень  $\bar{P} \approx 0,40$ ,  $\bar{N} \approx 0,33$ ,  $\bar{X} \approx 0,07$ . Крім того, можна встановити, що середня тривалість життя нехижих організмів становить  $\bar{L} \approx 3,5$  тактів, хоча є особини, які доживають до глибокої старості (20 тактів), а середня тривалість життя хижаків —  $\bar{L}_X \approx 3,7$  тактів, хоча зустрічаються особини, які доживають до глибокої старості (25 тактів). Згасаючі коливання виявляються подібними до тих, що виникають при інтегруванні модифікованої системи диференціальних рівнянь Лотки–Вольтерри з конкурентною боротьбою (§ 2.6).

Модель можна модифікувати, як у § 3.5: організм завжди намагається перейти в сусідню вільну клітинку з найбільшим рівнем енергоємності. Якщо клітинки в околі мають менший запас енергії, то особина залишається у тій самій клітинці. Крім того, при діленні організм вибирає сусідню вільну клітинку з найбільшим рівнем енергоємності. Хижаки теж вибирають собі жертву з найбільшим рівнем енергетичної цінності. У цьому випадку для тих самих числових даних отримуємо результати, наведені на рис. 3.24–3.28.

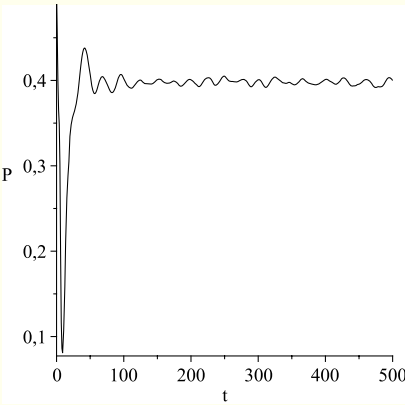


Рис. 3.23

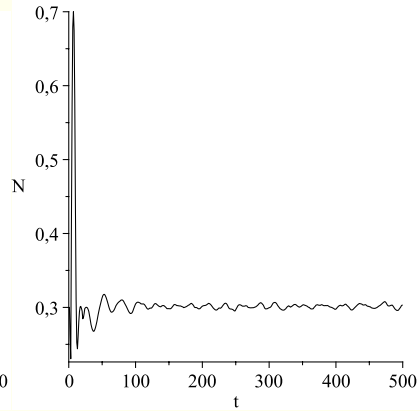


Рис. 3.24

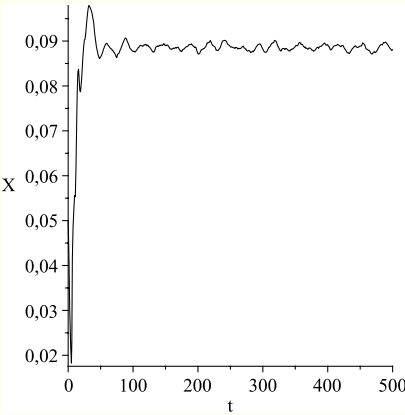


Рис. 3.25

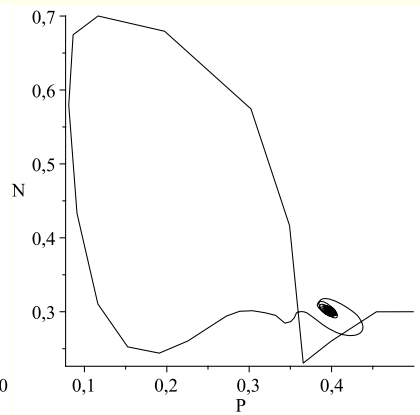


Рис. 3.26

Бачимо, що у цьому випадку мають місце коливання відносного рівня поживності середовища і відносних чисельностей нехижих і хижих організмів навколо їхніх середніх значень  $\bar{P} \approx 0,40$ ,  $\bar{N} \approx 0,30$ ,  $\bar{X} \approx 0,089$ , причому середня чисельність нехижих організмів у цьому випадку менша, а середня чисельність хижаків — більша, ніж у випадку «нерозумної» поведінки організмів. Крім того, середня тривалість життя

нехижких організмів становить  $\bar{L} \approx 3,2$  тактів, а середня тривалість життя хижаків —  $\bar{L}_X \approx 3,7$  тактів.

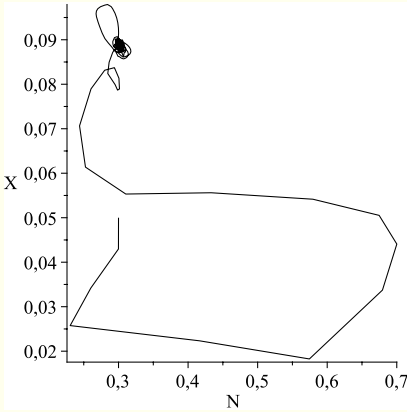


Рис. 3.27

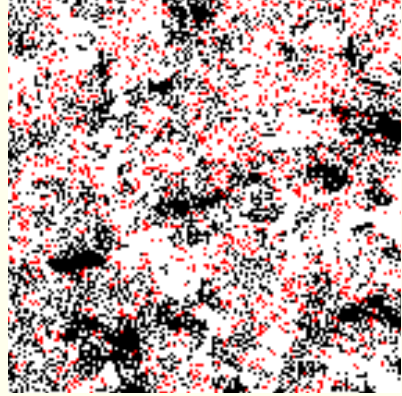


Рис. 3.28

З рис. 3.22, 3.28 видно, що на поживному середовищі є області, вільні від нехижких і хижих організмів (білого кольору), області, зайняті нехижими організмами (чорного кольору), області, які складаються з вільних клітинок і приречених на голодну смерть хижаків (сірого кольору) та області, які містять як нехижі і хижі організми, так і вільні клітинки. У випадку «розумної» поведінки нехижі організми утворюють фронти, які рухаються на вільні області, причому хижаки рухаються за фронтами і поїдають тих, хто відстає. Відсутність хижаків перед фронтами пояснюється великою кількістю зустрічей хижаків з нехижими організмами, які мають певні засоби захисту, що не залишає для хижаків жодних шансів.

### Питання до розділу 3

1. Що називають скінченим автоматом Мілі?
2. Що називають скінченим автоматом Мура? У чому полягає різниця між автоматами Мілі і Мура?
3. Що таке клітинний автомат? У чому полягають крайові умови циклічного типу для клітинних автоматів?

### Вправи до розділу 3

1. Створіть автомат Мілі для керування ліфтом у триповерховому будинку. Перетворіть його в автомат Мура.

2. Перетворіть автомат Мура для продажу кави з прикладу на с. 53 в автомат Мілі.

3. Складіть програму для реалізації клітинного автомата «Життя». Стан клітинного простору потрібно відображати графічно. Дослідіть еволюцію клітинного автомата для деяких початкових розташувань активних клітинок.

4. Розробіть програму для реалізації клітинного автомата «Дюни», поведінка якого відповідає наступним правилам:

- квадратна клітинка може перебувати в активному або пасивному («схованому») стані;
- якщо клітинка була активною і з восьми сусідніх клітинок є  $N$  або більше активних клітинок, то вона «ховається»;
- час перебування в «схованому» стані становить  $W$  тактів;
- якщо час «ховання» закінчився і в околі є не більше, ніж  $M$  активних клітинок, то клітинка знову стає активною.

Здійсніть моделювання для поля  $250 \times 250$  клітинок з крайовими умовами циклічного типу, чисел  $N = 3$ ,  $W = 5$ ,  $M = 3$  і деякого початкового розподілу активних клітинок. Дослідіть еволюцію клітинного автомата.

5. Розробіть програму для реалізації клітинного автомата «Робот», поведінка якого відповідає наступним правилам:

- квадратна клітинка може перебувати в активному або пасивному стані;
- в початковий момент часу всі клітинки пасивні, в центральній клітинці перебуває «робот», напрямлений направо;
- «робот» переходить у сусідню клітинку, якщо вона активна, він робить її пасивною і повертає наліво на  $90^\circ$ , якщо клітинка пасивна, то «робот» робить її активною і повертає направо на  $90^\circ$ .

## Розділ 4. Мережі Петрі

### § 4.1. Поняття класичних мереж Петрі

Мережі Петрі — це ще один з видів математичного моделювання. Їх використовують для моделювання асинхронних паралельних систем і процесів. Вперше мережі Петрі були запропоновані в докторській дисертації німецького математика Карла Адама Петрі в 1962 році.

Області застосування мереж Петрі включають дослідження і моделювання телекомунікаційних мереж, обчислювальних, виробничих і організаційних систем. Зокрема, мережі Петрі використовують при проектуванні мобільних телефонів, систем електронних розрахунків, систем керування рухом поїздів, систем безпеки та контролю доступу, при плануванні військових операцій і навчань.

Ми лише оглядово розглянемо мережі Петрі, перш за все, у найпростішому класичному розумінні. Повний виклад теорії мереж Петрі вимагає окремої книги.

*Мережа Петрі* є орієнтованим дводольним графом, який має чотири базові елементи: вузли, переходи, дуги і маркери. У літературі вузли інколи називають позиціями або вершинами, дуги — ребрами, а маркери — фішками. Дводольним називається граф, який має дві множини вершин і не має ребер, які з'єднують вершини однієї множини. Вузли визначають стан, в якому може знаходитись мережа чи її частина. Переходи — це активні елементи мережі, які визначають дії, що будуть виконуватись при спрацюванні переходів. Дуги з'єднують елементи протилежних типів: вузол з переходом і перехід з вузлом. Дуги не можуть з'єднувати безпосередньо вузли чи переходи. Вузли позначають кружками, переходи — прямокутниками або рисками, дуги — лініями зі стрілками на кінцях, маркери — крапками. Маркери знаходяться всередині вузлів і переміщуються в мережі в результаті спрацювання

переходів. При великій кількості маркерів у вузлі відображаються не крапки, а число відповідних маркерів. Дуги можуть бути простими або кратними. Розглянемо спочатку випадок простих дуг.

У випадку простих дуг *перехід* називається *дозволеним* (*збудженням*), якщо всі його вхідні вузли містять маркери. На рис. 4.1 і 4.2 наведено приклади дозволеного і недозволеного переходів.

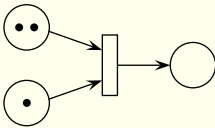


Рис. 4.1. Дозволений перехід

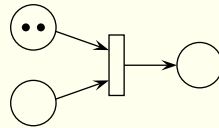


Рис. 4.2. Недозволений перехід

Дозволений перехід може спрацювати. У випадку простих дуг при *спрацюванні* (*запуску*) *переходу* вилучається по одному маркеру з кожного його вхідного вузла і розміщується по одному маркеру у всіх його вихідних вузлах (рис. 4.3).

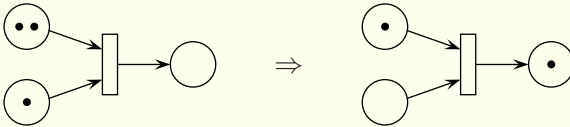


Рис. 4.3. Графічне зображення спрацювання переходу

Для моделювання реальних об'єктів буває зручно використовувати кратні дуги. Графічно кратні дуги, як правило, зображають як одну дугу, записуючи біля неї її кратність (рис. 4.4). У цьому випадку граф мережі насправді є мультиграфом.



Рис. 4.4

Для дозволеності переходу у випадку кратних дуг потрібно, щоб у кожному його вхідному вузлі було принаймні стільки маркерів, якою є кратність відповідної дуги.

При спрацюванні такого переходу з кожного його вхідного вузла вилучаються маркери в кількості, відповідній кратності дуги, а в кожному вихідному вузлі розміщується стільки маркерів, якою є кратність дуги від переходу до цього вузла (рис. 4.5).

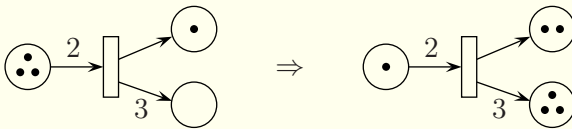


Рис. 4.5

У класичній мережі Петрі перехід спрацьовує миттєво. Одночасно може бути кілька дозволених переходів, але спрацьовує з них лише один, який вибирається довільним чином. Довільність вибору переходу для спрацювання визначає недетермінований характер поведінки мережі. Після спрацювання переходу в мережі знову може бути один або кілька дозволених переходів, з яких на наступному кроці спрацьовує тільки один.

Мережа Петрі описує множину різних допустимих варіантів поведінки модельованих систем. Може бути, що спрацювання якогось переходу (чи послідовності переходів) приведе систему в тупикову ситуацію, коли не зможе відбутись жоден перехід. Побудова й аналіз мереж Петрі дає змогу виявити всі варіанти поведінки реальних систем і тупикові ситуації.

У мережах Петрі можуть бути петлі. Але мережу Петрі з петлею завжди можливо подати у вигляді мережі без петлі, ввівши додатковий вузол і перехід (рис. 4.6).

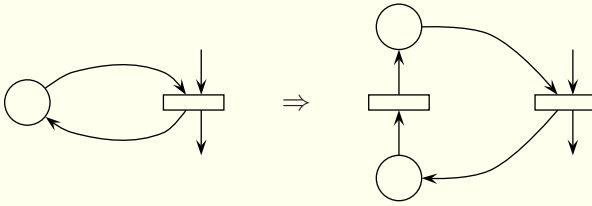
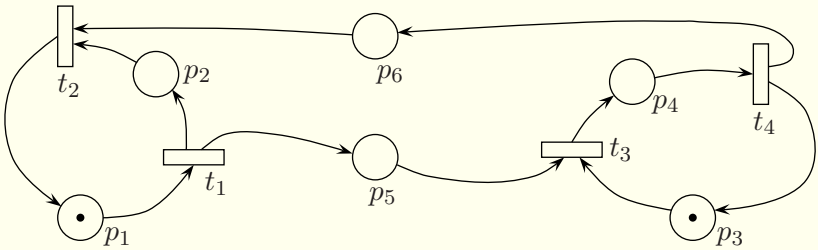


Рис. 4.6

При розумінні функціонування мережі Петрі вузли разом з маркерами в них розглядають як певні умови, а переходи — як події. Виникнення певної комбінації маркерів у вузлах приводить до настання деякої події, яка у свою чергу викликає зміну стану умов мережі. Стан мережі визначається сукупністю маркерів кожного вузла. Цю сукупність маркерів називають *розміткою*.

Класичні мережі Петрі є дискретно-ймовірнісними моделями.

**Приклад.** Розглянемо мережу Петрі  $N_1$  для передачі односторонніх повідомлень і підтвердження їхнього прийняття (рис. 4.7).

Рис. 4.7. Мережа  $N_1$ 

У цій мережі перехід  $t_1$  моделює відправлення повідомлення, перехід  $t_3$  — прийом повідомлення, перехід  $t_4$  — відправлення підтвердження, перехід  $t_2$  — отримання підтвердження. Переходи спрацьовують саме у цій послідовності, утворюючи цикл  $t_1 t_3 t_4 t_2$ .

## § 4.2. Формальне означення мереж Петрі і їхнє матричне подання

Є кілька способів формального означення мереж Петрі. Наведемо один з них. *Графом мережі Петрі* називають трійку  $G = (P, T, F)$ , де  $P = \{p_i\}$ ,  $i = 1, 2, \dots, m$ , — непорожня скінченна множина вузлів,  $T = \{t_j\}$ ,  $j = 1, 2, \dots, n$ , — непорожня скінченна множина переходів,  $F$  — функція інцидентності, що задає зв'язок між елементами множин  $P$  і  $T$ , тобто  $F : P \times T \cup T \times P \rightarrow N_0$ , де  $N_0 = \mathbb{N} \cup \{0\}$ . Значення функції  $F$  визначають наявність і кратність дуг.

*Розміткою (маркуванням)* мережі називають відображення  $M : P \rightarrow N_0$ , яке задає розміщення маркерів у вузлах. Тоді *мережа Петрі* — це четвірка  $N = (P, T, F, M_0)$  або  $N = (G, M_0)$ , де  $M_0$  — початкова розмітка. Значення розмітки для  $j$ -го вузла називають  $j$ -м *значенням розмітки*.

Мережу Петрі  $N_1$  з рисунка 4.7 можна формально записати так:

$$\begin{aligned} N_1 &= (P, T, F, M_0), \\ P &= \{p_1, p_2, p_3, p_4, p_5, p_6\}, \quad T = \{t_1, t_2, t_3, t_4\}, \\ F(p_1, t_1) &= 1, \quad F(p_2, t_2) = 1, \quad F(p_3, t_3) = 1, \quad F(p_4, t_4) = 1, \\ F(p_5, t_3) &= 1, \quad F(p_6, t_2) = 1, \quad F(t_1, p_2) = 1, \quad F(t_1, p_5) = 1, \\ F(t_2, p_1) &= 1, \quad F(t_3, p_4) = 1, \quad F(t_4, p_3) = 1, \quad F(t_4, p_6) = 1, \\ M_0(p_1) &= 1, \quad M_0(p_3) = 1. \end{aligned}$$

Для скорочення запису вказані лише ненульові значення функцій  $F$  і  $M_0$ . У випадку кратних дуг відповідні значення функції  $F$  були б більшими за одиницю.

Поряд з графічним і теоретико-множинним визначенням мереж Петрі широко використовується їхнє матричне подан-

ня. Функцію інцидентності  $F$  подають матрицями  $B$  і  $D$ :

$$\begin{aligned} B &= (b_{ij}), \quad i = \overline{1, m}, \quad j = \overline{1, n}, \quad b_{ij} = F(p_i, t_j), \\ D &= (d_{ij}), \quad i = \overline{1, m}, \quad j = \overline{1, n}, \quad d_{ij} = F(t_j, p_i). \end{aligned}$$

Матрицю  $B$  називають *матрицею входів*, а матрицю  $D$  — *матрицею виходів*. Для мереж без петель замість матриць  $B$  і  $D$  можна використовувати одну матрицю  $C = D - B$ . Її називають *матрицею змінювань*.

Розмітку мережі подають вектором

$$\overline{M} = (M(p_1) \quad M(p_2) \quad \dots \quad M(p_m))^T,$$

який частіше заради компактності будемо записувати у вигляді  $\overline{M} = (M(p_1), M(p_2), \dots, M(p_m))$ , продовжуючи розуміти його як вектор-стовпець. Тоді мережу Петрі однозначно задає набір  $(B, D, \overline{M}_0)$  або у випадку мережі без петель —  $(C, \overline{M}_0)$ , де  $\overline{M}_0$  — вектор початкової розмітки.

Побудуємо матричне подання мережі Петрі  $N_1$ , зображеної на рис. 4.7:

$$\begin{aligned} N_1 &= (B, D, \overline{M}_0), \\ B &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \quad D = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \overline{M}_0 = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}. \end{aligned}$$

Оскільки мережа не має петель, то можливе подання  $N_1 = (C, \overline{M}_0)$ , де

$$C = \begin{pmatrix} -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & -1 \\ 1 & 0 & -1 & 0 \\ 0 & -1 & 0 & 1 \end{pmatrix}. \quad (4.1)$$

### § 4.3. Властивості мереж Петрі

Використовуватимемо наступні допоміжні позначення: дозволений (збуджений) перехід  $t \in T$  у розмітці  $\overline{M}$  позначимо через  $\overline{M} \xrightarrow{t}$ , спрацювання переходу  $t \in T$  у розмітці  $\overline{M}$  позначимо через  $\overline{M} \xrightarrow{t} \overline{M}'$ . Аналогічні позначення будемо використовувати також для послідовностей переходів  $\sigma \in T^*$  ( $\sigma = t_{j_1} t_{j_2} \dots t_{j_k}$ ,  $T^*$  — позначення для послідовності множин  $T$ ): послідовність  $\sigma \in T^*$  дозволена у розмітці  $\overline{M}$ , якщо  $\overline{M} \xrightarrow{\sigma}$ , послідовність  $\sigma \in T^*$  спрацьовує у розмітці  $\overline{M}$ , якщо  $\overline{M} \xrightarrow{\sigma} \overline{M}'$ .

Наприклад, якщо для мережі  $N_1$  (рис. 4.7)

$$\overline{M} = (1, 0, 1, 0, 0, 0), \quad \sigma = t_1 t_3,$$

то  $\overline{M} \xrightarrow{\sigma}$ ,  $\overline{M} \xrightarrow{\sigma} \overline{M}'$ , де  $\overline{M}' = (0, 1, 0, 1, 0, 0)$ .

Під властивостями мереж Петрі розуміють набір корисних чи небажаних характеристик, притаманних досліджуваній мережі. Тому властивості можна умовно поділити на позитивні і негативні. Крім того, розрізняють поведінкові і структурні властивості мереж. Поведінкові властивості пов'язані з конкретною початковою розміткою, а структурні властивості мають місце незалежно від початкової розмітки.

Розглянемо основні властивості мереж Петрі.

1. *Досяжність розмітки.* Розмітка  $\overline{M}$  є досяжною для заданої мережі  $N = (G, \overline{M}_0)$ , якщо існує така дозволена у початковій розмітці  $\overline{M}_0$  послідовність переходів  $\sigma$ , що  $\overline{M}_0 \xrightarrow{\sigma} \overline{M}$ . Множину досяжних розміток мережі позначають через  $R(N)$ ,  $R(N) = \{\overline{M} \mid \exists \sigma \in T^* : \overline{M}_0 \xrightarrow{\sigma} \overline{M}\}$ .

2. *Потенційна живість.* Перехід  $t \in T$  називають потенційно живим у розмітці  $\overline{M}$ , якщо існує дозволена послідовність переходів, яка містить цей перехід:  $\exists \sigma \in T^* : t \in \sigma$ ,  $\overline{M} \xrightarrow{\sigma}$ .

3. *Живість.* Перехід називають живим, якщо він є потенційно живим у будь-якій досяжній розмітці, тобто перехід

$t \in T$  живий, якщо  $\forall \overline{M} \in R(N) \exists \sigma \in T^* : t \in \sigma, \overline{M} \xrightarrow{\sigma}$ . Мережа Петрі жива, якщо живі всі її переходи.

4. *Термінальність*. Розмітку  $\overline{M}$  мережі Петрі називають  $t$ -термінальною, якщо перехід  $t \in T$  не є в ній потенційно живим. Розмітку  $\overline{M}$  мережі називають термінальною, якщо вона є  $t$ -термінальною для всіх переходів мережі. Термінальну розмітку називають також тупиковою. У ній не є дозволеним жоден перехід.

5. *Обмеженість*. Вузол  $p \in P$  називають обмеженим ( $l$ -обмеженим), якщо існує таке ціле число  $l$ , що для будь-якої досяжної розмітки мережі кількість маркерів у вузлі  $p$  не перевищує числа  $l$ , тобто  $\exists l \in \mathbb{Z} \forall \overline{M} \in R(N) : M(p) \leq l$ . Мережа Петрі є обмеженою, якщо обмежені всі її вузли.

6. *Безпечність*. Вузол  $p \in P$  називають безпечним, якщо для довільної досяжної розмітки мережі кількість маркерів у вузлі  $p$  не перевищує одиниці, тобто дорівнює 0 або 1. Мережа Петрі є безпечною, якщо безпечні всі її вузли.

7. *Повторюваність*. Послідовність переходів  $\sigma$  називають повторюваною, якщо її можна запустити довільну кількість разів.

8. *Стаціонарна повторюваність*. Послідовність переходів  $\sigma$  називають стаціонарно повторюваною, якщо вона є повторюваною і переводить мережу в ту саму розмітку:  $\overline{M} \xrightarrow{\sigma} \overline{M}$ .

9. *Оборотність*. Мережу Петрі називають оборотною, якщо з довільної досяжної розмітки досяжною є початкова розмітка, тобто  $\forall \overline{M} \in R(G, \overline{M}_0) : \overline{M}_0 \in R(G, \overline{M})$ .

10. *Стійкість*. Мережу Петрі називають стійкою, якщо спрацювання одного з будь-яких двох дозволених переходів не може зробити недозволеним другий перехід.

11. *Консервативність*. Консервативною називають мережу, яка зберігає зважену суму маркерів відносно певного вагового вектора  $\overline{w} = (w_1 \ w_2 \ \dots \ w_m)$  з натуральними елементами, тобто  $\overline{w}\overline{M} = \overline{w}\overline{M}_0 = \text{const}$ , де  $\overline{M}$  — довільна розмітка,  $\overline{M}_0$  — початкова розмітка.

12. *Строго консервативність.* Строго консервативною називають мережу, сума маркерів якої є сталою, тобто консервативну мережу відносно одиничного вагового вектора.

Наприклад, для мережі Петрі  $N_1$  (рис. 4.7) множина досяжних розміток має вигляд

$$R(N_1) = \{(1, 0, 1, 0, 0, 0), (0, 1, 1, 0, 1, 0), (0, 1, 0, 1, 0, 0), (0, 1, 1, 0, 0, 1)\}.$$

Безпосередньо видно, що мережа  $N_1$  є живою, обмеженою, безпечною, оборотною, стійкою і має стаціонарно повторювану послідовність переходів  $t_1 t_3 t_4 t_2$ . Консервативність цієї мережі буде досліджено у наступному параграфі.

#### § 4.4. Матричний аналіз мереж Петрі

*Індикатором переходу*  $t_j$  назвемо вектор-стовпець

$$\bar{u}_{t_j} = (u_1 \quad u_2 \quad \dots \quad u_n)^T, \quad u_k = \begin{cases} 1, & k = j, \\ 0, & k \neq j. \end{cases}$$

Вхідні дуги переходу  $t_j$  можна подати як  $t_j^- = B\bar{u}_{t_j}$ , а вихідні — як  $t_j^+ = D\bar{u}_{t_j}$ , де  $B$  — матриця входів, а  $D$  — матриця виходів.

Розглянемо приклад для мережі  $N_1$  (рис. 4.7):

$$\bar{u}_{t_2} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \quad t_2^- = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \quad t_2^+ = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Умову дозволених переходу  $t \in T$  у розмітці  $\bar{M}$  можна подати нерівністю  $\bar{M} \geq B\bar{u}_t$  (ця нерівність означає, що кожен

елемент вектора-стовпця  $\overline{M}$  є не меншим за відповідний елемент вектора-стовпця  $t^-$ ). Спрацювання переходу  $t \in T$  у розмітці  $\overline{M}$  подається співвідношенням  $\overline{M}' = \overline{M} - B\overline{u}_t + D\overline{u}_t$ . Отже, формальний опис процесу функціонування мережі Петрі за відсутності петель дає наступна система:

$$\begin{cases} \overline{M}_{k+1} = \overline{M}_k + C\overline{u}_t, \\ \overline{M}_k \geq B\overline{u}_t, \quad t \in T, \end{cases} \quad (4.2)$$

де  $\overline{M}_k$  і  $\overline{M}_{k+1}$  — розмітки мережі Петрі на  $k$ -му і  $(k+1)$ -му кроках, а  $C$  — матриця змінювань. Систему (4.2) називають *системою станів* мережі Петрі.

Система станів мережі Петрі формально подає динаміку функціонування мережі, однак вона містить векторну нерівність і є рекурентною. Подамо з допомогою системи станів розмітку  $\overline{M}_k$  на довільному кроці  $k$ , отриману в результаті спрацювання дозволеної послідовності переходів  $\sigma$ , через початкову розмітку  $\overline{M}_0$ ,  $\overline{M}_0 \xrightarrow{\sigma} \overline{M}_k$ :

$$\overline{M}_k = \overline{M}_0 + C \sum_{t \in \sigma} \overline{u}_t.$$

Ввівши позначення

$$\overline{\sigma} = \sum_{t \in \sigma} \overline{u}_t,$$

отримаємо рівняння

$$\overline{M}_k = \overline{M}_0 + C\overline{\sigma}, \quad (4.3)$$

яке називають *фундаментальним рівнянням* мережі Петрі. Вектор  $\overline{\sigma}$  називають *вектором лічби спрацювань* послідовності  $\sigma$ .

При побудові фундаментального рівняння відбувається втрата інформації, проте це рівняння є лінійним, а тому його розв'язок відносно нескладно знайти. Розв'язність фундаментального рівняння в цілих невід'ємних числах є необхідною

умовою досяжності розмітки: якщо розмітка  $\overline{M}$  досяжна в мережі, то фундаментальне рівняння має цілий невід'ємний розв'язок  $\overline{\sigma}$ , такий що  $\overline{M}_0 \xrightarrow{\sigma} \overline{M}$ .

Однак існування цілого невід'ємного розв'язку рівняння (4.3) є недостатнім для досяжності розмітки. Розглянемо приклад мережі Петрі  $N_2$  з рис. 4.8.

Знайдемо матриці входів, виходів і змінювань (§ 4.2):

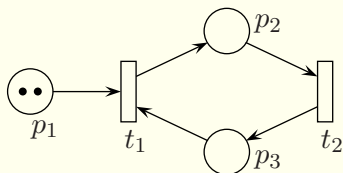


Рис. 4.8. Мережа  $N_2$

$$B = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad D = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad C = \begin{pmatrix} -1 & 0 \\ 1 & -1 \\ -1 & 1 \end{pmatrix}.$$

З допомогою рівняння (4.3) перевіримо виконання необхідної умови досяжності розмітки  $\overline{M}' = (1, 0, 0)$  з початкової розмітки  $\overline{M}_0 = (2, 0, 0)$ :

$$\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} -1 & 0 \\ 1 & -1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} \sigma_1 \\ \sigma_2 \end{pmatrix}.$$

Ця система має розв'язок  $\sigma_1 = \sigma_2 = 1$ , а тому виконується необхідна умова досяжності розмітки  $\overline{M}'$ . У той самий час, початкова розмітка мережі Петрі  $N_2$  є тупиковою, бо в ній недозволений жодний перехід. Отже, розмітка  $\overline{M}'$  не є досяжною.

*Інваріантом вузлів* мережі або *p-інваріантом* називають ненульовий вектор-рядок  $\overline{x} = (x_1 \ x_2 \ \dots \ x_m)$ , який складається з невід'ємних цілих чисел і є розв'язком рівняння

$$\overline{x}C = 0. \tag{4.4}$$

*Інваріантом переходів* мережі або *t-інваріантом* називають ненульовий вектор-стовпець  $\overline{y} = (y_1 \ y_2 \ \dots \ y_n)^T$ , який складається з невід'ємних цілих чисел і є розв'язком рівняння

$$C\overline{y} = 0. \tag{4.5}$$

Інваріанти вузлів і переходів визначаються неоднозначно. Якщо існує інваріант вузлів, який складається лише з натуральних чисел, то мережу Петрі називають  *$p$ -інваріантною*. Аналогічно,  *$t$ -інваріантною* називають мережу Петрі, у якій існує інваріант переходів, що складається лише з натуральних чисел.

**Теорема 1.** Мережа Петрі є консервативною тоді і тільки тоді, коли вона є  $p$ -інваріантною.

**Доведення.** Нехай мережа Петрі є  $p$ -інваріантною, тобто існує інваріант вузлів  $\bar{x}$ , який складається лише з натуральних чисел. Помножимо обидві частини фундаментального рівняння (4.3) для довільної досяжної розмітки  $\bar{M}$  зліва на вектор  $\bar{x}$ :

$$\bar{x}\bar{M} = \bar{x}\bar{M}_0 + \bar{x}C\bar{\sigma}. \quad (4.6)$$

Внаслідок рівності (4.4) маємо

$$\bar{x}\bar{M} = \bar{x}\bar{M}_0, \quad (4.7)$$

звідки випливає консервативність мережі Петрі.

Нехай тепер мережа Петрі є консервативною, тобто для довільної досяжної розмітки  $\bar{M}$  виконується рівність (4.7) з ваговим вектором  $\bar{x}$ , всі елементи якого є натуральними. Помноживши обидві частини фундаментального рівняння (4.3) зліва на вектор  $\bar{x}$ , отримаємо рівняння (4.6), з якого випливає рівність  $\bar{x}C\bar{\sigma} = 0$  для кожного вектора  $\bar{\sigma}$ . Отже, виконується рівність (4.4) для вектора  $\bar{x}$  з натуральними елементами. Тому мережа Петрі є  $p$ -інваріантною. Теорему доведено.

**Теорема 2.** Кожна консервативна мережа Петрі є обмеженою.

**Доведення.** Для кожної консервативної мережі Петрі виконується рівність (4.7) з ваговим вектором  $\bar{x}$ , всі елементи якого є натуральними. Виберемо довільний вузол  $q \in P$  і роз-

пишемо ліву частину рівності (4.7):

$$x_q M(q) + \sum_{p \neq q} x_p M(p) = \bar{x} \bar{M}_0.$$

Враховуючи невід'ємність доданків, маємо:

$$x_q M(q) \leq \bar{x} \bar{M}_0,$$

звідки

$$M(q) \leq \frac{\bar{x} \bar{M}_0}{x_q}.$$

Теорему доведено.

**Наслідок 1.** Кожна  $p$ -інваріантна мережа Петрі є обмеженою.

**Наслідок 2.** Необмежена мережа Петрі не є консервативною.

**Теорема 3.** Якщо в мережі Петрі немає інваріанта переходів, то в ній не існує стаціонарно повторювана послідовність.

**Доведення.** Припустимо, що в мережі Петрі існує стаціонарно повторювана послідовність. Тоді з фундаментального рівняння (4.3) випливає, що існує така досяжна розмітка  $\bar{M}$ , для якої  $\bar{M} = \bar{M} + C\bar{\sigma}$ , звідки  $C\bar{\sigma} = 0$ , де  $\bar{\sigma}$  — ненульовий вектор, що складається з невід'ємних цілих чисел. Існування такого вектора суперечить умові теореми. Теорему доведено.

Наявність інваріанта переходів мережі Петрі може означати існування у мережі стаціонарно повторюваної послідовності. Справді, використаємо інваріант переходів  $\bar{y}$  замість вектора лічби спрацьовувань у фундаментальному рівнянні (4.3). Тоді  $\bar{M}_k = \bar{M}_0$ , що означає наявність у мережі стаціонарно повторюваної послідовності. Однак, оскільки фундаментальне рівняння (4.3) не враховує дозволеність переходів, ця послідовність переходів може виявитись недозволеною.

**Приклад.** Для мережі Петрі  $N_1$  (рис. 4.7) у § 4.2 встановлено, що матриця  $C$  подається формулою (4.1). Знайдемо інваріанти мережі і зробимо висновки.

Інваріант вузлів знаходимо за формулою (4.4):

$$\begin{cases} -x_1 + x_2 + x_5 = 0, \\ x_1 - x_2 - x_6 = 0, \\ -x_3 + x_4 - x_5 = 0, \\ x_3 - x_4 + x_6 = 0 \end{cases} \Rightarrow \begin{cases} x_1 = x_2 + x_5, \\ x_4 = x_3 + x_5, \\ x_6 = x_5. \end{cases}$$

Надамо вільним невідомим  $x_2$ ,  $x_3$  і  $x_5$  натуральних значень, наприклад,  $x_2 = x_3 = x_5 = 1$ . Тоді  $x_1 = 2$ ,  $x_4 = 2$ ,  $x_6 = 1$  і існує інваріант вузлів  $\bar{x} = (2 \ 1 \ 1 \ 2 \ 1 \ 1)$ , який складається з натуральних чисел. Отже, згідно з теоремою 1 і наслідком 1 з теореми 2 мережа Петрі  $N_1$  є консервативною й обмеженою, проте не є строго консервативною, бо ваговий вектор  $\bar{x}$  не є одиничним.

Інваріант переходів знаходимо за формулою (4.5):

$$\begin{cases} -y_1 + y_2 = 0, \\ y_1 - y_2 = 0, \\ -y_3 + y_4 = 0, \\ y_3 - y_4 = 0, \\ y_1 - y_3 = 0, \\ -y_2 + y_4 = 0 \end{cases} \Rightarrow y_1 = y_2 = y_3 = y_4.$$

Інваріант переходів є, наприклад, таким:  $\bar{y} = (1 \ 1 \ 1 \ 1)^T$ . Тому мережа Петрі  $N_1$  може мати стаціонарно повторювану послідовність переходів. З аналізу мережі Петрі  $N_1$  у § 4.1 випливає, що вона справді має стаціонарно повторювану послідовність. ■

### § 4.5. Дерево досяжності

Дерево досяжності подає множину досяжних розміток мережі Петрі. Досяжних розміток у мережі може бути безмежна

кількість, тому для подання їх всіх у скінченному дереві використовують спеціальні позначення і терміни.

Кожна розмітка в дереві досяжності класифікується як початкова, гранична, термінальна, дублююча або внутрішня. Так само називають вершини дерева досяжності, в яких знаходяться ці розмітки. *Термінальною* називають розмітку, в якій жоден з переходів мережі Петрі не є дозволеним. *Дублюючою* називають розмітку, що раніше зустрічалась у дереві досяжності. *Граничною* називають ту розмітку, яка обробляється. Після обробки вона стає початковою, внутрішньою, термінальною або дублюючою. *Внутрішньою* є розмітка в дереві досяжності, яка не є початковою, термінальною, дублюючою і граничною.

Якщо в деякому вузлі мережі Петрі спостерігається зростання кількості маркерів до нескінченності, то для відображення цього факту використовується символ  $\omega$  у відповідному вузлі. Символ  $\omega$  в  $j$ -му вузлі розмітки  $M$  з'являється тоді, коли на шляху до розмітки  $M$  спостерігається розмітка  $M'$ , така, що всі значення розмітки  $M'$  не перевищують відповідних значень розмітки  $M$ , а  $j$ -те значення розмітки  $M'$  менше за  $j$ -те значення розмітки  $M$  ( $M'(k) \leq M(k)$ ,  $k = 1, 2, \dots, t$ ,  $M'(j) < M(j)$ ). Одного разу виникнувши, символ  $\omega$  вже не змінюється і не зникає в дереві досяжності, бо додавання до нескінченності або віднімання від нескінченності скінченного числа знову дає нескінченність.

Розглянемо алгоритм побудови дерева досяжності.

**Крок 1.** Вибираємо за граничну розмітку  $M$  початкову розмітку.

**Крок 2.** Обробку граничної розмітки  $M$  здійснюємо за наступними правилами.

1. Якщо у розмітці  $M$  не дозволений жоден перехід, то оголошуємо її термінальною.

2. Якщо в дереві досяжності є інша розмітка  $M'$ , яка не є граничною, така, що  $M' = M$ , то розмітка  $M$  оголошується

дублюючою.

3. Якщо на шляху від кореня дерева (початкової розмітки) до граничної є така розмітка  $M'$ , що  $M(k) \geq M'(k)$ ,  $k = 1, 2, \dots, m$ , то для всіх вузлів, таких, що  $M(j) > M'(j)$ , покладемо  $M(j) = \omega$ .

4. Якщо розмітка  $M$  не є початковою, дублюючою або термінальною, то оголошуємо розмітку  $M$  внутрішньою і для кожного дозволеного в цій розмітці переходу  $t_k \in T$  додаємо до дерева досяжності нову вершину з розміткою  $X_k$ , таку, що  $M \xrightarrow{t_k} X_k$ , і позначаємо дугу, яка веде з вершини  $M$  у вершину  $X_k$ , символом переходу  $t_k$ . Розмітку  $X_k$  будуємо за правилами спрацювання переходу  $t_k$  і оголошуємо граничною.

**Крок 3.** Якщо є нова гранична розмітка, то позначаємо її через  $M$  і переходимо до кроку 2, інакше — кінець роботи алгоритму.

**Приклад 1.** Побудуємо дерево досяжності для мережі Петрі  $N_1$  з рис. 4.7.

**Розв'язання.** У початковій розмітці  $(0, 1, 0, 1, 0, 0)$  єдиним дозволеним переходом є перехід  $t_1$ . Після його спрацювання отримуємо розмітку  $(0, 1, 1, 0, 1, 0)$ , яку оголошуємо внутрішньою. Тепер спрацьовує єдиний дозволений перехід  $t_3$  і отримуємо розмітку  $(0, 1, 0, 1, 0, 0)$ , яка знову буде внутрішньою. Після спрацювання єдиного дозволеного переходу  $t_4$  отримуємо внутрішню розмітку  $(0, 1, 1, 0, 0, 1)$ . Внаслідок спрацювання єдиного дозволеного переходу  $t_2$  отримуємо розмітку  $(0, 1, 0, 1, 0, 0)$ , яка збігається з початковою. Називаємо її дублюючою. Отримане дерево досяжності зображене на рис. 4.9. ■

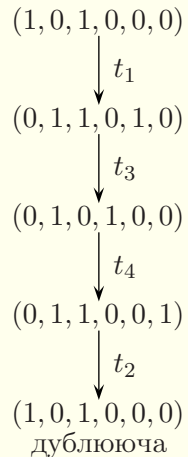


Рис. 4.9

Мережа Петрі  $l$ -обмежена тоді і тільки тоді, коли у вершинах її дерева досяжності відсутній символ  $\omega$ . Розміщення

символу  $\omega$  вказує, які вузли є необмеженими. Число  $l$  визначається найбільшою кількістю маркерів у розмітках дерева досяжності. Обмежена мережа Петрі може виявитись консервативною, перевіряють це з допомогою матричного аналізу. Мережа Петрі є строго консервативною тоді і тільки тоді, коли сума маркерів у всіх вузлах є сталою для всіх розміток. Мережа Петрі є безпечною тоді і тільки тоді, коли у розмітках дерева досяжності зустрічаються лише числа 0 і 1. Наявність дублюючих розміток без символів  $\omega$  вказує на наявність у мережі Петрі стаціонарно повторюваних послідовностей переходів. Мережа Петрі є живою тоді і тільки тоді, коли дерево досяжності не містить термінальних розміток. З побудованого дерева досяжності завжди видно множину досяжних розміток і можна зробити висновок про оборотність і стійкість мережі.

На основі дерева досяжності (рис. 4.9) робимо висновок, що мережа Петрі  $N_1$  є обмеженою, безпечною, живою, стійкою, оборотною і має стаціонарно повторювану послідовність переходів  $t_1 t_3 t_4 t_2$ .

**Приклад 2.** Побудуємо дерево досяжності для мережі Петрі  $N_3$  з рис. 4.10.

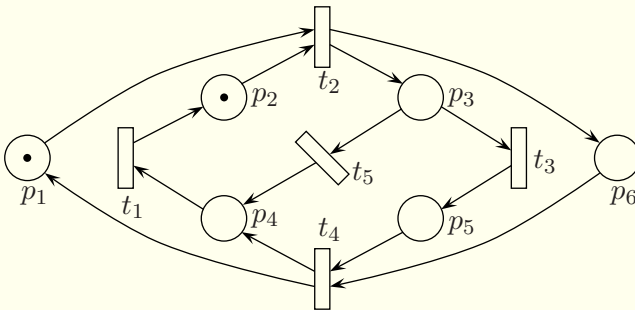


Рис. 4.10. Мережа  $N_3$

**Розв'язання.** У початковій розмітці  $(1, 1, 0, 0, 0, 0)$  дозволеним є лише перехід  $t_2$ . Запуск переходу  $t_2$  приводить до розмітки  $(0, 0, 1, 0, 0, 1)$ , в якій дозволеними є переходи

$t_3$  і  $t_5$ . Спрацювання переходу  $t_3$  приводить до розмітки  $(0, 0, 0, 0, 1, 1)$  з єдиним дозволеним переходом  $t_4$ . Запуск переходу  $t_4$  приводить до розмітки  $(1, 0, 0, 1, 0, 0)$  з єдиним дозволеним переходом  $t_1$ . Спрацювання переходу  $t_1$  приводить до розмітки  $(1, 1, 0, 0, 0, 0)$ , яка збігається з початковою. Називаємо її дублюючою. Якщо у розмітці  $(0, 0, 1, 0, 0, 1)$  спрацює перехід  $t_5$ , то отримаємо розмітку  $(0, 0, 0, 1, 0, 1)$  з єдиним дозволеним переходом  $t_1$ . Спрацювання переходу  $t_1$  приводить до розмітки  $(0, 1, 0, 0, 0, 1)$ , яка є термінальною. Дерево досяжності для мережі  $N_3$  зображене на рис. 4.11.

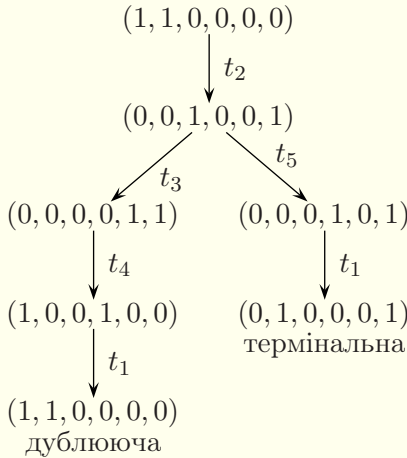


Рис. 4.11

На основі дерева досяжності робимо висновок, що мережа Петрі  $N_3$  є обмеженою, безпечною і строго консервативною (бо сума маркерів у всіх можливих розмітках дорівнює 2), проте не є живою, стійкою і оборотною. Мережа  $N_3$  має стаціонарно повторювану послідовність переходів  $t_2t_3t_4t_1$ , а спрацювання послідовності переходів  $t_5t_1$  призводить до термінальної розмітки. ■

**Приклад 3.** Побудуємо дерево досяжності для мережі Петрі  $N_4$  з рис. 4.12.

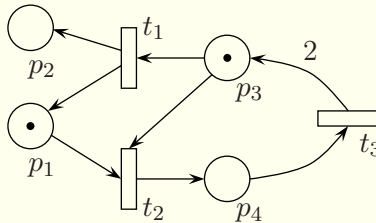


Рис. 4.12. Мережа  $N_4$

**Розв’язання.** У початковій розмітці  $(1, 0, 1, 0)$  дозволені є переходи  $t_1$  і  $t_2$ . Спрацювання переходу  $t_1$  призводить до термінальної розмітки  $(2, 1, 0, 0)$ . Запуск переходу  $t_2$  з початкової розмітки приводить до розмітки  $(0, 0, 0, 1)$  з єдиним дозволеним переходом  $t_3$ . Спрацювання переходу  $t_3$  дає розмітку  $(0, 0, 2, 0)$  з єдиним дозволеним переходом  $t_1$ . Спрацювання переходу  $t_1$  приводить до розмітки  $(1, 1, 1, 0)$ . Порівнюючи цю розмітку з розміткою  $(1, 0, 1, 0)$ , яка є на шляху до розглядуваної, помічаємо, що в другому вузлі спостерігається зростання кількості маркерів  $((1, 1, 1, 0) \geq (1, 0, 1, 0))$ . Тому в цьому вузлі треба покласти символ  $\omega$ , внаслідок чого отримуємо розмітку  $(1, \omega, 1, 0)$  з дозволеними переходами  $t_1$  і  $t_2$ . Запуск переходу  $t_1$  призводить до термінальної розмітки  $(2, \omega, 0, 0)$ . Спрацювання переходу  $t_2$  з розмітки  $(1, \omega, 1, 0)$  приводить до розмітки  $(0, \omega, 0, 1)$  з єдиним дозволеним переходом  $t_3$ . Спрацювання переходу  $t_3$  дає розмітку  $(0, \omega, 2, 0)$  з єдиним дозволеним переходом  $t_1$ . Запуск переходу  $t_1$  приводить до розмітки  $(1, \omega, 1, 0)$ , яка вже зустрічалась раніше. Називаємо її дублюючою. Дерево досяжності для мережі  $N_4$  зображене на рис. 4.13.

На основі дерева досяжності робимо висновки, що мережа Петрі  $N_4$  не є обмеженою, безпечною, консервативною, оборотною, стійкою і живою. Мережа  $N_4$  не має стаціонарно повторюваної послідовності, у деяких випадках спрацювання переходу  $t_1$  призводить до термінальної розмітки. Вузол  $p_2$  є фактично лічильником кількості спрацювань переходу

$t_1$ . Цей вузол може містити довільну кількість маркерів, що порушує обмеженість мережі Петрі. ■

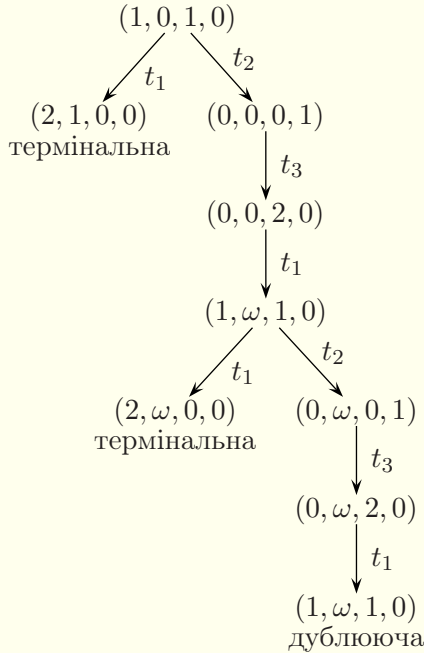


Рис. 4.13

## § 4.6. Некласичні мережі Петрі

Некласичні мережі Петрі дають більше можливостей для опису і моделювання систем. Існують різні некласичні мережі Петрі, орієнтовані на моделювання систем різних типів: імовірнісних, детермінованих та ін.

Найчастіше у некласичних мережах Петрі розширюються можливості переходів. По-перше, переходи можуть спрацювати паралельно й одночасно. По-друге, перехід може спрацювати не миттєво, а протягом певного часу. Такі мережі

Петрі називають часовими. Якщо збуджується кілька переходів, які претендують на використання тих самих маркерів, перехід, який спрацюватиме, вибирається у системі довільним чином. Можуть також задаватись імовірності спрацювання переходів, які записують біля відповідних вхідних дуг (рис. 4.14). Ці ймовірності є меншими від одиниці, тому вони відрізняються від кратностей дуг, що вказуються натуральними числами. У деяких випадках задають пріоритети для переходів.

Для того щоб мати можливість моделювати динамічні системи, тривалість спрацювання переходів має бути пов'язаною з модельним часом. У неklasичних мережах Петрі переходи бувають кількох типів — *миттєвий*, *детермінований*, *експоненціальний*, *рівномірний* тощо. Миттєвий перехід не пов'язаний з часом і спрацює миттєво, тобто так, як спрацювають переходи у класичних мережах Петрі. Детермінований перехід спрацює протягом вказаного для нього фіксованого часу. Час спрацювання експоненціального переходу є випадковою величиною з експоненціальним законом розподілу із заданим середнім значенням. Для рівномірного переходу задають два числа — початок і кінець проміжку, рівномірно розподілені випадкові числа з якого будуть братись як час спрацювання переходу. Деколи тип переходу вказують певним кольором.

Такі розширення роблять мережі Петрі неперервно-ймовірнісними моделями.

У неklasичних мережах Петрі можуть накладатись обмеження на максимальну кількість маркерів, які одночасно можуть перебувати у вузлі. Це може привести до того, що переходи, які містять достатню кількість маркерів у вхідних вузлах, не зможуть спрацювати, якщо хоч один з вихідних вузлів містить максимально допустиму кількість маркерів.

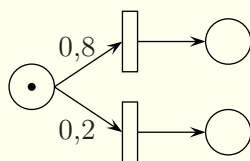


Рис. 4.14

У деяких випадках у мережах Петрі доводиться вводити дуги заперечення та інформаційні дуги. *Дуга заперечення* (її ще називають *інгібіторною дугою*) має вигляд лінії з кружечком на кінці замість стрілки і може з'єднувати лише вхідний вузол і перехід. Перехід з дугою заперечення є дозволим лише тоді, коли у всіх вхідних вузлах, з'єднаних з ним звичайними дугами, є достатня кількість маркерів, а у вхідних вузлах, з'єднаних з ним дугами заперечення, немає маркерів. Приклад спрацювання такого переходу є на рис. 4.15. Дуга заперечення дозволяє моделювати в мережах Петрі логічний оператор заперечення.

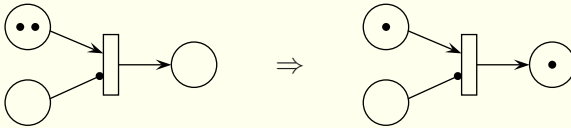


Рис. 4.15. Спрацювання переходу з дугою заперечення

*Інформаційна дуга* має вигляд пунктирної лінії зі стрілкою на кінці і може з'єднувати лише вхідний вузол і перехід. Перехід з інформаційною дугою є дозволим лише тоді, коли у всіх його вхідних вузлах є достатня кількість маркерів. При спрацюванні такого переходу маркери вилучаються лише з вхідних вузлів, з'єднаних з ним звичайними дугами. Маркери у вхідних вузлах, з'єднаних з переходом інформаційними дугами, лише дозволяють перехід, але не вилучаються з вузлів при його спрацюванні. Приклад спрацювання переходу з інформаційною дугою наведено на рис. 4.16.

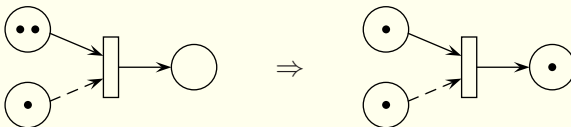


Рис. 4.16. Спрацювання переходу з інформаційною дугою

**Приклад 1.** В аеропорт для перевезення за тим самим маршрутом надходять вантажі у контейнерах, утворюючи стаціонарний пуассонівський потік з інтенсивністю 0,5 контейнера за хвилину. Вантажний аеропорт не має фіксованого розкладу, а літаки відправляються по мірі їхнього повного завантаження. У розпорядженні керуючого аеропортом для перевезення вантажів за даним маршрутом є три літаки з вантажопідйомністю 70 контейнерів і один літак з вантажопідйомністю 120 контейнерів. Час польоту кожного літака туди й назад, включаючи час розвантаження, розподілено рівномірно на проміжку від чотирьох до шести годин. Керуючий аеропортом намагається якомога частіше використовувати літаки меншої вантажопідйомності для зменшення часу, необхідного для транспортування вантажів. Літак, що піднімає 120 контейнерів, використовуються тільки тоді, коли інших літаків немає в наявності. Припускається, що часом завантаження літаків можна нехтувати. Скласти мережу Петрі для моделювання роботи аеропорту.

**Розв'язання.** Для моделювання стаціонарного пуассонівського потоку використовується експоненціальний закон розподілу (§ 5.3). У випадку експоненціально розподіленої випадкової величини середнє значення є оберненою величиною до інтенсивності. За одиницю модельного часу вибираємо хвилину. Мережа Петрі  $N_5$ , яка може використовуватись для моделювання цієї задачі, наведена на рис. 4.17.

У цій мережі перехід  $t_1$  є експоненціальним з середнім значенням 2 і моделює надходження контейнерів, перехід  $t_2$  є миттєвим і моделює завантаження малого літака, перехід  $t_3$  є рівномірним на проміжку  $[240, 360]$  і моделює політ малого літака, перехід  $t_4$  є миттєвим і моделює завантаження великого літака, перехід  $t_5$  є рівномірним на проміжку  $[240, 360]$  і моделює політ великого літака. У процесі моделювання вузол  $p_2$  міститиме чергу з контейнерів,  $p_3$  — кількість вільних малих літаків,  $p_4$  — кількість малих літаків, що перебувають

у польоті,  $p_5$  — кількість рейсів, виконаних малими літаками,  $p_6$  — кількість вільних великих літаків,  $p_7$  — кількість великих літаків, що перебувають у польоті,  $p_8$  — кількість рейсів, виконаних великими літаками. ■

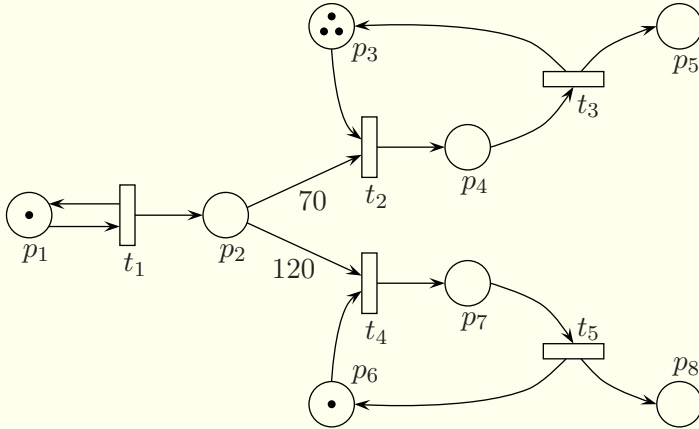


Рис. 4.17. Мережа  $N_5$

**Приклад 2.** На дорозі з двостороннім рухом і однією смугою для руху в кожен бік є світлофор на пішохідному переході (рис. 4.18). Автомобілі у напрямку  $AB$  утворюють стаціонарний пуассонівський потік з інтенсивністю 5 машин

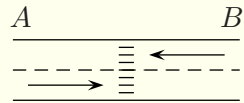
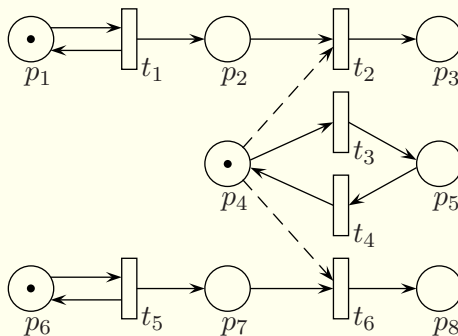


Рис. 4.18

за хвилину, а у напрямку  $BA$  — стаціонарний пуассонівський потік з інтенсивністю 6 машин за хвилину. На перетин пішохідного переходу автомобілі витрачають час, який є рівномірно розподіленою на проміжку  $[2, 5]$  секунд випадковою величиною. Жовтий сигнал світлофора можна не враховувати, бо він визначає рух відповідно до попереднього сигналу і є лише попередженням про зміну руху. Тривалість зеленого сигналу для автомобілів — 1 хвилину, а червоного — 45 секунд. Скласти мережу Петрі для моделювання руху автомобілів на цьому переході.

**Розв'язання.** За одиницю модельного часу вибираємо секунду. Мережа Петрі  $N_6$ , яка може використовуватись для моделювання цієї задачі, наведена на рис. 4.19. У ній перехід  $t_1$  є експоненціальним з середнім значенням 12 і моделює появу машин у напрямі  $AB$ , перехід  $t_2$  є рівномірним на проміжку  $[2, 5]$  і моделює проїзд пішохідного переходу у напрямі  $AB$ , перехід  $t_3$  є детермінованим зі значенням 60 і моделює тривалість зеленого сигналу світлофора для автомобілів, перехід  $t_4$  є детермінованим зі значенням 45 і моделює тривалість червоного сигналу світлофора для автомобілів, перехід  $t_5$  є експоненціальним з середнім значенням 10 і моделює появу машин у напрямі  $BA$ , перехід  $t_6$  є рівномірним на проміжку  $[2, 5]$  і моделює проїзд пішохідного переходу у напрямі  $BA$ . У процесі моделювання вузол  $p_2$  міститиме чергу з автомобілів у напрямі  $AB$ ,  $p_3$  — кількість автомобілів, що проїхали через перехід у напрямі  $AB$ ,  $p_7$  — чергу з автомобілів у напрямі  $BA$ ,  $p_8$  — кількість автомобілів, що проїхали через перехід у напрямі  $BA$ . Наявність маркера у вузлі  $p_4$  моделює зелений сигнал світлофора для автомобілів. Інформаційні дуги від вузла  $p_4$  до переходів  $t_2$  і  $t_6$  дозволяють використовувати один маркер у вузлі  $p_4$  для багатьох запусків цих переходів. ■

Рис. 4.19. Мережа  $N_6$ 

Ще одним способом розширення класичних мереж Петрі є кольорові мережі Петрі. У них маркери мають певний колір,





який визначає конкретний тип даних. Лише маркери певного кольору (типу) можуть дозволити і запустити конкретний перехід. У кольорових мережах Петрі задаються логічні умови переходів. Тоді логіка спрацювання переходу залежить не лише від кількості маркерів у вхідних вузлах, але і від їхнього змісту.

Високорівневим розширенням класичних і кольорових мереж є ієрархічні мережі Петрі. Пряме подання реальної системи у вигляді єдиної мережі Петрі може призводити до громіздкості, втрати наочності та труднощів аналізу. Ієрархічний підхід дозволяє поділити модель на логічні модулі, які описують окремі підсистеми реальної системи. У ієрархічних мережах Петрі використовуються спеціальні переходи, які називають субмережами. На верхньому рівні вони подаються як єдині елементи, але вони мають свою внутрішню структуру.

Детальніше з класичними і некласичними мережами Петрі можна ознайомитись у посібниках [12, 13].

### § 4.7. Програма HPetriSim для імітаційного моделювання мереж Петрі

Для імітаційного моделювання мереж Петрі існує ряд програм. Коротко розглянемо одну з них. Програма HPetriSim дозволяє моделювати як класичні, так і деякі некласичні мережі Петрі. На рис. 4.20 наведено загальний вигляд програми HPetriSim зі створеною у ній модифікованою мережею  $N_5$  без петлі (приклад 1 § 4.6). Числа біля вузлів визначають їхню ємність, біля переходів — величину затримки або її нижню межу, а біля дуг — їхню кратність. Немиттеві переходи відображаються з годинником.

Мережу Петрі у програмі HPetriSim будують з допомогою миші і кнопок панелі інструментів. Для створення вузлів, переходів і дуг використовують інструменти ,  і  відповідно. Для виділення створених об'єктів вибирають інструмент . Виділений об'єкт відображається червоним ко-

льором, а зверху на спеціальній панелі Format наводяться його властивості. Виділений об'єкт можна переміщувати мишею і змінювати його властивості.

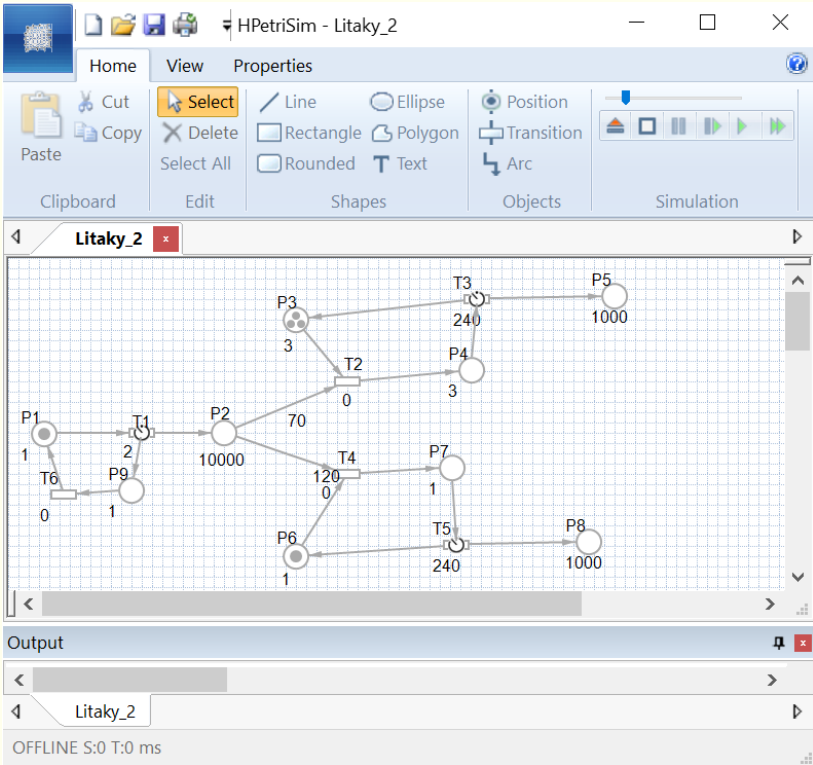


Рис. 4.20






Вузол має наступні властивості: Name (ім'я), Size (розмір), Label (показувати ім'я Name і властивості Property), To або Initial (початкова кількість маркерів) і Capacity (ємність, тобто максимальна допустима кількість маркерів у вузлі). Для того щоб у вузлі могло перебувати більше одного маркера, потрібно властивості Capacity надати більше від одиниці значення. Наприклад, на рис. 4.20 ємності вузла P2 присвоєно

значення 10000. Поточна кількість маркерів у вузлах відображається графічно у процесі моделювання.

Перехід має наступні властивості: Name, Size, Label, список для вибору типу переходу, T1 (тривалість спрацьовування переходу або її нижня межа) і T2 (верхня межа тривалості спрацьовування переходу). Тип переходу може бути таким: immediate (миттєвий перехід, значення за замовчуванням), exponential (експоненціальний перехід), delay (детермінований перехід) і equal distr. (рівномірний перехід). Значення властивості T1 (ціле число) має сенс середнього значення для експоненціального переходу, фіксованого значення для детермінованого переходу і лівої межі проміжку для рівномірного переходу. Властивість T2 використовують лише при рівномірному переході для вказування правої межі проміжку.

Дуга має такі властивості: Name, Size, Label, список для вибору типу дуги і Weight (кратність). Тип дуги може набувати значень: normal (звичайна дуга, за замовчуванням), inhibitor (дуга заперечення), test (інформаційна дуга).

До мережі Петрі можна додатково додавати текстові написи, лінії і багатокутники. Ці об'єкти не впливають на функціонування мережі і використовуються лише з метою її оформлення.

Кнопка  дозволяє перейти з режиму редагування мережі Петрі до режиму її моделювання і навпаки. Для запуску імітаційного моделювання мережі Петрі використовують кнопку , для покрокового виконання — , для зупинки — , для повернення до початкового стану — .

Процес моделювання графічно відображається у вікні з мережею Петрі. За замовчуванням моделювання триває протягом 1000 кроків або до досягнення термінальної (тупикової) розмітки чи обмеження на кількість маркерів у вузлі. Для збільшення тривалості моделювання вносять відповідні зміни на панелі інструментів Properties.

Абсолютно аналогічно можна працювати з попередньою

версією цієї програми HPSim.

Вільні версії програм HPSim і HPetriSim можна знайти на сайті [www.winpsim.de](http://www.winpsim.de).

### Питання до розділу 4

1. З яких основних елементів складається мережа Петрі?
2. Як позбутись петлі у мережі Петрі?
3. Що таке розмітка?
4. Як будуються матриці входів, виходів і змінювань?
5. Назвіть і поясніть основні властивості мереж Петрі.
6. Який вигляд мають система станів і фундаментальне рівняння мереж Петрі?
7. Що є інваріантом вузлів (переходів) мережі Петрі?
8. Як матричним методом дослідити мережу Петрі на консервативність і наявність стаціонарно повторюваної послідовності?
9. Як будується дерево досяжності? Які висновки про мережу можна зробити на його основі?
10. Назвіть основні розширення переходів і дуг для неklasичних мереж Петрі.
11. Назвіть особливості імітаційного моделювання у програмі HPetriSim небезпечних мереж Петрі.

### Вправи до розділу 4

1. Проаналізуйте мережу Петрі  $N_7$  з рис. 4.21. Подайте мережу Петрі у теоретико-множинному і матричному виглядах. Знайдіть інваріанти мережі і зробіть висновки.
2. Побудуйте дерево досяжності для мережі Петрі  $N_7$  з рис. 4.21 і зробіть висновки про властивості мережі.
3. Подайте мережу Петрі  $N_8$  з рис. 4.22 у матричному вигляді. Знайдіть інваріанти мережі і зробіть висновки.
4. Побудуйте дерево досяжності для мережі Петрі  $N_8$  з рис. 4.22 і зробіть висновки про властивості мережі.

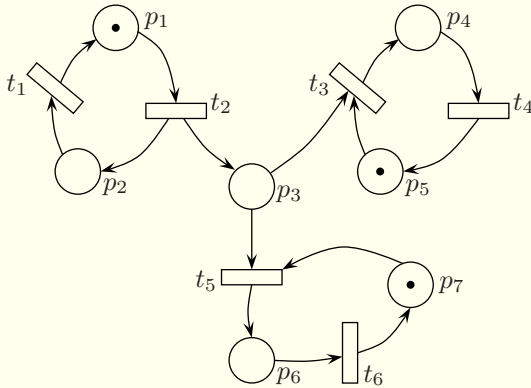


Рис. 4.21. Мережа  $N_7$

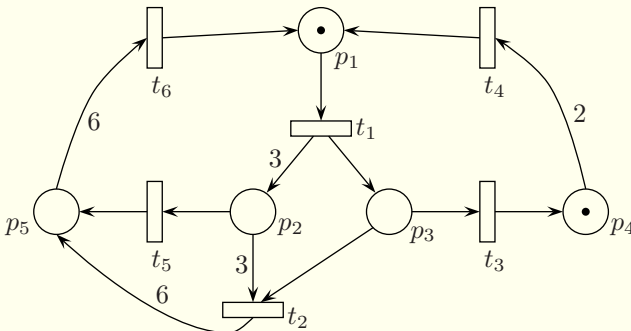


Рис. 4.22. Мережа  $N_8$

5. Проаналізуйте мережу Петрі  $N_9$  з рис. 4.23. Позбувшись петлі, подайте мережу Петрі у теоретико-множинному і матричному виглядах. Знайдіть інваріанти мережі і зробіть висновки.

6. Побудуйте дерево досяжності для мережі Петрі  $N_9$  з рис. 4.23 і зробіть висновки про властивості мережі.

7. З допомогою програми HPetriSim виконайте імітаційне моделювання мережі Петрі  $N_7$  з рис. 4.21.

8. З допомогою програми HPetriSim виконайте імітаційне моделювання мережі Петрі  $N_8$  з рис. 4.22.

9. З допомогою програми HPetriSim виконайте імітаційне моделювання мережі Петрі  $N_9$  з рис. 4.23.

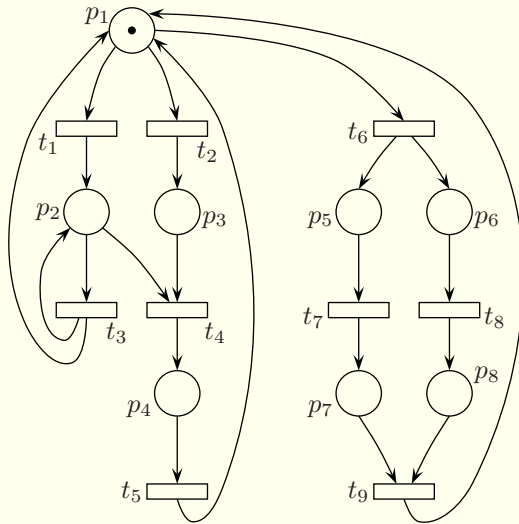


Рис. 4.23. Мережа  $N_9$

10. З допомогою програми HPetriSim виконайте імітаційне моделювання мережі Петрі  $N_6$  з рис. 4.19.

11. На комплектувальний конвеєр складального цеху кожні  $5 \pm 1$  хвилин надходять 5 деталей першого типу і кожні  $20 \pm 6$  хвилин надходять 20 деталей другого типу. Конвеєр складається із секцій, що вміщують по 10 деталей кожного типу. Чергова секція конвеєра надходить кожні 10 хвилин. Комплектація починаються тільки при наявності деталей обох типів у необхідній кількості і триває 10 хвилин. При недостатній кількості деталей секція конвеєра залишається порожньою. Складіть неklasичну мережу Петрі для моделювання надходження деталей і розміщення їх на конвеєрі. Яким чином можна визначити ймовірність пропуску секції?

## Розділ 5. Аналітичне моделювання систем масового обслуговування

### § 5.1. Поняття системи масового обслуговування

У різних сферах людської діяльності часто виникає масовий попит на певні послуги. Системи, спрямовані на задоволення цього попиту, називають *системами масового обслуговування*. З такими системами нам доводиться зустрічатись щодня. Це процеси обслуговування в магазині, кафе, бібліотеці, на автозаправній станції, транспортні системи, служби ремонту тощо. Виробничі процеси — теж системи масового обслуговування. У теорії і практиці моделювання важливе місце посідають моделі систем масового обслуговування.

Кожна система масового обслуговування містить один або декілька обслуговуючих пристроїв, до яких можуть формуватись черги з вимог, що потребують обслуговування. Вимоги можуть називатись по-різному — повідомленнями, клієнтами, замовленнями, викликами тощо. Правила або алгоритми взаємодії пристроїв і вимог називають *правилами обслуговування й організації черги*.

Системи масового обслуговування вивчає теорія масового обслуговування (її також називають теорією черг). Важливим типом завдань у теорії масового обслуговування є визначення місць накопичування вимог у системі обслуговування, наприклад, визначення необхідної кількості місць на стелажах на складі, кількості місць у палатах госпіталю та ін.

Найчастіше моделювання систем масового обслуговування здійснюють для визначення наступних показників:

- середньої кількості вимог, які система може обслужити за одиницю часу;
- імовірності того, що вимозі буде відмовлено в обслуговуванні;
- середнього часу очікування вимоги в черзі;

- середньої довжини черги;
- коефіцієнту завантаження обслуговуючого пристрою;
- середньої кількості пристроїв, зайнятих паралельним обслуговуванням;
- закону розподілу часу перебування вимог у системі;
- закону розподілу часу очікування;
- закону розподілу кількості вимог у черзі.

Моделі систем масового обслуговування бувають аналітичними й імітаційними. Аналітичне моделювання систем масового обслуговування можливе лише в окремих випадках і описується у наступних параграфах цього розділу. Імітаційне моделювання таких систем буде розглянуто у двох **наступних** розділах.

Перші теоретичні результати для систем масового обслуговування було отримано датським вченим, співробітником телефонної компанії Копенгагена Агнером Крарупом Ерлангом на початку ХХ століття.

Система масового обслуговування характеризується вхідним потоком вимог, правилами постановки вимог у черги і вибору з них і правилами обслуговування.

## § 5.2. Характеристики потоку подій і вхідний потік вимог

*Потоком подій* називають послідовність однотипних ситуацій, які відбуваються одна за одною в які-небудь моменти часу. Прикладом потоку подій є поява покупців у магазині.

Найважливішою характеристикою потоку подій є *інтенсивність потоку*  $\lambda$  — середня кількість подій, які відбуваються за одиницю часу. Інтенсивність може бути як сталою ( $\lambda = \text{const}$ ), так і залежною від часу  $t$ .

Потік подій, в якому інтервали часу  $\tau_k$  між сусідніми подіями набувають заздалегідь відомих значень, називають *детермінованим*. Якщо при цьому інтервали однакові ( $\tau_k = \tau$  для всіх  $k = 1, 2, \dots$ ), то потік подій називають *регулярним*.

Потік подій, в якому інтервали часу  $\tau_k$  між сусідніми подіями є випадковими величинами, називають *випадковим*. Потік подій називають *стаціонарним*, якщо його ймовірнісні характеристики не залежать від часу. Іншими словами, потік є стаціонарним, якщо ймовірність здійснення певної кількості подій за деякий проміжок часу  $\tau$  залежить тільки від довжини цього проміжку  $\tau$  і інтенсивності  $\lambda$ , і не залежить від місця цього проміжку на осі часу. Зокрема, інтенсивність стаціонарного потоку є сталою.

Потік подій є *потоком без післядії*, якщо ймовірність виникнення певної кількості подій за деякий проміжок часу  $\tau$  не залежить від кількості і часу настання подій, які відбулись у системі раніше, тобто не залежить від передісторії. Прикладом потоку подій без післядії може бути потік моментів часу реєстрації елементарних частинок лічильником Гейгера.

Потік подій називають *ординарним*, якщо події в ньому з'являються поодиноці, а не групами по кілька подій одразу. Наприклад, ординарним є потік автомобілів повз спостерігача по дорозі з однорядним рухом. У той самий час, по дорозі з кількома смугами для руху в один бік може одночасно повз спостерігача проїхати кілька автомобілів. Цей потік не є ординарним.

*Стаціонарним пуассонівським* або *найпростішим потоком* називають потік, який має властивості стаціонарності, ординарності і відсутності післядії.

Якщо подія в потоці полягає в появі вимоги, то маємо потік вимог. Для визначення вхідного потоку вимог потрібно задати моменти часу їхнього надходження до системи (закон надходження) і кількість вимог, які можуть надійти одночасно. Закон надходження може бути детермінованим (вимоги надходять у систему у фіксовані моменти часу) або ймовірнісним (проміжки часу між вимогами є випадковими величинами з деяким законом розподілу).

Для визначення вхідного потоку вимог, крім закону надхо-

дження, потрібно задати кількість вимог, які приходять одночасно. Системи, в яких вимоги з'являються не поодиноці, а групами (пакетами), називають системами з груповим надходженням вимог.

Стаціонарний пуассонівський (найпростіший) потік вимог відіграє серед інших потоків особливу роль. Це пов'язано з тим, що при накладанні достатньо великої кількості незалежних потоків отримується потік, близький до пуассонівського. На практиці вважають, що вхідний потік є пуассонівським, якщо вимоги надходять до системи від великої кількості незалежних джерел. Прикладами таких потоків можуть бути дзвінки абонентів у телефонній мережі або проїзд автомобілів повз спостерігача по одній смузі руху. Якщо розглядається нетривалий проміжок часу, то такий потік буде стаціонарним пуассонівським, інакше — нестаціонарним пуассонівським.

### § 5.3. Моделювання стаціонарного пуассонівського потоку

Покажемо тепер, як при моделюванні систем масового обслуговування можна задати стаціонарний пуассонівський потік вимог.

Імовірність того, що за час  $t$  надійде  $k$  вимог, які утворюють стаціонарний пуассонівський потік з інтенсивністю  $\lambda$ , є випадковою величиною з законом розподілу Пуассона і дорівнює

$$P_k(t) = \frac{(\lambda t)^k}{k!} e^{-\lambda t}, \quad k \geq 0, \quad t \geq 0. \quad (5.1)$$

Визначимо, який розподіл мають проміжки часу  $\tau$  між моментами надходження двох сусідніх вимог у стаціонарному пуассонівському потоці.

Функція розподілу  $F(t)$  проміжків часу між подіями в стаціонарному пуассонівському потоці визначає ймовірність того, що випадкова величина  $\tau$  набуде значення, меншого від  $t$ , тобто  $F(t) = P(\tau < t)$ . Тоді  $F(t) = 1 - P_0$ , де  $P_0$  — ймовірність

протилежної події, тобто ймовірність того, що за проміжок часу  $t$  до системи не надійде жодна вимога. Значення ймовірності  $P_0$  знаходимо за формулою (5.1) для  $k = 0$ :

$$P_0 = \frac{(\lambda t)^0}{0!} e^{-\lambda t} = e^{-\lambda t}.$$

Отже, функція розподілу випадкової величини  $\tau$  має вигляд:

$$F(t) = 1 - e^{-\lambda t}, \quad t \geq 0.$$

Ми отримали функцію розподілу експоненціального закону розподілу. Відповідна функція щільності  $f(t)$  знаходиться диференціюванням:

$$f(t) = \lambda e^{-\lambda t}, \quad t \geq 0.$$

Таким чином, приходимо до висновку: *проміжки часу між подіями в стаціонарному пуассонівському потоці розподілені за експоненціальним законом розподілу.*

## § 5.4. Правила обслуговування, організація черги і типи моделей систем масового обслуговування

*Правила обслуговування* вимог характеризуються часом (розподілом часу) обслуговування, кількістю вимог, які обслуговуються одночасно, і дисципліною обслуговування.

У системі масового обслуговування можуть використовуватись *одноканальні* і *багатоканальні* пристрої. В одноканальному пристрої може обслуговуватись не більше однієї вимоги у кожен конкретний момент часу. Натомість у багатоканальному пристрої може одночасно обслуговуватись кілька вимог. Час обслуговування може бути детермінованим або визначатись випадковою величиною з деяким законом розподілу.

Розрізняють безпріоритетні і пріоритетні дисципліни обслуговування. Пріоритет буває відносним і абсолютним. Відносний пріоритет визначає лише порядок перебування вимог у

черзі. Вимоги з однаковими пріоритетами можуть утворювати черги. У випадку абсолютного пріоритету вимога з вищим пріоритетом перериває обслуговування вимоги з нижчим пріоритетом. Вимога, обслуговування якої було перерване, може дообслуговуватись на тому самому пристрої після завершення обслуговування вимоги з вищим пріоритетом, проте вона може також втрачати право на дообслуговування.

Система масового обслуговування називається *багатофазною*, якщо вимоги послідовно обслуговується в кількох пристроях (наприклад, конвеєр). У протилежному разі її називають *однофазною*.

Перед пристроями обслуговування можуть формуватись черги, які можуть функціонувати за різними принципами.

Класична черга функціонує за принципом «*першим прийшов — першим обслуговують*», в англомовній літературі його часто називають FIFO (First In First Out). Прикладом такої черги є звичайна черга в магазині.

Існує також інший спосіб організації черги за принципом «*останнім прийшов — першим обслуговують*», його ще називають LIFO (Last In First Out). Таку чергу часто називають стеком. Прикладом такої черги є пором: автомобіль, який заїхав на пором першим, виїде з нього останнім.

Черга може функціонувати з урахуванням пріоритету: вимоги з одним пріоритетом формують звичайну чергу, вимоги з вищим пріоритетом розміщуються в голові черги. Пріоритет вимог може з часом змінюватись. У цьому випадку його називають динамічним на відміну від статичного, який не змінюється.

Крім того, черга може функціонувати за деяким нестандартним принципом, наприклад, першими можуть обслуговуватись вимоги, які потребують найменшого часу обслуговування, або вимоги з найбільшим часом перебування в багатофазній системі масового обслуговування. Вимоги з черги можуть також вибиратись випадковим чином, або з черги будуть

вибиратись лише ті вимоги, які задовольняють певні умови. Зокрема, вимоги можуть вибиратись за певним розкладом (планом). Наприклад, якщо вимоги належать до трьох класів, розклад може мати вигляд:  $(1, 2, 3, 1, 2, 1, 3)$ , тобто спочатку береться одна вимога першого класу, потім — другого і т. д. Розклад може бути прив'язаним до часу.

Черга може мати обмеження на довжину. Наприклад, якщо у черзі перебуває п'ять вимог при такій максимально можливій довжині черги, то при появі нової вимоги, їй буде відмовлено в обслуговуванні, бо вже досягнута максимально можлива довжина черги. Прикладом системи з обмеженою довжиною черги є комп'ютерна система з обмеженим буфером для зберігання вхідних повідомлень.

Може також накладатись обмеження на час перебування вимоги в черзі. Наприклад, якщо час перебування вимоги в черзі обмежений десятьма хвилинами, то при досягненні вимогою цього часу, вона взагалі вилучається з черги і отримує відмову в обслуговуванні.

Черга в системі масового обслуговування може не створюватись. У цьому випадку частина вимог отримує відмову в обслуговуванні.

Для позначення моделей систем масового обслуговування часто використовують запис  $X/Y/n/m$ , де  $X$  — розподіл часу надходження вимог,  $Y$  — розподіл часу обслуговування,  $n$  — кількість каналів для обслуговування в багатоканальному пристрої, а  $m$  — максимальна можлива довжина черги (відсутність  $m$  означає, що черга може мати необмежену довжину). Замість букв  $X$  і  $Y$  найчастіше використовують наступні позначення:  $D$  — детермінований потік вимог або детермінований час обслуговування,  $M$  — стаціонарний пуассонівський потік вимог або час обслуговування, заданий експоненціальним законом розподілу,  $G$  — довільний вхідний потік вимог або будь-який закон розподілу часу обслуговування.

З допомогою теорії систем масового обслуговування ана-

літичне моделювання можна здійснити лише у випадках  $M/M/1/m$ ,  $D/D/1/m$ ,  $M/G/1/m$  і  $M/M/n/m$ , де  $n = 1, 2, \dots$ ,  $m = 0, 1, 2, \dots, \infty$ . В інших випадках для визначення показників функціонування систем масового обслуговування треба використовувати імітаційне моделювання.

### § 5.5. Аналіз часової діаграми для системи масового обслуговування

Розглянемо приклад роботи багатоканальної системи масового обслуговування з двома пристроями (Пр1 і Пр2) з трьома позиціями для очікування в черзі (Поз1, Поз2 і Поз3). Її часову діаграму наведено на рис. 5.1. Час надходження вимоги до системи і час, коли вона залишила систему, наведено поряд з номером вимоги відповідно в нижній і верхній частинах рис. 5.1. Час вимірюється у хвилинали і для зручності заокруглений до цілого числа хвилин.

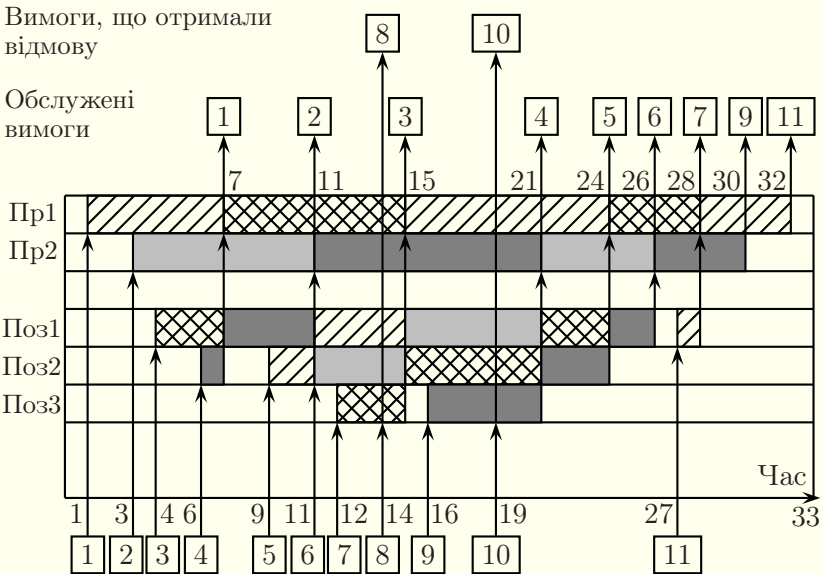


Рис. 5.1.

На основі цієї діаграми знайдемо основні характеристики роботи заданої системи масового обслуговування. З діаграми видно, що час спостереження за роботою системи становить  $T_{\text{СП}} = 33$  хв.

1. Імовірність обслуговування вимоги:

$$P_{\text{об}} = \frac{N_{\text{об}}}{N} = \frac{9}{11} \approx 0,818,$$

де  $N_{\text{об}}$  і  $N$  — відповідно кількість обслужених вимог і загальна кількість вимог.

2. Інтенсивність вхідного потоку:

$$\lambda = \frac{N}{T_{\text{СП}}} = \frac{11}{33} = \frac{1}{3} \approx 0,333 \text{ вимоги/хв.}$$

3. Пропускна здатність системи:

$$X = \frac{N_{\text{об}}}{T_{\text{СП}}} = \frac{9}{33} = \frac{3}{11} \approx 0,273 \text{ вимоги/хв.}$$

4. Імовірність відмови в обслуговуванні:

$$P_{\text{від}} = \frac{N_{\text{від}}}{N} = \frac{2}{11} \approx 0,182,$$

де  $N_{\text{від}}$  — кількість вимог, яким відмовлено в обслуговуванні.

5. Імовірність того, що вимога зустрине обидва пристрої вільними:

$$P_0 = \frac{T_{\text{вільн}}}{T_{\text{СП}}} = \frac{2}{33} \approx 0,061,$$

де  $T_{\text{вільн}}$  — час, протягом якого обидва пристрої були вільними.

6. Імовірність того, що обслуговуванням зайнятий тільки один пристрій з двох:

$$P_1 = \frac{T_{31} + T_{32}}{T_{\text{СП}}} = \frac{2 + 2}{33} \approx 0,121,$$

де  $T_{31}$  і  $T_{32}$  — час, протягом якого був зайнятим відповідно лише перший і лише другий пристрої.

7. Імовірність того, що обслуговуванням зайняті обидва пристрої, а в черзі відсутні вимоги:

$$P_2 = \frac{T_{31,2}}{T_{\text{СП}}} = \frac{(4-3) + (27-26) + (30-28)}{33} = \frac{4}{33} \approx 0,121,$$

де  $T_{31,2}$  — час, протягом якого були зайнятими обидва пристрої, а в черзі не було вимог.

8. Імовірність того, що в черзі є лише одна вимога:

$$\begin{aligned} P_3 &= \frac{T_{\text{чер},1}}{T_{\text{СП}}} = \frac{(6-4) + (9-7) + (26-24) + (28-27)}{33} = \\ &= \frac{7}{33} \approx 0,212, \end{aligned}$$

де  $T_{\text{чер},1}$  — час, протягом якого в черзі перебувала лише одна вимога.

9. Імовірність того, що в черзі знаходиться дві вимоги:

$$\begin{aligned} P_4 &= \frac{T_{\text{чер},2}}{T_{\text{СП}}} = \frac{(7-6) + (12-9) + (16-15) + (24-21)}{33} = \\ &= \frac{8}{33} \approx 0,242, \end{aligned}$$

де  $T_{\text{чер},2}$  — час, протягом якого в черзі перебувало дві вимоги.

10. Імовірність того, що в черзі знаходиться три вимоги:

$$P_5 = \frac{T_{\text{чер},3}}{T_{\text{СП}}} = \frac{(15-12) + (21-16)}{33} = \frac{8}{33} \approx 0,242,$$

де  $T_{\text{чер},3}$  — час, протягом якого в черзі було три вимоги.

11. Середня кількість пристроїв, зайнятих обслуговуванням:

$$\begin{aligned} N_{\text{пр}} &= 0 \cdot P_0 + 1 \cdot P_1 + 2 \cdot (P_2 + P_3 + P_4 + P_5) = \\ &= 1 \cdot \frac{4}{33} + 2 \cdot \left( \frac{4}{33} + \frac{7}{33} + \frac{8}{33} + \frac{8}{33} \right) = \frac{58}{33} \approx 1,76. \end{aligned}$$

12. Середня кількість вимог у черзі:

$$\begin{aligned} N_{\text{чер}} &= 1 \cdot P_3 + 2 \cdot P_4 + 3 \cdot P_5 = \\ &= 1 \cdot \frac{7}{33} + 2 \cdot \frac{8}{33} + 3 \cdot \frac{8}{33} = \frac{47}{33} \approx 1,42. \end{aligned}$$

13. Середній час перебування вимог у черзі:

$$\begin{aligned} t_{\text{чер}} &= \frac{\sum_{i=1}^N t_{\text{чер},i}}{N_{\text{об}}} = \\ &= \frac{0 + 0 + 3 + 5 + 6 + 10 + 12 + 0 + 10 + 0 + 1}{9} = \frac{47}{9} \approx 5,22 \text{ хв}, \end{aligned}$$

де  $t_{\text{чер},i}$  — час перебування  $i$ -ї вимоги в черзі ( $i = 1, 2, \dots, N$ ).

14. Середній час перебування вимог у черзі без врахування вимог, які не чекали:

$$t_{\text{чер}}^- = \frac{\sum_{i=1}^N t_{\text{чер},i}}{N_{\text{об}(-)}} = \frac{3 + 5 + 6 + 10 + 12 + 10 + 1}{7} = \frac{47}{7} \approx 6,71 \text{ хв},$$

де  $N_{\text{об}(-)}$  — кількість вимог, які чекали в черзі.

15. Середній час обслуговування вимоги пристроями:

$$\begin{aligned} t_{\text{об}} &= \frac{\sum_{i=1}^N t_{\text{об},i}}{N_{\text{об}}} = \frac{6 + 8 + 8 + 10 + 9 + 5 + 4 + 4 + 4}{9} = \frac{58}{9} \approx \\ &\approx 6,44 \text{ хв}, \end{aligned}$$

де  $t_{\text{об},i}$  — час обслуговування  $i$ -ї вимоги пристроєм ( $i = 1, 2, \dots, N$ ).

16. Загальний середній час перебування вимоги в системі масового обслуговування:

$$T = t_{\text{чер}} + t_{\text{об}} = \frac{47}{9} + \frac{58}{9} = \frac{35}{3} \approx 11,67 \text{ хв}.$$

17. Середня кількість вимог у системі масового обслуговування:

$$\bar{N} = N_{\text{чер}} + N_{\text{пр}} = \frac{47}{33} + \frac{58}{33} = \frac{35}{11} \approx 3,18.$$

### § 5.6. Формула Літгла

У теорії масового обслуговування важливе значення має формула Літгла, яку ще називають *законом стаціонарної черги*. Для виведення формули Літгла розглянемо довільну систему масового обслуговування з очікуванням.

Нехай  $\alpha(t)$  — вхідний потік вимог, який розглядається на проміжку  $[0, t]$ . Через  $\beta(t)$  позначимо вихідний потік вимог на цьому ж проміжку. Побудуємо графіки кусково сталих функцій  $\alpha(t)$  і  $\beta(t)$  (рис. 5.2).

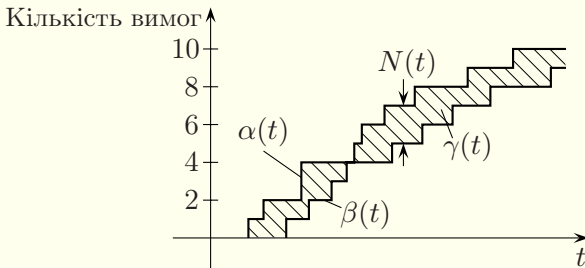


Рис. 5.2.

Кількість вимог, які знаходяться в системі у конкретний момент часу  $t$ , можна знайти за формулою  $N(t) = \alpha(t) - \beta(t)$ . Площа  $\gamma(t)$  заштрихованої фігури між лініями  $\alpha(t)$  і  $\beta(t)$  визначає загальний час, проведений всіма вимогами в системі.

Інтенсивність надходження вимог до системи масового обслуговування за час спостереження  $[0, t]$  визначається формулою

$$\lambda_t = \frac{\alpha(t)}{t}, \quad (5.2)$$

середній час перебування вимог у системі за той самий проміжок часу

$$T_t = \frac{\gamma(t)}{\alpha(t)}, \quad (5.3)$$

а середня кількість вимог, що перебували в системі за проміжок часу  $[0, t]$ ,

$$\bar{N}_t = \frac{\gamma(t)}{t}. \quad (5.4)$$

Використовуючи вирази (5.2) – (5.4), отримуємо формулу

$$\bar{N}_t = \lambda_t T_t. \quad (5.5)$$

Припустимо, що існують границі  $\lambda = \lim_{t \rightarrow \infty} \lambda_t$  і  $T = \lim_{t \rightarrow \infty} T_t$ , де  $\lambda$  – інтенсивність надходження вимог, а  $T$  – середній час перебування вимог у системі. У такому разі існує також границя для середньої кількості вимог, що знаходяться у системі, тобто  $\bar{N} = \lim_{t \rightarrow \infty} \bar{N}_t$ . Тоді з формули (5.5) отримуємо *формулу Літтла*:

$$\bar{N} = \lambda T.$$

Отже, для будь-якого вхідного потоку вимог і будь-якого розподілу часу їхнього обслуговування середня кількість вимог, що перебуває в системі, дорівнює добутку інтенсивності вхідного потоку вимог і середнього часу перебування їх у системі.

Формула Літтла має місце також для черги.

### § 5.7. Багатоканальні системи масового обслуговування з відмовами

Розглянемо багатоканальну систему масового обслуговування типу  $M/M/n/0$ . Ця система має  $n$  каналів, якщо зайняті всі канали, то вимога отримує відмову в обслуговуванні, а черга не формується. Прикладом такої системи є автостоянка. Якщо на ній всі місця зайняті, то водій поїде шукати поблизу

іншу стоянку. Інший приклад — телефонна система: якщо всі лінії зайняті, то телефонний дзвінок не обслуговується.

Зробимо наступні припущення:

– вхідний потік є стаціонарним пуассонівським з інтенсивністю  $\lambda$ ;

– всі канали є однотипними;

– час обслуговування вимоги в каналі є випадковим і утворює стаціонарний пуассонівський потік з інтенсивністю  $\mu$ .

Система масового обслуговування має  $n + 1$  станів:  $S_0$  — всі канали вільні,  $S_1$  — зайнято один канал (не важливо, який саме),  $S_2$  — зайнято два канали (не важливо, які саме) і т. д.,  $S_n$  — всі канали зайнято. Якщо система перебуває в стані  $S_n$ , то чергова вимога не буде обслуговуватись.

Оскільки вхідний потік є пуассонівським, а час обслуговування утворює пуассонівський потік, то система може переходити з одного стану в інші послідовно: з  $S_0$  в  $S_1$ , з  $S_1$  в  $S_0$  або в  $S_2$  і т. д. (рис. 5.3).

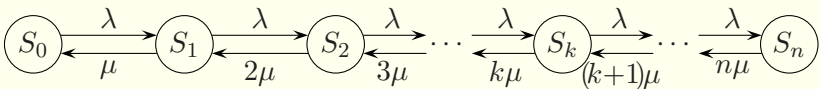


Рис. 5.3.

Зі стану  $S_k$  в стан  $S_{k-1}$  існує потік з інтенсивністю  $k\mu$  ( $k = 2, 3, \dots, n$ ), бо є  $k$  зайнятих каналів і кожен з них може звільнитись відповідно до потоку з інтенсивністю  $\mu$ , причому ці події незалежні. Отже, вихідний потік також є стаціонарним пуассонівським з інтенсивністю  $\lambda^* = \min\{\lambda, n\mu\}$ .

Для моделювання систем з пуассонівськими потоками використовують системи диференціальних рівнянь Колмогорова, в яких шукані функції, залежні від часу  $t$ , є ймовірностями станів.

Нехай  $p_i(t)$  — ймовірність  $i$ -го стану, тобто ймовірність того, що в момент часу  $t$  система перебуває в стані  $S_i$ ,  $i = 0, 1, 2, \dots, n$ . Для ймовірностей станів має виконуватись

умова

$$\sum_{i=0}^n p_i(t) = 1. \quad (5.6)$$

Розглянемо спочатку ймовірність  $p_0(t)$  того, що в момент часу  $t$  система перебуває в стані  $S_0$  (тобто всі канали вільні). Надамо  $t$  малий приріст  $\Delta t$  і знайдемо  $p_0(t + \Delta t)$  — ймовірність того, що в момент часу  $t + \Delta t$  система буде в стані  $S_0$ . Це можливо, очевидно, при реалізації одного з двох сценаріїв: або в момент часу  $t$  система вже була в стані  $S_0$  і за час  $\Delta t$  не вийшла з нього, або в момент часу  $t$  система була в стані  $S_1$ , а за час  $\Delta t$  перейшла з нього в стан  $S_0$ . Оскільки проміжок часу  $\Delta t$  дуже малий (потім ми навіть спрямуємо його до нуля), а всі потоки є пуассонівськими (зокрема, ординарними), то за цей час система не могла перейти зі стану  $S_2$  в стан  $S_0$ .

Знайдемо ймовірність розвитку подій по першому сценарію. Система знаходиться в стані  $S_0$ . У цьому випадку можливий перехід у стан  $S_1$ , ймовірність якого  $\lambda\Delta t$ . Тоді ймовірність того, що система залишиться в стані  $S_0$  дорівнює  $1 - \lambda\Delta t$ . Оскільки ймовірність опинитись у стані  $S_0$  у момент часу  $t$  дорівнює  $p_0(t)$ , а наступні переходи не залежать від того, яким чином система прийшла до цього стану, то ймовірність першого сценарію дорівнює добутку ймовірностей  $p_0(t)(1 - \lambda\Delta t)$ .

Міркуючи аналогічно, отримуємо ймовірність другого сценарію  $p_1(t)\mu\Delta t$ . Оскільки перший і другий сценарії є несумісними (реалізується тільки один з них), то ймовірність того, що система знаходиться у момент часу  $t + \Delta t$  в стані  $S_0$ , дорівнює

$$p_0(t + \Delta t) = p_0(t)(1 - \lambda\Delta t) + p_1(t)\mu\Delta t,$$

звідки

$$\frac{p_0(t + \Delta t) - p_0(t)}{\Delta t} = \mu p_1(t) - \lambda p_0(t).$$

Спрямовуючи  $\Delta t$  до нуля і переходячи до границі, отри-

муємо диференціальне рівняння для  $p_0(t)$ :

$$\frac{dp_0}{dt} = \mu p_1 - \lambda p_0.$$

Розглянемо тепер  $p_1(t)$  — ймовірність того, що система в момент часу  $t$  перебуває в стані  $S_1$ . Знайдемо ймовірність того, що система перебуватиме в стані  $S_1$  у момент часу  $t + \Delta t$ . Це може відбутись, якщо система перебувала в цьому стані і в час  $t$ . Крім того, це може трапитись, якщо система перебувала в момент часу  $t$  в стані  $S_0$  чи  $S_2$ . Оскільки зі стану  $S_1$  система могла перейти в стан  $S_0$  чи  $S_2$  (що є несумісними подіями), то ймовірність того, що система була і залишається в стані  $S_1$ , дорівнює  $p_1(t)(1 - (\lambda + \mu)\Delta t)$ . Ймовірність переходу за час  $\Delta t$  зі стану  $S_0$  в стан  $S_1$  дорівнює  $p_0(t)\lambda\Delta t$ , а ймовірність переходу за той самий час зі стану  $S_2$  в стан  $S_1$  дорівнює  $p_2(t) \cdot 2\mu\Delta t$ . Отже, ми маємо:

$$\begin{aligned} p_1(t + \Delta t) &= p_1(t)(1 - (\lambda + \mu)\Delta t) + p_0(t)\lambda\Delta t + 2p_2(t)\mu\Delta t \Rightarrow \\ \frac{p_1(t + \Delta t) - p_1(t)}{\Delta t} &= p_0(t)\lambda + 2p_2(t)\mu - (\lambda + \mu)p_1(t) \Rightarrow \\ \frac{dp_1}{dt} &= \lambda p_0 + 2\mu p_2 - (\lambda + \mu)p_1. \end{aligned}$$

Аналогічно будуємо й інші рівняння, внаслідок чого отримуємо систему диференціальних рівнянь Колмогорова

$$\begin{cases} \frac{dp_0}{dt} = \mu p_1 - \lambda p_0, \\ \frac{dp_1}{dt} = \lambda p_0 + 2\mu p_2 - (\lambda + \mu)p_1, \\ \frac{dp_2}{dt} = \lambda p_1 + 3\mu p_3 - (\lambda + 2\mu)p_2, \\ \dots \\ \frac{dp_{n-1}}{dt} = \lambda p_{n-2} + n\mu p_n - (\lambda + (n-1)\mu)p_{n-1}, \\ \frac{dp_n}{dt} = \lambda p_{n-1} - n\mu p_n. \end{cases} \quad (5.7)$$

Замість будь-якого з рівнянь у цій системі можна покласти умову (5.6) (наприклад, замість останнього).

Початкові умови для системи диференціальних рівнянь (5.7) легко записуються, якщо відомий початковий стан системи масового обслуговування. Нехай, наприклад, початковим був стан  $S_0$ . Тоді  $p_0(0) = 1$ ,  $p_1(0) = 0$ ,  $p_2(0) = 0$ ,  $\dots$ ,  $p_n(0) = 0$ .

Отриману задачу Коші для лінійної системи диференціальних рівнянь зі сталими коефіцієнтами (5.7) завжди можна однозначно розв'язати і знайти ймовірності станів у кожен момент часу. Типові графіки для ймовірностей станів для  $n = 2$  показані на рис. 5.4.

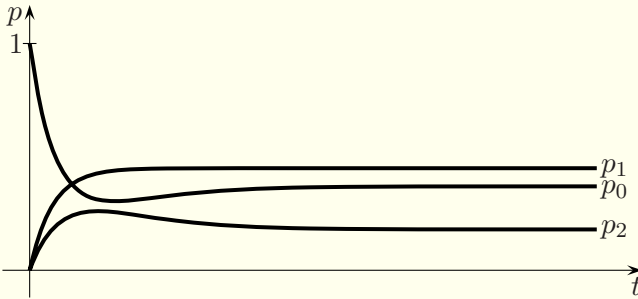


Рис. 5.4.

У теорії випадкових процесів доводиться, що якщо кількість станів є скінченною і з кожного з них можна перейти в інший за скінченну кількість кроків, то існують границі ймовірностей  $p_k(t)$  при  $t \rightarrow \infty$ , які не залежать від початкового стану. Їх називають *фінальними ймовірностями*<sup>1</sup>.

Фінальні ймовірності позначатимемо тими самими буквами  $p_1, p_2, \dots, p_n$ , що й ймовірності станів, але тепер їх трактуватимемо як сталі числа.

Для фінальних ймовірностей з системи диференціальних рівнянь (5.7) і умови (5.6) отримуємо систему лінійних алге-

<sup>1</sup>В деяких книгах фінальні ймовірності називають граничними ймовірностями.

бричних рівнянь

$$\begin{cases} \mu p_1 - \lambda p_0 = 0, \\ \lambda p_0 + 2\mu p_2 - (\lambda + \mu)p_1 = 0, \\ \lambda p_1 + 3\mu p_3 - (\lambda + 2\mu)p_2 = 0, \\ \dots \\ \lambda p_{n-2} + n\mu p_n - (\lambda + (n-1)\mu)p_{n-1} = 0, \\ p_0 + p_1 + \dots + p_n = 1. \end{cases} \quad (5.8)$$

З перших  $n$  рівнянь цієї системи, використовуюючи метод математичної індукції, знаходимо:

$$\begin{aligned} p_1 &= \frac{\lambda}{\mu} p_0, \\ p_2 &= \frac{(\lambda + \mu) \frac{\lambda}{\mu} p_0 - \lambda p_0}{2\mu} = \frac{\lambda^2}{2\mu^2} p_0, \\ p_3 &= \frac{(\lambda + 2\mu) \frac{\lambda^2}{2\mu^2} p_0 - \lambda \frac{\lambda}{\mu} p_0}{3\mu} = \frac{\lambda^3}{3 \cdot 2\mu^3} p_0, \\ &\dots \\ p_{k-1} &= \frac{\lambda^{k-1}}{(k-1)! \mu^{k-1}} p_0, \\ p_k &= \frac{(\lambda + (k-1)\mu) \frac{\lambda^{k-1}}{(k-1)! \mu^{k-1}} p_0 - \lambda \frac{\lambda^{k-2}}{(k-2)! \mu^{k-2}} p_0}{k\mu} = \frac{\lambda^k}{k! \mu^k} p_0, \\ &\dots \\ p_n &= \frac{\lambda^n}{n! \mu^n} p_0. \end{aligned}$$

Нехай  $\alpha = \frac{\lambda}{\mu}$  — *зведена інтенсивність вимог*, вона характеризує середню кількість вимог, що надійшли протягом середнього часу обслуговування вимоги. Підставляючи отримані вирази для фінальних імовірностей в останнє рівняння

системи (5.8) (тобто умову (5.6)), отримуємо:

$$p_0 \left( 1 + \alpha + \frac{\alpha^2}{2!} + \frac{\alpha^3}{3!} + \dots + \frac{\alpha^n}{n!} \right) = 1,$$

звідки

$$p_0 = \left( 1 + \alpha + \frac{\alpha^2}{2!} + \frac{\alpha^3}{3!} + \dots + \frac{\alpha^n}{n!} \right)^{-1} = \left( \sum_{k=0}^n \frac{\alpha^k}{k!} \right)^{-1}. \quad (5.9)$$

Імовірності решти станів подаються формулами:

$$p_k = \frac{\alpha^k}{k!} p_0, \quad k = 1, 2, \dots, n. \quad (5.10)$$

Співвідношення (5.9), (5.10) називають *формулами Ерланга*.

Якщо всі канали зайняті, то чергова вимога не буде обслуговуватись системою й отримає відмову. Тому *імовірність відмови* системи

$$P_{\text{відм}} = p_n = \frac{\alpha^n}{n!} p_0. \quad (5.11)$$

*Імовірність обслуговування* вимоги

$$P_{\text{обсл}} = 1 - P_{\text{відм}} = 1 - \frac{\alpha^n}{n!} p_0. \quad (5.12)$$

*Пропускна здатність*  $A$  — це середня кількість вимог, обслугованих системою за одиницю часу:

$$A = \lambda P_{\text{обсл}}. \quad (5.13)$$

Знайдемо *середню кількість зайнятих каналів*:

$$\begin{aligned} N_3 &= p_1 + 2p_2 + \dots + np_n = \\ &= \sum_{k=1}^n k \frac{\alpha^k}{k!} p_0 = \alpha p_0 \sum_{k=1}^n \frac{\alpha^{k-1}}{(k-1)!} = \alpha p_0 \sum_{k=0}^{n-1} \frac{\alpha^k}{k!} = \\ &= \alpha \left( \sum_{k=0}^n \frac{\alpha^k}{k!} \right)^{-1} \cdot \left( \sum_{k=0}^n \frac{\alpha^k}{k!} - \frac{\alpha^n}{n!} \right) \Rightarrow \\ &N_3 = \alpha(1 - p_n). \end{aligned} \quad (5.14)$$

*Коефіцієнт завантаження одного каналу*

$$K_3 = \frac{N_3}{n}. \quad (5.15)$$

Зокрема, для  $n = 1$  маємо:

$$P_{\text{відм}} = N_3 = K_3 = \frac{\alpha}{1 + \alpha}.$$

**Приклад.** У перукарні є три перукарі. Час обслуговування клієнта кожним з них є випадковою величиною з експоненціальним законом розподілу з середнім значенням 10 хвилин. Клієнти часом своєї появи утворюють стаціонарний пуассонівський потік з інтенсивністю  $\lambda = 1/5$ . Якщо всі перукарі зайняті, то клієнт іде в іншу перукарню, яка знаходиться поруч. Знайти ймовірність того, що клієнт піде в іншу перукарню, пропускну здатність системи, середню кількість зайнятих перукарів, коефіцієнт завантаження перукаря.

**Розв'язання.** З курсу теорії ймовірностей відомо, що інтенсивність потоку подій  $\lambda$  для експоненціального закону розподілу проміжків між подіями є оберненою величиною до середнього значення. Тому в цій задачі  $\lambda = 1/5$ ,  $\mu = 1/10$ ,  $n = 3$ . Отже, зведена інтенсивність  $\alpha = 2$ , а за формулами (5.9)–(5.15) маємо:

$$\begin{aligned} p_0 &= \left(1 + 2 + \frac{2^2}{2!} + \frac{2^3}{3!}\right)^{-1} = \frac{3}{19}, \\ p_1 &= \frac{6}{19}, \quad p_2 = \frac{6}{19}, \quad p_3 = \frac{4}{19} \quad \Rightarrow \\ P_{\text{відм}} &= \frac{4}{19} \approx 0,21, \quad A = \frac{3}{19} \approx 0,16, \\ N_3 &= \frac{6}{19} + 2 \cdot \frac{6}{19} + 3 \cdot \frac{4}{19} = \frac{30}{19} \approx 1,58, \quad K_3 = \frac{10}{19} \approx 0,53. \quad \blacksquare \end{aligned}$$

### § 5.8. Багатоканальні системи масового обслуговування з обмеженою чергою

Розглянемо багатоканальну систему масового обслуговування типу  $M/M/n/m$  з обмеженою чергою. Ця система має  $n$  каналів і  $m$  позицій для очікування в черзі. Якщо зайняті всі  $n$  каналів, а довжина черги менша від  $m$ , то вимога стає у хвіст черги. Якщо зайняті всі  $n$  каналів і довжина черги дорівнює  $m$ , то вимога отримує відмову в обслуговуванні.

Зробимо наступні припущення:

- вхідний потік є стаціонарним пуассонівським з інтенсивністю  $\lambda$ ;
- всі канали є однотипними;
- час обслуговування вимоги в каналі є випадковим і утворює стаціонарний пуассонівський потік з інтенсивністю  $\mu$ .

Система масового обслуговування має  $n + m + 1$  станів:  $S_0$  — всі канали вільні,  $S_1$  — зайнято один канал (не важливо, який саме),  $S_2$  — зайнято два канали (не важливо, які саме) і т. д.,  $S_n$  — всі канали зайнято, але черга є порожньою,  $S_{n+k}$  — всі канали зайнято, а в черзі перебуває  $k$  вимог. Якщо система перебуває в стані  $S_{n+m}$ , то черговій вимозі буде відмовлено в обслуговуванні.

Оскільки вхідний потік є пуассонівським і час обслуговування утворює пуассонівський потік, то система може переходити з одного стану в інші послідовно: з  $S_0$  в  $S_1$ , з  $S_1$  в  $S_0$  або в  $S_2$  і т. д. (рис. 5.5).

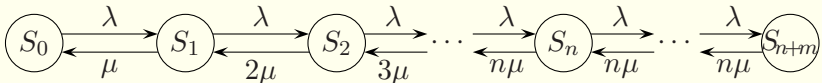


Рис. 5.5.

Вихідний потік тут також є стаціонарним пуассонівським з інтенсивністю  $\lambda^* = \min\{\lambda, n\mu\}$ .

Нехай  $p_i(t)$  — ймовірність  $i$ -го стану, тобто ймовірність того, що в момент часу  $t$  система перебуває в стані  $S_i$ ,

$i = 0, 1, 2, \dots, n + m$ . Для ймовірностей станів має виконуватись умова

$$\sum_{i=0}^{n+m} p_i(t) = 1. \quad (5.16)$$

Аналогічно § 5.7 будується система диференціальних рівнянь Колмогорова

$$\left\{ \begin{array}{l} \frac{dp_0}{dt} = \mu p_1 - \lambda p_0, \\ \frac{dp_1}{dt} = \lambda p_0 + 2\mu p_2 - (\lambda + \mu)p_1, \\ \frac{dp_2}{dt} = \lambda p_1 + 3\mu p_3 - (\lambda + 2\mu)p_2, \\ \dots \\ \frac{dp_n}{dt} = \lambda p_{n-1} + n\mu p_{n+1} - (\lambda + n\mu)p_n, \\ \dots \\ \frac{dp_{n+m-1}}{dt} = \lambda p_{n+m-2} + n\mu p_{n+m} - (\lambda + n\mu)p_{n+m-1}, \\ \frac{dp_{n+m}}{dt} = \lambda p_{n+m-1} - n\mu p_{n+m}. \end{array} \right. \quad (5.17)$$

Замість будь-якого з рівнянь у цій системі (наприклад, замість останнього) можна покласти умову (5.16).

Якщо початковим станом є стан  $S_0$ , то маємо початкові умови:  $p_0(0) = 1, p_1(0) = 0, p_2(0) = 0, \dots, p_{n+m}(0) = 0$ .

Ймовірності станів  $p_k(t)$  при  $t \rightarrow \infty$  прямують до *фінальних імовірностей*  $p_k, k = 0, 1, 2, \dots, n + m$ .

Для фінальних імовірностей з системи диференціальних рівнянь (5.17) і умови (5.16) отримуємо систему лінійних ал-

гебричних рівнянь

$$\left\{ \begin{array}{l} \mu p_1 - \lambda p_0 = 0, \\ \lambda p_0 + 2\mu p_2 - (\lambda + \mu)p_1 = 0, \\ \lambda p_1 + 3\mu p_3 - (\lambda + 2\mu)p_2 = 0, \\ \dots \\ \lambda p_{n-1} + n\mu p_{n+1} - (\lambda + n\mu)p_n = 0, \\ \dots \\ \lambda p_{n+m-2} + n\mu p_{n+m} - (\lambda + n\mu)p_{n+m-1} = 0, \\ p_0 + p_1 + \dots + p_{n+m} = 1. \end{array} \right. \quad (5.18)$$

Нехай  $\alpha = \frac{\lambda}{\mu}$  — *зведена інтенсивність вимог*. Тоді, враховуючи, що перші  $n$  рівнянь системи (5.18) такі самі, як і в системі (5.8), маємо:

$$p_k = \frac{\alpha^k}{k!} p_0, \quad k = 1, 2, \dots, n. \quad (5.19)$$

З  $(n+1)$ -го рівняння системи (5.18) отримуємо:

$$n\mu p_{n+1} = (\lambda + n\mu) \frac{\alpha^n}{n!} p_0 - \lambda \frac{\alpha^{n-1}}{(n-1)!} p_0 \Rightarrow p_{n+1} = \frac{\alpha}{n} \frac{\alpha^n}{n!} p_0.$$

Припустимо, що

$$p_{n+k-1} = \left(\frac{\alpha}{n}\right)^{k-1} \frac{\alpha^n}{n!} p_0.$$

Тоді, враховуючи принцип математичної індукції, знаходимо:

$$\begin{aligned} n\mu p_{n+k} &= (\lambda + n\mu) \left(\frac{\alpha}{n}\right)^{k-1} \frac{\alpha^n}{n!} p_0 - \lambda \left(\frac{\alpha}{n}\right)^{k-2} \frac{\alpha^n}{n!} p_0 \Rightarrow \\ p_{n+k} &= \frac{\alpha^k}{n^k} \frac{\alpha^n}{n!} p_0, \quad k = 1, 2, \dots, m. \end{aligned} \quad (5.20)$$

Підставляючи отримані вирази для фінальних імовірностей в останнє рівняння системи (5.18) (тобто умову (5.16)), отримуємо:

$$p_0 \left( 1 + \alpha + \frac{\alpha^2}{2!} + \dots + \frac{\alpha^{n-1}}{(n-1)!} + \frac{\alpha^n}{n!} \left( 1 + \frac{\alpha}{n} + \frac{\alpha^2}{n^2} + \dots + \frac{\alpha^m}{n^m} \right) \right) = 1,$$

звідки для  $\alpha \neq n$

$$p_0 = \left( 1 + \alpha + \frac{\alpha^2}{2!} + \dots + \frac{\alpha^{n-1}}{(n-1)!} + \frac{\alpha^n}{n!} \frac{1 - \left(\frac{\alpha}{n}\right)^{m+1}}{1 - \frac{\alpha}{n}} \right)^{-1} \Rightarrow$$

$$p_0 = \left( 1 + \alpha + \frac{\alpha^2}{2!} + \dots + \frac{\alpha^{n-1}}{(n-1)!} + \frac{\alpha^n}{n!} \frac{n^{m+1} - \alpha^{m+1}}{n^m(n - \alpha)} \right)^{-1}. \quad (5.21)$$

Для  $\alpha = n$  виникає формула

$$p_0 = \left( 1 + \alpha + \frac{\alpha^2}{2!} + \dots + \frac{\alpha^{n-1}}{(n-1)!} + \frac{\alpha^n}{n!} (m+1) \right)^{-1}.$$

Якщо всі канали зайняті і в черзі перебуває  $m$  вимог, то чергова вимога не буде обслуговуватись системою й отримає відмову. Тому *ймовірність відмови* системи

$$P_{\text{відм}} = p_{n+m} = \frac{\alpha^{n+m}}{n! n^m} p_0. \quad (5.22)$$

*Імовірність обслуговування* вимоги

$$P_{\text{обсл}} = 1 - P_{\text{відм}} = 1 - \frac{\alpha^{n+m}}{n! n^m} p_0. \quad (5.23)$$

*Пропускна здатність*  $A$  — це середня кількість вимог, обслугованих системою за одиницю часу:

$$A = \lambda P_{\text{обсл}}. \quad (5.24)$$

Знайдемо *середню кількість зайнятих каналів* для  $\alpha \neq n$ :

$$N_3 = p_1 + 2p_2 + \dots + np_n + np_{n+1} + \dots + np_{n+m} \Rightarrow$$

$$N_3 = p_0 \left( \alpha + 2 \frac{\alpha^2}{2!} + 3 \frac{\alpha^3}{3!} + \dots + n \frac{\alpha^n}{n!} + \right.$$

$$\left. + n \frac{\alpha^n}{n!} \frac{\alpha}{n} \left( 1 + \frac{\alpha}{n} + \dots + \frac{\alpha^{m-1}}{n^{m-1}} \right) \right) \Rightarrow$$

$$N_3 = p_0 \alpha \left( 1 + \alpha + \frac{\alpha^2}{2!} + \dots + \frac{\alpha^{n-1}}{(n-1)!} + \frac{\alpha^n}{n!} \frac{n^m - \alpha^m}{n^{m-1}(n - \alpha)} \right). \quad (5.25)$$

Для  $\alpha = n$  маємо формулу:

$$N_3 = p_0 \alpha \left( 1 + \alpha + \frac{\alpha^2}{2!} + \dots + \frac{\alpha^{n-1}}{(n-1)!} + \frac{\alpha^n}{n!} m \right).$$

*Коефіцієнт завантаження одного каналу*

$$K_3 = \frac{N_3}{n}. \quad (5.26)$$

Для того щоб знайти середню довжину черги, спочатку доведемо методом математичної індукції допоміжну формулу:

$$\sum_{k=1}^m kq^k = \frac{q^{m+1}(mq - m - 1) + q}{(1 - q)^2}, \quad q \neq 1. \quad (5.27)$$

Справді, для  $m = 1$  рівність (5.27) виконується:

$$q = \frac{q^2(q - 2) + q}{(1 - q)^2}.$$

Припустимо, що формула (5.27) виконується для  $m = s$ :

$$\sum_{k=1}^s kq^k = \frac{q^{s+1}(sq - s - 1) + q}{(1 - q)^2}.$$

Доведемо її для  $m = s + 1$ :

$$\begin{aligned} \sum_{k=1}^{s+1} kq^k &= \frac{q^{s+1}(sq - s - 1) + q}{(1 - q)^2} + (s + 1)q^{s+1} = \\ &= \frac{q^{s+1}(sq - s - 1) + q + (s + 1)q^{s+1}(1 - 2q + q^2)}{(1 - q)^2} = \\ &= \frac{q^{s+2}((s + 1)q - s - 2) + q}{(1 - q)^2}. \end{aligned}$$

Шукаємо середню кількість вимог у черзі:

$$L_{\text{ч}} = p_{n+1} + 2p_{n+2} + \dots + mp_{n+m} \Rightarrow$$

$$L_{\text{ч}} = \frac{\alpha^n}{n!} \left( \frac{\alpha}{n} + 2\frac{\alpha^2}{n^2} + 3\frac{\alpha^3}{n^3} + \dots + m\frac{\alpha^m}{n^m} \right) p_0.$$

Для  $\alpha \neq n$  маємо:

$$L_{\text{ч}} = \frac{\alpha^n \frac{\alpha^{m+1}}{n^{m+1}} (m\frac{\alpha}{n} - m - 1) + \frac{\alpha}{n}}{n! (1 - \frac{\alpha}{n})^2} p_0 \Rightarrow$$

$$L_{\text{ч}} = \frac{\alpha^{n+1} \alpha^m (m\alpha - mn - n) + n^{m+1}}{n! n^m (n - \alpha)^2} p_0.$$

Якщо  $\alpha = n$ , то

$$L_{\text{ч}} = \frac{\alpha^n m(m+1)}{2 \cdot n!} p_0.$$

Отже, *середня кількість вимог у черзі* подається формулою

$$L_{\text{ч}} = \begin{cases} \frac{\alpha^{n+1}}{n!} \cdot \frac{\alpha^m (m\alpha - mn - n) + n^{m+1}}{n^m (n - \alpha)^2} p_0, & \alpha \neq n, \\ \frac{\alpha^n m(m+1)}{2 \cdot n!} p_0, & \alpha = n. \end{cases} \quad (5.28)$$

Згідно з формулою Літгла *середній час очікування в черзі*

$$W_{\text{ч}} = \frac{L_{\text{ч}}}{\lambda}. \quad (5.29)$$

Зокрема, для  $n = 1$  маємо:

$$p_0 = \begin{cases} \frac{1-\alpha}{1-\alpha^{m+2}}, & \alpha \neq 1, \\ \frac{1}{m+2}, & \alpha = 1, \end{cases} \quad P_{\text{вїдм}} = \alpha^{m+1} p_0,$$

$$N_3 = K_3 = \begin{cases} p_0 \alpha \frac{1-\alpha^{m+1}}{1-\alpha}, & \alpha \neq 1, \\ p_0 \alpha (1 + \alpha m), & \alpha = 1, \end{cases}$$

$$L_{\text{ч}} = \begin{cases} \frac{\alpha^m (m\alpha - m - 1) + 1}{(1-\alpha)^2} \alpha^2 p_0, & \alpha \neq 1, \\ \frac{m(m+1)}{2} p_0, & \alpha = 1. \end{cases}$$

**Приклад.** У перукарні є три перукарі. Час обслуговування клієнта кожним з них є випадковою величиною з експоненціальним законом розподілу з середнім значенням 10 хвилин. Клієнти часом своєї появи утворюють стаціонарний пуассонівський потік з інтенсивністю  $\lambda = 1/5$ . Якщо всі перукарі зайняті, а довжина черги не перевищує п'ять осіб, то клієнт стає в спільну чергу. Якщо в черзі вже є шість осіб, то черговий клієнт йде в іншу перукарню. Знайти ймовірність того, що клієнт піде в іншу перукарню, пропускну здатність системи, середню кількість зайнятих перукарів, коефіцієнт завантаження перукаря, середню довжину черги і середній час очікування в черзі.

**Розв'язання.** З курсу теорії ймовірностей відомо, що інтенсивність потоку подій  $\lambda$  для експоненціального закону розподілу проміжків між подіями є оберненою величиною до середнього значення. Тому в цій задачі  $\lambda = 1/5$ ,  $\mu = 1/10$ ,  $n = 3$ ,  $m = 6$ . Отже, зведена інтенсивність  $\alpha = 2$ , а за формулами (5.21)–(5.26), (5.28), (5.29) маємо:

$$\begin{aligned}
 p_0 &= \left( 1 + 2 + \frac{2^2}{2!} + \frac{2^3}{3!} \cdot \frac{3^7 - 2^7}{3^6(3-2)} \right)^{-1} = \frac{2187}{19171}, \\
 P_{\text{відм}} &= \frac{2^9}{3!3^6} p_0 = \frac{256}{19171} \approx 0,0134, \\
 A &= \frac{1}{5} (1 - P_{\text{відм}}) = \frac{3783}{19171} \approx 0,197, \\
 N_3 &= 2p_0 \left( 1 + 2 + \frac{2^2}{2!} + \frac{2^3}{3!} \cdot \frac{3^6 - 2^6}{3^5(3-2)} \right) = \frac{37830}{19171} \approx 1,97, \\
 K_3 &= \frac{N_3}{3} = \frac{12610}{19171} \approx 0,658, \\
 L_{\text{ч}} &= \frac{2^4}{3!} \cdot \frac{2^6(6 \cdot 2 - 6 \cdot 3 - 3) + 3^7}{3^6(3-2)^2} p_0 = \frac{12888}{19171} \approx 0,672, \\
 W_{\text{ч}} &= 5L_{\text{ч}} = \frac{64440}{19171} \approx 3,36. \blacksquare
 \end{aligned}$$

### § 5.9. Багатоканальні системи масового обслуговування з чергою довільної довжини

Розглянемо багатоканальну систему масового обслуговування типу  $M/M/n/\infty$  з безліччю позицій для очікування в черзі. Якщо зайняті всі  $n$  каналів, то вимога стає у хвіст черги. Прикладом системи може бути торговельна точка або частина виробництва.

Зробимо наступні припущення:

- вхідний потік є стаціонарним пуассонівським з інтенсивністю  $\lambda$ ;
- всі канали є однотипними;
- час обслуговування вимоги в каналі є випадковим і утворює стаціонарний пуассонівський потік з інтенсивністю  $\mu$ .

Система масового обслуговування може мати, взагалі кажучи, безмежну, але зліченну кількість станів:  $S_0$  — всі канали вільні,  $S_1$  — зайнято один канал (не важливо, який саме),  $S_2$  — зайнято два канали (не важливо, які саме) і т. д.,  $S_n$  — всі канали зайнято, але черга є порожньою,  $S_{n+k}$  — всі канали зайнято, а в черзі перебуває  $k$  вимог.

Оскільки вхідний потік є пуассонівським, а час обслуговування утворює пуассонівський потік, то система може переходити з одного стану в інші послідовно: з  $S_0$  в  $S_1$ , з  $S_1$  в  $S_0$  або в  $S_2$  і т. д. (рис. 5.6).

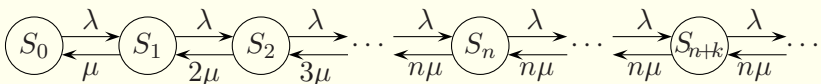


Рис. 5.6.

Вихідний потік тут також є стаціонарним пуассонівським з інтенсивністю  $\lambda^* = \min\{\lambda, n\mu\}$ .

Нехай  $p_i(t)$  — ймовірність  $i$ -го стану, тобто ймовірність того, що в момент часу  $t$  система перебуває в стані  $S_i$ ,

$i = 0, 1, 2, \dots$  Для ймовірностей станів має виконуватись умова

$$\sum_{i=0}^{\infty} p_i(t) = 1. \quad (5.30)$$

Аналогічно § 5.7 будується система диференціальних рівнянь Колмогорова

$$\left\{ \begin{array}{l} \frac{dp_0}{dt} = \mu p_1 - \lambda p_0, \\ \frac{dp_1}{dt} = \lambda p_0 + 2\mu p_2 - (\lambda + \mu)p_1, \\ \frac{dp_2}{dt} = \lambda p_1 + 3\mu p_3 - (\lambda + 2\mu)p_2, \\ \dots \\ \frac{dp_n}{dt} = \lambda p_{n-1} + n\mu p_{n+1} - (\lambda + n\mu)p_n, \\ \dots \\ \frac{dp_{n+k}}{dt} = \lambda p_{n+k-1} + n\mu p_{n+k+1} - (\lambda + n\mu)p_{n+k}, \\ \dots \end{array} \right. \quad (5.31)$$

Якщо початковим станом є стан  $S_0$ , то маємо початкові умови:  $p_0(0) = 1$ ,  $p_k(0) = 0$ ,  $k = 1, 2, 3, \dots$

Припустимо, що ймовірності станів  $p_k(t)$  при  $t \rightarrow \infty$  прянують до *фінальних ймовірностей*  $p_k$ ,  $k = 0, 1, 2, \dots$

Для фінальних ймовірностей з системи диференціальних рівнянь (5.31) отримуємо систему лінійних алгебричних рівнянь

$$\left\{ \begin{array}{l} \mu p_1 - \lambda p_0 = 0, \\ \lambda p_0 + 2\mu p_2 - (\lambda + \mu)p_1 = 0, \\ \lambda p_1 + 3\mu p_3 - (\lambda + 2\mu)p_2 = 0, \\ \dots \\ \lambda p_{n-1} + n\mu p_{n+1} - (\lambda + n\mu)p_n = 0, \\ \dots \\ \lambda p_{n+k-1} + n\mu p_{n+k+1} - (\lambda + n\mu)p_{n+k} = 0, \\ \dots \end{array} \right. \quad (5.32)$$

Нехай  $\alpha = \frac{\lambda}{\mu}$  — *зведена інтенсивність вимог*. Тоді, враховуючи, що система (5.32) відрізняється від системи (5.18) лише тим, що має більшу кількість однотипних рівнянь, і повторюючи міркування § 5.8, отримуємо:

$$p_k = \frac{\alpha^k}{k!} p_0, \quad k = 1, 2, \dots, n. \quad (5.33)$$

$$p_{n+k} = \frac{\alpha^k \alpha^n}{n^k n!} p_0, \quad k = 1, 2, \dots \quad (5.34)$$

Підставляючи отримані вирази для фінальних імовірностей в умову нормування (5.30), маємо:

$$p_0 \left( 1 + \alpha + \frac{\alpha^2}{2!} + \frac{\alpha^3}{3!} + \dots + \frac{\alpha^{n-1}}{(n-1)!} + \frac{\alpha^n}{n!} \left( 1 + \frac{\alpha}{n} + \frac{\alpha^2}{n^2} + \dots \right) \right) = 1. \quad (5.35)$$

Безмежна сума в дужках у формулі (5.35) утворює геометричну прогресію зі знаменником  $\frac{\alpha}{n}$ . Геометрична прогресія має скінченну суму, якщо її знаменник

$$\frac{\alpha}{n} < 1. \quad (5.36)$$

Тоді

$$p_0 = \left( 1 + \alpha + \frac{\alpha^2}{2!} + \dots + \frac{\alpha^{n-1}}{(n-1)!} + \frac{\alpha^n}{n!} \frac{1}{1 - \frac{\alpha}{n}} \right)^{-1} \Rightarrow$$

$$p_0 = \left( 1 + \alpha + \frac{\alpha^2}{2!} + \dots + \frac{\alpha^{n-1}}{(n-1)!} + \frac{\alpha^n}{(n-1)!(n-\alpha)} \right)^{-1}. \quad (5.37)$$

Припустимо, що виконується умова (5.36). Тоді *середня кількість зайнятих каналів*

$$\begin{aligned}
N_3 &= p_1 + 2p_2 + \dots + np_n + np_{n+1} + \dots + np_{n+k} + \dots \Rightarrow \\
N_3 &= p_0 \left( \alpha + 2\frac{\alpha^2}{2!} + 3\frac{\alpha^3}{3!} + \dots + n\frac{\alpha^n}{n!} + \right. \\
&\quad \left. + n\frac{\alpha^n}{n!} \frac{\alpha}{n} \left( 1 + \frac{\alpha}{n} + \frac{\alpha^2}{n^2} + \dots \right) \right) \Rightarrow \\
N_3 &= p_0 \alpha \left( 1 + \alpha + \frac{\alpha^2}{2!} + \dots + \frac{\alpha^{n-1}}{(n-1)!} + \frac{\alpha^n}{(n-1)!(n-\alpha)} \right) \Rightarrow \\
N_3 &= \alpha. \tag{5.38}
\end{aligned}$$

*Коефіцієнт завантаження одного каналу*

$$K_3 = \frac{\alpha}{n}. \tag{5.39}$$

Для виведення *середньої кількості вимог у черзі* скористаємось формулою (5.27), з якої для  $|q| < 1$  випливає, що

$$\sum_{k=1}^{\infty} kq^k = \frac{q}{(1-q)^2}.$$

Тоді

$$\begin{aligned}
L_{\text{ч}} &= p_{n+1} + 2p_{n+2} + \dots + kp_{n+k} + \dots \Rightarrow \\
L_{\text{ч}} &= \frac{\alpha^n}{n!} \left( \frac{\alpha}{n} + 2\frac{\alpha^2}{n^2} + 3\frac{\alpha^3}{n^3} + \dots + k\frac{\alpha^k}{n^k} + \dots \right) p_0 = \\
&= \frac{\alpha^n}{n!} \frac{\frac{\alpha}{n}}{\left(1 - \frac{\alpha}{n}\right)^2} p_0 \Rightarrow \\
L_{\text{ч}} &= \frac{\alpha^{n+1}}{(n-1)!(n-\alpha)^2} p_0. \tag{5.40}
\end{aligned}$$

Згідно з формулою Літла *середній час очікування в черзі*

$$W_{\text{ч}} = \frac{L_{\text{ч}}}{\lambda}. \tag{5.41}$$

Зокрема, для  $n = 1$  при виконанні умови  $\alpha < 1$  маємо:

$$p_0 = 1 - \alpha, \quad N_3 = K_3 = \alpha, \quad L_{\text{ч}} = \frac{\alpha^2}{1 - \alpha}.$$

Якщо  $\frac{\alpha}{n} \geq 1$ , то з формули (5.35) випливає, що  $p_0 = 0$  і, отже,  $p_k = 0$  для кожного скінченного  $k$ . Це означає, що у випадку  $\frac{\alpha}{n} \geq 1$  довжина черги і середній час очікування в черзі при  $t \rightarrow \infty$  прямують до нескінченності, коефіцієнт завантаження одного каналу прямує до одиниці, а середня кількість зайнятих каналів — до  $n$ .

**Приклад.** У перукарні є три перукарі. Час обслуговування клієнта кожним з них є випадковою величиною з експоненціальним законом розподілу з середнім значенням 10 хвилин. Клієнти часом своєї появи утворюють стаціонарний пуассонівський потік з інтенсивністю  $\lambda = 1/5$ . Знайти середню кількість зайнятих перукарів, коефіцієнт завантаження перукаря, середню довжину черги і середній час очікування в черзі.

**Розв'язання.** З курсу теорії ймовірностей відомо, що інтенсивність потоку подій  $\lambda$  для експоненціального закону розподілу проміжків між подіями є оберненою величиною до середнього значення. Тому в цій задачі  $\lambda = 1/5$ ,  $\mu = 1/10$ ,  $n = 3$ . Отже, зведена інтенсивність  $\alpha = 2$ , а оскільки  $\frac{\alpha}{n} = \frac{2}{3} < 1$ , то черга є скінченною і за формулами (5.37)–(5.41) маємо:

$$p_0 = \left( 1 + 2 + \frac{2^2}{2!} + \frac{2^3}{2!(3-2)} \right)^{-1} = \frac{1}{9},$$

$$N_3 = 2, \quad K_3 = \frac{2}{3} \approx 0,667,$$

$$L_{\text{ч}} = \frac{2^4}{2!(3-2)^2} \frac{1}{9} = \frac{8}{9} \approx 0,889, \quad W_{\text{ч}} = 5L_{\text{ч}} = \frac{40}{9} \approx 4,44. \quad \blacksquare$$

### § 5.10. Основи операційного аналізу мереж систем масового обслуговування

У попередніх параграфах розглядалися моделі простих систем масового обслуговування. Проте зустрічаються складніші структури — мережі систем масового обслуговування, вузлами яких є прості системи масового обслуговування. Найпростіша мережа утворюється шляхом послідовного з'єднання кількох систем масового обслуговування. Таку мережу називають багатофазною. Разом з тим, існують більш складні мережі. На рис. 5.7 наведено приклад мережі, що складається з чотирьох вузлів. Зовнішнє середовище позначено нулем. Стрілки вказують напрям руху вимог.

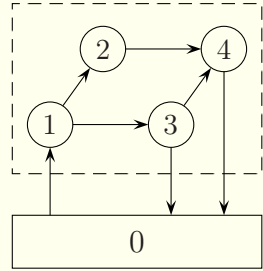


Рис. 5.7.

Операційний аналіз мереж систем масового обслуговування можна застосовувати до мереж, для яких існує баланс потоків вимог: кількість вимог, які залишають мережу протягом певного проміжку часу, має дорівнювати кількості вимог, що потрапили в мережу за цей самий проміжок часу. Це означає, що операційний аналіз можна застосовувати для дослідження мереж, які працюють з перевантаженням, тобто в них завжди є черга вимог. У цьому разі можна вважати, що замість вимоги, яка залишила мережу, до неї надходить інша вимога.

Крім того, для застосування операційного аналізу до мереж систем масового обслуговування припускають, що процес надходження вимог від одного вузла до іншого не повинен залежати від довжини черг у вузлах і часу закінчення обслуговування вимог у них.

Операційний аналіз часто використовують для визначення коефіцієнтів завантаження вузлів мережі та аналізу вузьких місць у ній.

Введемо основні операційні змінні, значення яких можливо отримати шляхом безпосереднього вимірювання параметрів реальної системи:

–  $q_{0j}$  — імовірність надходження вимог з зовнішнього середовища до  $j$ -го вузла мережі,  $j = 1, 2, \dots, n$ , де  $n$  — загальна кількість вузлів;

–  $q_{kj}$  — імовірність переходу вимог від  $k$ -го до  $j$ -го вузла мережі,  $k, j = 1, 2, \dots, n$ ;

–  $q_{k0}$  — імовірність того, що після закінчення обслуговування у  $k$ -му вузлі вимоги залишать мережу,  $k = 1, 2, \dots, n$ ;

–  $A_k$  — кількість вимог, які надійшли до  $k$ -го вузла за час спостереження за системою,  $k = 1, 2, \dots, n$ ;

–  $A_{0j}$  — кількість вимог, які надійшли з зовнішнього середовища до  $j$ -го вузла за час спостереження за системою,  $j = 1, 2, \dots, n$ ;

–  $C_{kj}$  — кількість вимог, які залишили  $k$ -й вузол і перейшли до  $j$ -го вузла мережі за час спостереження за системою,  $k, j = 1, 2, \dots, n$ ;

–  $C_{k0}$  — кількість вимог, які залишили  $k$ -й вузол і перейшли до зовнішнього середовища за час спостереження за системою,  $k = 1, 2, \dots, n$ ;

–  $T$  — загальний час спостереження за системою;

–  $B_k$  — загальний час обслуговування вимог у  $k$ -му вузлі,  $k = 1, 2, \dots, n$  (вузол вважається зайнятим, якщо в ньому перебуває хоч одна вимога).

Як видно, зовнішнє середовище тут позначається нулем. Введемо додаткові операційні змінні:

$$C_k = \sum_{j=0}^n C_{kj}, \quad A_0 = \sum_{j=1}^n A_{0j}, \quad C_0 = \sum_{k=1}^n C_{k0},$$

де  $C_k$  — кількість вимог, які залишили  $k$ -й вузол,  $A_0$  — кількість вимог, які надійшли до вузлів мережі із зовнішнього середовища,  $C_0$  — кількість вимог, які залишили мережу.

Крім того, розглядають наступні додаткові операційні змінні:

- інтенсивність надходження вимог до  $k$ -го вузла

$$\lambda_k = \frac{A_k}{T};$$

- коефіцієнт використання  $k$ -го вузла

$$U_k = \frac{B_k}{T}; \quad (5.42)$$

- середній час обслуговування у  $k$ -му вузлі

$$S_k = \frac{B_k}{C_k}; \quad (5.43)$$

- інтенсивність вихідного потоку вимог з  $k$ -го вузла

$$X_k = \frac{C_k}{T}. \quad (5.44)$$

Основні співвідношення операційного аналізу формулюються у вигляді операційних залежностей. Розглянемо деякі з них.

Ймовірність переміщення вимог між вузлами можна подати формулою

$$q_{kj} = \begin{cases} \frac{C_{kj}}{C_k}, & k = 1, 2, \dots, n, \quad j = 0, 1, \dots, n, \\ \frac{A_{0j}}{A_0}, & k = 0, \quad j = 1, 2, \dots, n. \end{cases} \quad (5.45)$$

Крім того, для ймовірностей переходів між вузлами мають справджуватись формули:

$$\sum_{k=0}^n q_{kj} = 1, \quad \sum_{j=0}^n q_{kj} = 1.$$

Використовуючи формули (5.42)–(5.44), маємо:

$$U_k = X_k S_k. \quad (5.46)$$

Останній вираз — це закон коефіцієнта використання вузла.

Рівність  $A_0 = C_0$  є ознакою балансу вхідного і вихідного потоків вимог для мережі. Рівності  $A_j = C_j$ ,  $j = 1, 2, \dots, n$ , є ознаками балансів вхідних і вихідних потоків вимог для вузлів мережі.

У припущенні, що  $A_j = C_j$ ,  $j = 0, 1, \dots, n$ , має місце *система балансу потоків вимог*

$$X_j = \sum_{k=0}^n X_k q_{kj}, \quad j = 0, 1, \dots, n. \quad (5.47)$$

Доведемо систему (5.47). З рівностей

$$A_j = A_{0j} + \sum_{k=1}^n C_{kj}, \quad j = 0, 1, \dots, n,$$

враховуючи, що  $A_j = C_j$ , маємо:

$$C_j = A_{0j} + \sum_{k=1}^n C_{kj}.$$

З формули (5.45) випливає, що  $C_{kj} = C_k q_{kj}$ ,  $A_{0j} = A_0 q_{0j}$ . Отже,

$$C_j = A_0 q_{0j} + \sum_{k=1}^n C_k q_{kj} \Rightarrow C_j = \sum_{k=0}^n C_k q_{kj}, \quad j = 0, 1, \dots, n.$$

Поділивши всі рівняння останньої системи на  $T$ , отримаємо систему (5.47), що й потрібно було довести.

**Приклад.** На рис. 5.8 подано мережу систем масового обслуговування, яка працює з перевантаженням. Вузли мережі позначено квадратиками з їхніми номерами, для позначення черг перед вузлами використовуються символи  $\square$ . Середній час (у секундах) обслуговування вимог у вузлах мережі є наступним:  $S_1 = 0,05$ ,  $S_2 = 0,03$ ,  $S_3 = 0,09$ ,  $S_4 = 0,01$ . Проаналізувати вузькі місця мережі і визначити найбільшу кількість

вимог, яку мережа зможе опрацювати за одиницю часу, тобто максимальну інтенсивність вхідного потоку. Обчислити коефіцієнти використання всіх вузлів мережі і вихідні потоки з усіх вузлів для знайденого вхідного потоку.

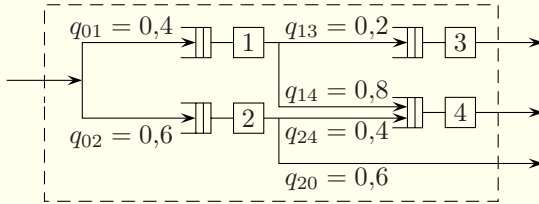


Рис. 5.8.

**Розв'язання.** Будуємо систему балансу (5.47) для мережі з рис. 5.8:

$$\begin{cases} X_0 = X_3 + X_4 + 0,6X_2, \\ X_1 = 0,4X_0, \\ X_2 = 0,6X_0, \\ X_3 = 0,2X_1, \\ X_4 = 0,8X_1 + 0,4X_2 \end{cases} \Rightarrow \begin{cases} X_1 = 0,4X_0, \\ X_2 = 0,6X_0, \\ X_3 = 0,08X_0, \\ X_4 = 0,56X_0. \end{cases}$$

За формулами (5.46) визначаємо коефіцієнти використання вузлів через інтенсивність вхідного потоку  $X_0$ :

$$U_1 = 0,02X_0, \quad U_2 = 0,018X_0, \quad U_3 = 0,0072X_0, \quad U_4 = 0,0056X_0.$$

Найбільшим є коефіцієнт використання першого вузла, тому саме перший вузол є вузьким місцем даної мережі. Оскільки коефіцієнт використання вузла не може бути більшим від 1, то з рівності  $0,02X_0 = 1$  отримуємо  $X_0 = 50$  — максимальну інтенсивність вхідного потоку. Тепер можна обчислити відповідні інтенсивності вихідних потоків і коефіцієнти використання всіх вузлів:

$$\begin{aligned} X_1 &= 20, & X_2 &= 30, & X_3 &= 4, & X_4 &= 28, \\ U_1 &= 1, & U_2 &= 0,9, & U_3 &= 0,36, & U_4 &= 0,28. \quad \blacksquare \end{aligned}$$

Інші способи аналітичного моделювання систем масового обслуговування можна знайти в книгах [1, 3, 13].

### Питання до розділу 5

1. Що таке потік подій? Який потік подій називається стаціонарним пуассонівським? Як моделюється стаціонарний пуассонівський потік подій?

2. Назвіть основні принципи організації черги і правила обслуговування систем масового обслуговування.

3. У чому полягає формула Літгла?

4. За якими формулами можна обчислити основні характеристики системи масового обслуговування з відмовами (з обмеженою чергою)?

5. За якими формулами можна обчислити основні характеристики системи масового обслуговування з чергою довільної довжини? В якому випадку довжина черги в системі масового обслуговування прямуватиме до безмежності?

6. В якому випадку можна застосовувати операційний аналіз до мереж систем масового обслуговування? Який вигляд має система балансу для мережі систем масового обслуговування?

### Вправи до розділу 5

1. Задана двоканальна система масового обслуговування з двома пристроями (Пр1 і Пр2) з двома позиціями для чекання в черзі (Поз1 і Поз2). Її часову діаграму наведено на рис. 5.9. Час надходження вимоги до системи і час, коли вона залишила систему, наведено поряд з номером вимоги відповідно в нижній і верхній частинах рис. 5.9. Час вимірюється у хвилинах і для зручності заокруглений до цілого числа хвилин.

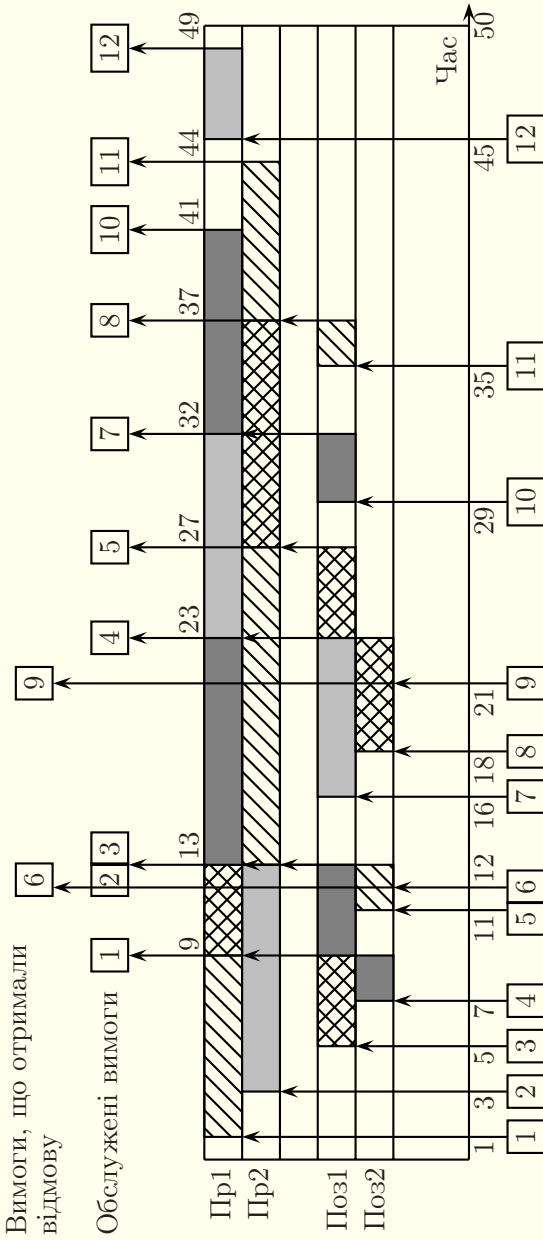


Рис. 5.9.

Знайдіть імовірність обслуговування вимоги, інтенсивність вхідного потоку, пропускну здатність системи, імовірність відмови в обслуговуванні, імовірність того, що вимога застане обидва пристрої вільними, імовірність того, що обслуговуванням зайнятий тільки один пристрій з двох, імовірність того, що обслуговуванням зайняті обидва пристрої, але в черзі відсутні вимоги, імовірність того, що в черзі є лише одна вимога, імовірність того, що в черзі знаходиться дві вимоги, середню кількість пристроїв, зайнятих обслуговуванням, середню кількість вимог у черзі, середній час перебування вимог у черзі, середній час перебування вимог у черзі без врахування вимог, які не чекали, середній час обслуговування вимоги пристроями, загальний середній час перебування вимоги в системі масового обслуговування, середню кількість вимог у системі масового обслуговування.

2. Система має чотири стани  $S_1, S_2, S_3, S_4$ . Імовірності перебування у цих станах дорівнюють  $p_1(t), p_2(t), p_3(t), p_4(t)$  відповідно. Розмічений граф станів зображено на рис. 5.10. Перехід системи зі стану  $i$  в стан  $j$  утворює стаціонарний пуассонівський потік з інтенсивністю  $\lambda_{ij}$ . Початковим є стан  $S_1$ . Складіть систему диференціальних рівнянь Колмогорова для ймовірностей  $p_1(t), p_2(t), p_3(t), p_4(t)$  і запишіть початкові умови. Побудуйте також систему для фінальних ймовірностей.

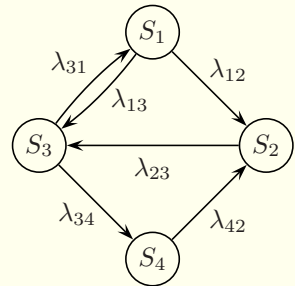


Рис. 5.10.

3. Розмічений граф станів зображено на рис. 5.11. Перехід системи зі стану  $i$  в стан  $j$  утворює стаціонарний пуассонівський потік з інтенсивністю  $\lambda_{ij}$ . Початковим є стан  $S_1$ . Складіть систему диференціальних рівнянь Колмогорова для ймовірностей перебування у станах  $p_1(t), p_2(t), p_3(t), p_4(t), p_5(t)$  і запишіть початкові умови. Побудуйте також систему для фінальних ймовірностей.

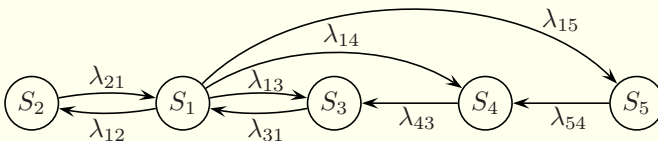


Рис. 5.11.

4. Розмічений граф станів зображено на рис. 5.12. Перехід системи зі стану  $i$  в стан  $j$  утворює стаціонарний пуассонівський потік з інтенсивністю  $\lambda_{ij}$ . Початковим є стан  $S_1$ . Складіть систему диференціальних рівнянь Колмогорова для ймовірностей перебування у станах  $p_1(t)$ ,  $p_2(t)$ ,  $p_3(t)$ ,  $p_4(t)$ ,  $p_5(t)$  і запишіть початкові умови. Побудуйте також систему для фінальних ймовірностей.

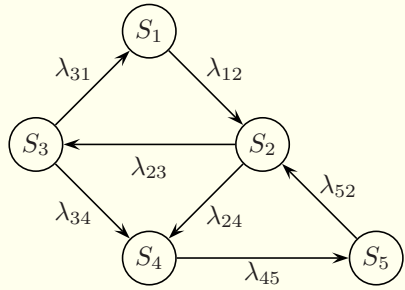


Рис. 5.12.

5. На виробничій дільниці є  $N$  однакових верстатів. Продуктивність одного верстата така, що в середньому протягом зміни на ньому можна обробити  $\mu$  деталей, а час обробки деталей утворює потік, близький до стаціонарного пуассонівського. Деталі надходять на обробку партіями по  $M$  штук. Партії надходять нерівномірно і їхній потік є близьким до стаціонарного пуассонівського з інтенсивністю  $\lambda$  партій за зміну. Побудуйте граф станів даної системи масового обслуговування і запишіть систему рівнянь Колмогорова для фінальних ймовірностей.

6. Автоматизована телефонна станція має 5 ліній зв'язку. Вхідний потік викликів є стаціонарним пуассонівським з інтенсивністю 2 виклики за хвилину. Тривалості телефонних розмов утворюють стаціонарний пуассонівський потік. Середня тривалість телефонної розмови становить 1 хвилину. Потрібно визначити ймовірність відмови, середню кількість зайнятих ліній зв'язку і коефіцієнт завантаження однієї лінії.

7. Автостоянка має 10 місць. Тривалість перебування автомобіля на стоянці є випадковою величиною з експоненціальним законом розподілу з середнім значенням 3 години. Інтенсивність вхідного стаціонарного пуассонівського потоку становить 3 машини за годину. Якщо вільних місць на стоянці немає, то автомобілі їдуть шукати іншу стоянку. Потрібно визначити ймовірність відмови, середню кількість зайнятих місць і коефіцієнт завантаження одного місця.

8. Визначте мінімальну потрібну кількість ліжок у стаціонарній лікарні, якщо час перебування в лікарні хворого є випадковою ве-

личною з експоненціальним законом розподілу з середнім значенням 14 днів. Нові хворі не приймаються, якщо всі ліжка в стаціонарі зайняті. Потік надходження хворих є близьким до стаціонарного пуассонівського з інтенсивністю 3 людини за день. Імовірність відмови має не перевищувати 5 %.

9. У майстерні, яка виконує ремонт телевізорів, працює троє майстрів. Вхідний потік замовлень на ремонт телевізорів є близьким до стаціонарного пуассонівського з інтенсивністю 1,3 телевізора за годину. У зв'язку з обмеженістю місця для зберігання замовлень на ремонт черга обмежена десятьма позиціями. Якщо в черзі перебуває 10 замовлень, то наступне замовлення отримує відмову і замовник ремонту йде в іншу майстерню. Час ремонту телевізора майстром є випадковою величиною з експоненціальним законом розподілу з середнім значенням 2 години. Визначте ймовірність відмови, середню кількість зайнятих роботою майстрів, коефіцієнт завантаження роботою кожного майстра, середню довжину черги, середній час перебування замовлення в черзі.

10. На автозаправці є дві бензоколонки. Майданчик поблизу автозаправки допускає одночасне очікування не більше  $m$  автомобілів. Потік автомобілів, які прибувають на заправку, є стаціонарним пуассонівським з інтенсивністю 0,2 автомобіля за хвилину. Час обслуговування автомобіля розподілений за експоненціальним законом з середнім значенням 7 хвилин. Визначте мінімальну кількість місць для очікування  $m$ , які забезпечать обслуговування не менше 95 % автомобілів, що потребують заправки. Для знайденого  $m$  обчисліть коефіцієнт завантаження бензоколонок, середню довжину черги і середній час перебування автомобіля в черзі.

11. По конвеєру надходять деталі двох видів:  $A$  і  $B$ . Іntenсивність надходження деталей однакова і становить  $\lambda_A = \lambda_B = 0,45$  деталей за хвилину, а їхній потік є близьким до стаціонарного пуассонівського. Для обробки деталей використовуються два верстати, які можуть обробляти як деталі виду  $A$ , так і деталі виду  $B$ . Середній час обробки деталей (як одного виду, так і іншого) становить 2 хвилини, причому час обробки утворює стаціонарний пуассонівський потік. Потрібно визначити коефіцієнт завантаження верстатів, середню довжину черги і середній час очікування в черзі.

12. Припустимо, що один раціоналізатор запропонував спеціалізувати кожен верстат з попередньої задачі тільки на один вид

деталей. У результаті час обробки деталей на кожному з верстатів зменшився. На верстаті, який обробляє деталі типу  $A$ , він зменшився в середньому до 1,9 хвилини, а на верстаті, який обробляє деталі типу  $B$ , — до 1,95 хвилини. Потрібно визначити коефіцієнти завантаження верстатів, середні довжини черг, середній час очікування в чергах, а також загальну довжину черги. Чи вдалою є така раціоналізаторська пропозиція?

13. У магазині є одна каса. Середній час обслуговування одного покупця в касі становить 0,3 хвилини, причому час обслуговування утворює стаціонарний пуассонівський потік. Потік покупців є близьким до пуассонівського з інтенсивністю 3 покупці за хвилину. Визначте коефіцієнт завантаження каси, середню довжину черги і середній час очікування в черзі.

14. Визначте, скільки в магазині з попередньої задачі необхідно встановити кас, якщо інтенсивність потоку покупців зростає втричі. Середня довжина спільної черги до всіх кас не повинна перевищувати 10 осіб.

15. На рис. 5.13 подано мережу систем масового обслуговування, яка працює з перевантаженням. Середній час (у секундах) обслуговування вимог у вузлах мережі є наступним:  $S_1 = 0,05$ ,  $S_2 = 0,08$ ,  $S_3 = 0,04$ . Проаналізуйте вузькі місця мережі і визначте найбільшу кількість вимог, яку мережа зможе опрацювати за одиницю часу, тобто максимальну інтенсивність вхідного потоку. Обчисліть коефіцієнти використання всіх вузлів мережі і вихідні потоки з усіх вузлів для знайденого вхідного потоку.

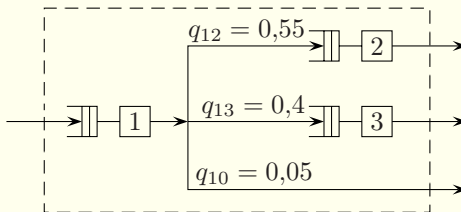


Рис. 5.13.

16. На рис. 5.14 подано мережу систем масового обслуговування, яка працює з перевантаженням. Середній час (у секундах) обслуговування вимог у вузлах мережі є наступним:  $S_1 = 0,03$ ,  $S_2 = 0,01$ ,  $S_3 = 0,09$ ,  $S_4 = 0,07$ . Проаналізуйте вузькі місця мережі

і визначте найбільшу кількість вимог, яку мережа зможе опрацювати за одиницю часу, тобто максимальну інтенсивність вхідного потоку. Обчисліть коефіцієнти використання всіх вузлів мережі і вихідні потоки з усіх вузлів для знайденого вхідного потоку.

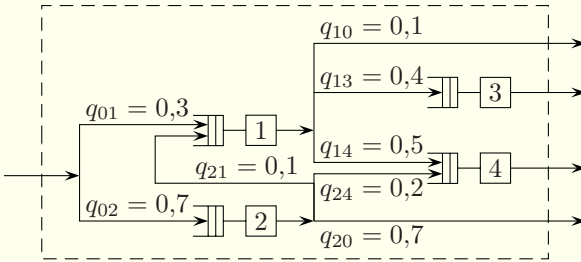


Рис. 5.14.

17. На рис. 5.15 подано мережу систем масового обслуговування, яка працює з перевантаженням. Середній час (у секундах) обслуговування вимог у вузлах мережі є наступним:  $S_1 = 0,02$ ,  $S_2 = 0,01$ ,  $S_3 = 0,04$ ,  $S_4 = 0,02$ ,  $S_5 = 0,03$ . Проаналізуйте вузькі місця мережі і визначте найбільшу кількість вимог, яку мережа зможе опрацювати за одиницю часу, тобто максимальну інтенсивність вхідного потоку. Обчисліть коефіцієнти використання всіх вузлів мережі і вихідні потоки з усіх вузлів для знайденого вхідного потоку.

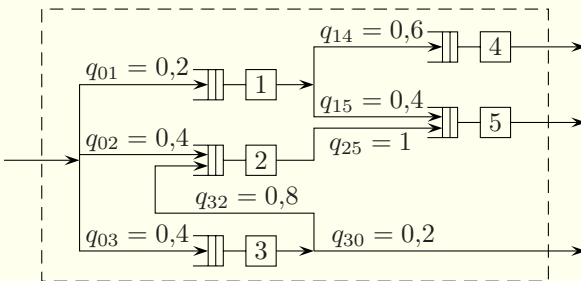


Рис. 5.15.

## Розділ 6. Основи мови імітаційного моделювання GPSS World

### § 6.1. Основи роботи з середовищем GPSS World

Імітаційне моделювання можна здійснювати з допомогою програм, написаних на універсальних мовах програмування, таких як Python, C++ та інші. Але набагато зручніше для цього використовувати спеціальні мови програмування моделей. Однією з найпоширеніших є мова GPSS, яка розшифровується як General Purpose System Simulation — система моделювання загального призначення. Ця мова була розроблена ще на початку 60-х років XX-го століття Джеффри Гордоном. У 2000 році фірмою Minuteman Software для операційної системи Windows було створено середовище GPSS World, студентська версія якого є безкоштовною.

Мову імітаційного моделювання GPSS World найчастіше використовують для моделювання систем масового обслуговування. Проте її успішно можна використовувати і для розв'язування ряду інших задач імітаційного моделювання. Можливість інтегрування диференціальних рівнянь паралельно з імітаційним моделюванням робить мову GPSS World придатною для застосування комбінованих методів математичного моделювання.

Модель на мові GPSS записується у вигляді програми — послідовності операторів. Великі і малі латинські букви в тексті програми не розрізняються. Написаний текст моделі (програми) транслюється у внутрішній код, а потім запускається на виконання. Відповідно можуть з'являтися повідомлення про помилки трансляції чи часу виконання. Якщо помилок немає, то відображається статистика змодельованого процесу, яка містить інформацію про час роботи, зайняті комірки пам'яті тощо. Ця статистика і є результатом роботи моделі. Її детальність визначається використаними в програмі спеціальними операторами. Як і в інших мовах програмування, є

можливість покрокового виконання програми, створення точок зупинки, перегляду змінних і т. п.

Робота з середовищем GPSS World суттєво не відрізняється від роботи з іншими програмами (рис. 6.1). Створення моделі здійснюється в окремому вікні. Для трансляції моделі і запуску її на виконання використовується команда меню **Command►Create Simulation**, цю команду можна також виконати, натиснувши комбінацію клавіш **Ctrl+Alt+S**. При кожній трансляції моделі створюється окреме вікно журналу з повідомленнями про результати трансляції і виявлені помилки часу трансляції і часу виконання (рис. 6.2). При кожному запуску процесу моделювання у вікні журналу відображаються повідомлення про хід процесу моделювання. Крім того, кожний успішний запуск моделі генерує ще одне вікно зі статистичним звітом (рис. 6.3). Переходити між вікнами можна звичайним способом з використанням меню **Window**.

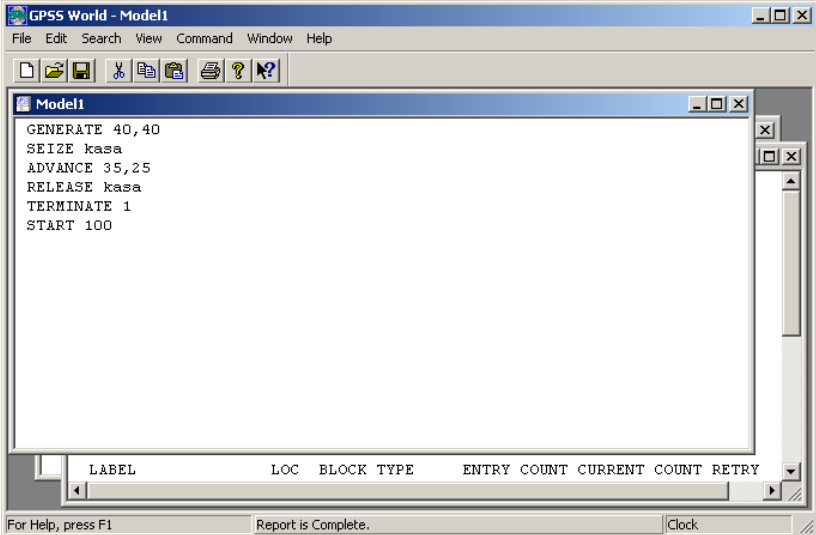


Рис. 6.1



Рис. 6.2

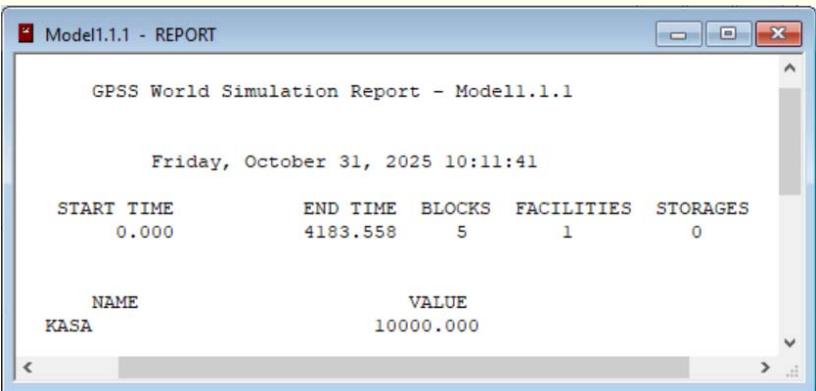


Рис. 6.3

За наявності помилок у моделі процес моделювання може продовжуватись необмежено довго. Перервати процес моделювання можна з допомогою команди меню **Command▶Halt**. Цю команду можна також викликати, натиснувши клавішу **F4**.

Всі повідомлення про помилки виводяться у вікні журналу з вказуванням номерів рядка і стовпця, назви помилки і відображенням рядка, який призвів до помилки. Команда меню **Search▶Next Error** відображає вікно з текстом моделі і виділяє помилковий (з точки зору транслятора) фрагмент рядка. Однак треба мати на увазі, що помилка насправді могла бути допущена у попередньому рядку програми. Якщо є кілька

помилки, то команда **Search▶Next Error** (або комбінація клавіш **Ctrl+Alt+N**) дозволяє переміщуватись між помилковими рядками.

Потрібно розрізняти модельний і реальний час моделювання. Зміст одиниці модельного часу визначає розробник моделі. Цією одиницею може бути секунда, хвилина, година тощо. Реальний час моделювання залежить від швидкодії комп'ютера. Початкове значення модельного часу — 0.

Програма складається з послідовності операторів, якими є блоки, команди опису і команди керування. Блоки є найважливішими складовими елементами програми. Вони формують саму модель і використовуються протягом всього процесу моделювання. Команди опису описують різні об'єкти мови GPSS і можуть розміщуватись у будь-якому місці програми і в будь-якому порядку. По аналогії з іншими мовами програмування їх часто наводять на початку програми. У програмі також можуть бути присутніми деякі команди керування. Їх краще розміщувати в кінці програми. Команди опису і команди керування виконуються лише по одному разу в тому порядку, в якому вони записані в програмі.

При роботі з GPSS World можна користуватись довідковою системою середовища імітаційного моделювання на англійській мові, до якої отримують доступ через меню **Help** або натисненням клавіші **F1**. Крім того, можна знайти електронну книгу **Reference Manual** у розділі **GPSS World Student Version** (меню **Пуск**). Можна також користуватись книгами [2, 11].

## § 6.2. Категорії і типи об'єктів мови GPSS

Моделі на мові GPSS є сукупностями окремих елементів, які називають об'єктами. Основні категорії об'єктів мови GPSS — операційна, динамічна, обчислювальна, запам'ятовуюча, апаратна і статистична.

До операційної категорії належить єдиний тип об'єктів —

*блоки*. Кожен блок має входи й виходи, з допомогою яких здійснюється зв'язок у моделі. Існує два особливих блоки: **GENERATE**, який має лише вихід, і **TERMINATE**, який має тільки вхід.

До динамічної категорії належать *транзакти*. Функціонування об'єкта відбувається в моделі у вигляді переміщення транзактів від блока **GENERATE** у блок **TERMINATE** через проміжні блоки. Транзакти (повідомлення) є абстрактними рухомими об'єктами, які, переміщуючись між блоками, викликають різні дії і зазнають їх. Транзакти послідовно входять у всі проміжні блоки, якщо не використовуються спеціальні блоки, які змінюють маршрут руху транзактів. При русі транзактів через модель команди опису і команди керування ігноруються. Поняття транзакта встановлює розробник моделі. Наприклад, при розробці моделі великого магазину в поняття транзакта можна закласти покупця, який переміщується між відділами, може стояти в черзі, здійснювати покупки, спілкуватись з іншими покупцями, вимагати книгу скарг тощо. При моделюванні виробничого процесу зручно під транзактами розуміти деталі.

Об'єктами обчислювальної категорії є змінні, функції, арифметичні вирази і PLUS-процедури.

Для збереження даних під час моделювання використовують об'єкти запам'ятовуючої категорії — комірки пам'яті та матриці комірок пам'яті. Вони можуть використовуватись також для обчислень.

До апаратної категорії відносять три типи об'єктів: одноканальні пристрої, багатоканальні пристрої і логічні перемикачі<sup>1</sup>. Одноканальні і багатоканальні пристрої (прилади) моделюють об'єкти, в яких відбувається обробка транзактів. Як правило, вона пов'язана з витратами часу. Одноканальний

---

<sup>1</sup>У довідковій системі GPSS World використовуються терміни: Facilities, Storages, Logicswitches. У деяких книгах багатоканальні пристрої називають пам'яттями, а логічні перемикачі — логічними ключами.

пристрій може бути зайнятий лише одним транзактом. Багатоканальні пристрої можуть обслуговувати кілька транзактив одночасно. Логічні перемикачі можуть перебувати в одному з двох станів — 0 (вимкнений) і 1 (увімкнений). Транзакти при русі по моделі можуть вмикати, вимикати чи інвертувати перемикачі.

До статистичної категорії належать черги і таблиці. Транзакти в процесі руху можуть затримуватись перед блоками, вхід в які в заданих умовах неможливий. У цьому випадку утворюються черги. Система GPSS дозволяє зібрати статистику про них. Для відображення даних у стандартній табличній формі про різні події під час моделювання використовуються таблиці.

### § 6.3. Системні числові атрибути

Всі об'єкти мови GPSS мають числові характеристики, які називають *системними числовими атрибутами*<sup>2</sup>. Вони виконують функцію імен. Системні числові атрибути бувають двох видів. Системний числовий атрибут першого виду складається з двох частин. Перша частина вказує на клас (тип) об'єкта — групове ім'я, а друга — ідентифікує конкретного члена групи (вузьке ім'я).

Є кілька десятків класів системних числових атрибутів першого виду. Усі вони мають вигляд однієї чи двох латинських букв. Наприклад, Q — клас черг, V — клас змінних, FN — клас функцій.

Друга частина системного числового атрибута (вузьке ім'я) може бути двох видів: 1) послідовність латинських букв, цифр і символів підкреслення «\_», яка починається з букви; 2) натуральне число. У деяких випадках вузьке ім'я може бути

---

<sup>2</sup>У деяких книгах їх називають стандартними числовими атрибутами. У довідковій системі GPSS World системний числовий атрибут позначають аббревіатурою SNA — System Numerical Attribute.

ім'ям лише першого виду, в інших — як першого, так і другого виду.

Системні числові атрибути другого виду є атомарними і не мають вузького імені. Їх є лише кілька: A1, AC1, C1, M1, PR, TG1, XN1, Z1. Наприклад, AC1 — абсолютний модельний час.

Великі і малі букви в системних числових атрибутах, як і скрізь у GPSS, не розрізняються. Конкретні класи системних числових атрибутів розглядаються у наступних параграфах, а їхній список наведено у § 7.3. У межах однієї моделі однакові вузькі імена можуть використовуватись для позначення об'єктів різного типу. Зарезервовані ключові слова не можна використовувати як імена. В одних випадках використовують лише вузькі імена, а в інших потрібно записувати (повні) системні числові атрибути. При порушенні правил запису виникатимуть повідомлення про помилки. Треба бути обережними також з назвами класів системних числових атрибутів. Якщо таку назву (наприклад, V) використати як вузьке ім'я, то виникнуть малозрозумілі повідомлення про помилки або модель працюватиме неправильно. Щоб гарантувати відмінність вузьких імен від класів системних числових атрибутів, рекомендується використовувати вузькі імена, що складаються не менше, ніж з трьох букв, або використовувати в іменах символ підкреслення.

Клас системного числового атрибута відокремлюється від вузького імені, що містить букви, знаком долара «\$». Число від класу системного числового атрибута нічим не відокремлюється. Приклади: Q\$otx, FN\$abcd1, P125, RN4.

Крім розглянутої прямої адресації системні числові атрибути підтримують непряму адресацію, про яку йтиметься в § 7.3.

#### § 6.4. Формат рядка

Кожен оператор програми (за винятком операторів у тілі PLUS-процедури) записується в окремому рядку, тобто в

рядку може бути тільки одна команда опису, один блок або одна команда керування. Рядок може бути також порожнім або містити лише коментар. Формат рядка є наступним:

номер мітка ключове\_слово операнди коментар

Всі елементи рядка відокремлюються один від одного одним чи кількома пропусками або символами табуляції. На початку рядка також може знаходитись довільна кількість пропусків чи знаків табуляції.

Номер є необов'язковим елементом рядка, ніде не використовується й ігнорується. Він може використовуватись хіба що для зручності перегляду першої таблиці статистики, яка відображає кількість транзактів, що пройшли через кожен блок моделі.

Мітка теж є необов'язковим елементом рядка. Як і вузьке ім'я першого виду, мітка складається з латинських букв, цифр і символів підкреслення. Зрозуміло, що вона не повинна збігатися з ключовими словами і класами системних числових атрибутів. Мітка, яка стоїть у блоці, використовується для (умовного чи безумовного) переходу транзактів у цей блок. При оголошенні змінної, функції, таблиці тощо мітка виконує роль вузького імені цієї змінної, функції, таблиці.

Ключове слово є головним елементом рядка і визначає його зміст. Ключовим словом може бути назва блока, команди опису або команди керування.

Кількість операндів і їхній формат залежить від ключового слова. Як правило, операнди записуються через кому без пропусків (до чи після неї). В окремих випадках перший операнд є умовним і відокремлюється від решти операндів пропуском, а не комою. Операндами можуть бути числа, системні числові атрибути, вузькі імена, назви міток, арифметичні і логічні вирази, деякі ключові слова. Ціла і дробова частини числа відокремлюються дробовою крапкою. Якщо операнд має бути цілим, а отримується дробове число, то береться його

ціла частина. Операнди можуть бути як обов'язковими, так і необов'язковими. Їхній порядок є строго фіксованим. Якщо котрийсь з проміжних необов'язкових операндів треба пропустити, а якийсь з наступних (можливо останній) операнд мусить бути, то всі коми до нього мають бути збережені. Кінцеві коми ігноруються. Замість ком операнди можна відокремлювати пропусками, при цьому один пропуск між операндами відповідає одній комі. Зокрема, запис операндів А, В рівносильний запису А,В або запису з двома пропусками між операндами А і В.

Коментар є необов'язковим елементом рядка і починається з символу «;». Усі символи після нього в рядку ігноруються. Інколи буває зручно «закоментувати» кілька рядків моделі, поклавши на початку цих рядків символи «;». Рядок, який починається з символу «\*», також розглядається як коментар.

Обсяг моделі при використанні вільної студентської версії середовища GPSS World має не перевищувати 180 блоків.

Створити рядок з блоком мови GPSS можна також командою меню **Edit►Insert GPSS Blocks**. У цьому випадку достатньо вибрати назву блока з запропонованих варіантів (рис. 6.4) і ввести в спеціальному вікні потрібні операнди, мітку та коментар.

Для кращого сприйняття програм на мові GPSS World скрізь у посібнику назви блоків, команд опису, команд керування, класів системних числових атрибутів та інших зарезервованих термінів наводяться великими буквами, всі імена і мітки — малими буквами, а імена стандартних функцій записуються з використанням великих і малих букв. Читач може не дотримуватись цього правила і набирати програми, наприклад, лише малими буквами. У статистичному звіті великими буквами відображаються не лише назви блоків, але й всі імена і мітки, використані в програмі.

ADOPT	ASSEMBLE	ALTER
ADVANCE	CLOSE	COUNT
ASSIGN	GATE	DISPLACE
BUFFER	JOIN	EXAMINE
DEPART	LINK	EXECUTE
ENTER	LOGIC	FAVAIL
GENERATE	LOOP	FUNAVAIL
LEAVE	MATCH	GATHER
MARK	OPEN	INDEX
MSAVEVALUE	PREEMPT	INTEGRATION
PLUS	PRIORITY	SAVAIL
QUEUE	READ	SCAN
RELEASE	REMOVE	SELECT
SAVEVALUE	RETURN	SUNAVAIL
SEIZE	SEEK	TABULATE
SPLIT	TEST	TRACE
TERMINATE	UNLINK	UNTRACE
TRANSFER	WRITE	

Рис. 6.4

## § 6.5. Арифметичні і логічні вирази

Замість більшості операндів у блоках мови GPSS World та в командах опису змінних можна використовувати арифметичні і логічні вирази. Вираз є розширенням стандарту мови GPSS і тому задається у круглих дужках. У командах опису змінних круглі дужки є необов'язковими. Вирази можуть містити числа, системні числові атрибути, змінні користувача, математичні функції і функції для генерації випадкових величин, поєднані знаками арифметичних і логічних операцій. Нагадаємо, що з допомогою системних числових атрибутів здійснюється доступ до змінних, функцій, комірок пам'яті та інших об'єктів.

У виразах можна використовувати наступні операції:

- ^ — піднесення до степеня;
- # — множення;
- / — ділення;
- \ — ціла частина від ділення;

- @ — остача від ділення;
- + — додавання;
- — віднімання;
- = або 'E' — дорівнює,  $A=B$  дає 1, якщо  $A$  дорівнює  $B$ , інакше — 0;
- < або 'L' — менше,  $A<B$  дає 1, якщо  $A$  менше  $B$ , інакше — 0;
- > або 'G' — більше,  $A>B$  дає 1, якщо  $A$  більше  $B$ , інакше — 0;
- <= або 'LE' — менше або дорівнює,  $A<=B$  дає 1, якщо  $A$  менше або дорівнює  $B$ , інакше — 0;
- >= або 'GE' — більше або дорівнює,  $A>=B$  дає 1, якщо  $A$  більше або дорівнює  $B$ , інакше — 0;
- 'NE' — не дорівнює,  $A'NE'B$  дає 1, якщо  $A$  не дорівнює  $B$ , інакше — 0;
- & або 'AND' — логічне «І»,  $A\&B$  дає 1, якщо  $A$  і  $B$  не дорівнюють нулю, інакше — 0;
- | або 'OR' — логічне «АБО»,  $A|B$  дає 1, якщо  $A$  або  $B$  не дорівнює нулю, інакше — 0.

Операції над елементами виразу виконуються в такій послідовності у порядку спадання пріоритетів:

- ^ — піднесення до степеня;
- #, /, \ — множення, ділення, ціла частина від ділення;
- @ — остача від ділення;
- , + — віднімання, додавання;
- <, >, <=, >= — порівняння;
- =, 'NE' — дорівнює, не дорівнює;
- & — логічне «І»;
- | — логічне «АБО».

Для зміни порядку виконання операцій використовують круглі дужки. У виразі біля знаків арифметичних і логічних операцій дозволяється ставити пропуски.

Приклади:  $((X4+V\$rrr)\#P2)$ ,  $(P1=1 \& Q\$chr1<3)$ .

## § 6.6. Блоки створення, знищення, затримки, зміни параметрів і пріоритету транзактів

Для створення і введення транзактів у модель використовують блок `GENERATE`. Його формат є наступним:

`GENERATE A,B,C,D,E.`

Обов'язковим є лише перший операнд  $A$  або четвертий операнд  $D$ . Якщо операнд  $A$  є числом, а операнд  $B$  — відсутній, то в модель вводиться детермінований потік транзактів (через кожні  $A$  одиниць модельного часу в модель вводиться один транзакт). Для моделювання випадкового потоку транзактів з рівномірним законом розподілу на проміжку  $[a, b]$  використовують операнди  $A$  і  $B$ . У цьому випадку операнд  $A$  задає середину проміжку  $\frac{a+b}{2}$ , а операнд  $B$  — половину довжини проміжку  $\frac{b-a}{2}$ . Якщо операндом  $A$  є функція, створена командою опису `FUNCTION` (див. § 6.13) чи одним з 35 розподілів випадкових величин (див. § 6.14), а операнд  $B$  — відсутній, то інтервали часу між створеними транзактами визначатимуться цією функцією. Якщо операнд  $A$  є функцією, а операнд  $B$  — числом або одним з розподілів випадкових величин, то проміжки часу між транзактами визначатимуться рівномірним розподілом на проміжку  $[A - B, A + B]$ .

Якщо операнд  $B$  є системним числовим атрибутом класу `FN` (тобто функцією, визначеною командою опису `FUNCTION`, див. § 6.13), то маємо особливий випадок, коли проміжки часу між транзактами визначатимуться за правилом  $A \cdot B$ .

Інші операнди блока `GENERATE` використовують зрідка. Операнд  $C$  задає час появи першого транзакта. Операнд  $D$  вказує кількість транзактів, які потрібно згенерувати. Ціле число на місці операнда  $E$  визначає пріоритет транзактів. Якщо операнд  $E$  не використовується, то створені транзакти мають пріоритет 0. Чим більшим є число, тим вищим є пріоритет.

Якщо проміжок часу між появою транзактів з якоїсь причини виявиться меншим від нуля, то з'явиться повідомлення про помилку. Транзакт не може зайти в блок **GENERATE**; при виникненні такої спроби буде згенеровано повідомлення про помилку.

Приклади:

**GENERATE 100** — через кожні 100 одиниць модельного часу в модель вводиться один транзакт;

**GENERATE 100,30** — інтервали часу між появами транзактів мають рівномірний розподіл на проміжку [70, 130];

**GENERATE FN\$func1** — інтервали часу між потраплянням транзактів у модель задаються значеннями функції **func1**;

**GENERATE ,,5,2** — на початку моделювання одночасно створюється п'ять транзактів з пріоритетом 2.

Процес моделювання запускається командою керування

**START A,**

де **A** — початкове значення лічильника завершень (натуральне число). Команду **START** можна дати з меню **Command** після трансляції моделі або навести її в кінці моделі. В останньому випадку процес моделювання автоматично запускається після трансляції моделі.

Вийшовши з блока **GENERATE** і пройшовши через модель, транзакт виводиться з моделі (знищується) блоком

**TERMINATE A,**

де **A** — ціле значення, яке віднімається від значення лічильника завершень. Коли значення лічильника завершень стає рівним нулю, моделювання припиняється. Якщо операнд **A** не використовується, то він вважається рівним нулю, тобто при знищенні транзакта відповідним блоком **TERMINATE** значення лічильника завершень не змінюється. Якщо використовується блок **TERMINATE 1**, то команда керування **START A** за-

пускає процес моделювання руху  $A$  транзактів через модель, наприклад:

```
GENERATE 40,10
```

```
...
```

```
TERMINATE 1
```

```
START 25
```

У деяких випадках необхідно здійснювати моделювання деякого процесу протягом певного часу. Для цього на початку або в кінці моделі додається пара блоків:

```
GENERATE A
```

```
TERMINATE 1
```

де  $A$  — час, протягом якого має здійснюватись моделювання. Крім того, всі інші блоки **TERMINATE** у моделі дають з нульовим операндом (**TERMINATE** 0), а для запуску моделювання застосовують команду **START** 1.

Для затримки транзактів з метою імітації їхнього обслуговування використовується блок

```
ADVANCE A,B.
```

Зміст операндів  $A$  і  $B$  тут той самий, що й у блоці **GENERATE**. Зокрема, якщо є лише операнд  $A$ , то він визначає час затримки транзакта. Якщо є операнди  $A$  і  $B$ , то час затримки є випадковим числом, рівномірно розподіленим на проміжку  $[A - B, A + B]$ .

Приклади:

**ADVANCE** 5,4 — затримка транзактів на 5,4 одиниці модельного часу;

**ADVANCE** 5,2 — час затримки транзактів є випадковою величиною, рівномірно розподіленою на проміжку  $[3, 7]$ ;

**ADVANCE FN\$func1** — час затримки транзактів визначається функцією **func1**.

Кожен транзакт характеризується своїм номером, параметрами, пріоритетом і часом перебування в моделі. Транзакти

автоматично нумеруються натуральними числами. У кожен момент часу обробляється тільки один транзакт. Такий транзакт називають *активним*. Номер активного транзакта міститься в системному числовому атрибуті `XN1`.

Параметри транзакта дозволяють зберігати і переміщувати по моделі інформацію, визначену розробником моделі. Блок `ASSIGN` змінює значення параметра транзакта, який зайшов у цей блок. Блок `ASSIGN` може працювати в трьох режимах: фіксації, збільшення і зменшення. Для цього використовуються три формати:

`ASSIGN A,B` присвоює параметру  $A$  значення  $B$  (режим фіксації);

`ASSIGN A+,B` збільшує значення параметра  $A$  на  $B$  одиниць (режим збільшення);

`ASSIGN A-,B` зменшує значення параметра  $A$  на  $B$  одиниць (режим зменшення).

Операндом  $A$  є вузьке ім'я параметра транзакта (ним може бути натуральне число або послідовність латинських букв, цифр і символів підкреслення, що починається з букви), а не системний числовий атрибут. Якщо параметру транзакта не було присвоєно значення, а він використовується в блоці `ASSIGN A+,B`, то вважається, що його початкове значення дорівнює нулю. Параметрам транзактів можна також надавати рядкових значень, їх задають у подвійних лапках.

Клас системного числового атрибута для параметрів транзактів позначається буквою `P`. Опитують параметри активного транзакта з допомогою системних числових атрибутів. Спроба опитування параметра транзакта, якому не присвоєно жодного значення, викликає повідомлення про помилку. Транзакт може мати довільну кількість параметрів. Різні транзакти можуть мати різні параметри, які можуть містити різні значення.

Приклади:

`ASSIGN 1,10` — параметру з ім'ям 1 (номер 1) транзакта,

який зайшов у блок ASSIGN, присвоюється значення 10;

ASSIGN  $abcd+, 2.1$  — параметр  $abcd$  транзакта, який зайшов у блок ASSIGN, збільшується на величину 2,1;

ASSIGN  $Q_$ , "Hello!" — параметру  $Q_$  транзакта, який зайшов у блок ASSIGN, присвоюється значення "Hello!";

$P1$  — звертання до параметра з ім'ям 1 (номер 1) активного транзакта;

$P\$abcd$  — звертання до параметра  $abcd$  активного транзакта;

ASSIGN  $2, (2\#P1+2.4)$  — параметру з номером 2 транзакта, який зайшов у блок ASSIGN, присвоюється результат обчислення виразу  $2\#P1+2.4$ .

Блок PRIORITY  $A$  змінює рівень пріоритету транзакта ( $A$  — новий пріоритет). Для визначення пріоритету активного транзакта використовується системний числовий атрибут PR. З врахуванням пріоритетів здійснюється моделювання роботи одноканальних і багатоканальних пристроїв.

### § 6.7. Основні блоки і команди опису, пов'язані з пристроями

Досить часто виникає потреба моделювати такі ситуації, коли одночасно може обслуговуватись не більше одного чи обмеженої кількості транзактів. Для імітації таких ситуацій використовують одноканальні і багатоканальні пристрої.

Блок SEIZE  $A$  моделює вхід в *одноканальний пристрій* з ім'ям  $A$ . Ім'ям може бути послідовність латинських букв, цифр і символів підкреслення, що починається з букви, або натуральне число. Якщо одноканальний пристрій  $A$  є вільним, то транзакт, який потрапляє в блок SEIZE  $A$ , займає цей пристрій. Якщо одноканальний пристрій  $A$  вже зайнятий іншим транзактом, то нові транзакти затримуються у попередньому блоці. Черга формується з врахуванням пріоритету транзактів. При однаковому пріоритеті нові транзакти стають у хвіст черги.

Блок **RELEASE A** моделює вихід з одноканального пристрою з ім'ям *A*. Транзакт, який входить у цей блок, звільняє одноканальний пристрій.

Блок **RETURN A** є синонімом до блока **RELEASE A**.

**Приклад.** У магазині є одна каса. Час обслуговування одного покупця в касі є рівномірно розподіленою випадковою величиною на проміжку  $[10, 60]$  секунд. Відомо, що покупці з'являються через кожні 0–80 секунд. З допомогою мови імітаційного моделювання GPSS здійснити моделювання обслуговування 100 покупців у касі, визначити коефіцієнт її завантаження і час обслуговування 100 покупців.

**Розв'язання.** За одиницю модельного часу в цій задачі зручно вибрати секунду. Роль покупців у моделі будуть виконувати транзакти. Інтервали часу між надходженням покупців вважаємо випадковими величинами з рівномірним розподілом на відрізку  $[0, 80]$ . Взагалі, у всіх задачах, де відомо лише проміжок зміни випадкової величини, будемо вважати, що ця випадкова величина розподілена рівномірно. Для моделювання обслуговування 100 покупців використовуємо команду **START** з операндом 100 і блок **TERMINATE** з операндом 1. Програма має вигляд:

```
GENERATE 40,40
SEIZE kasa
ADVANCE 35,25
RELEASE kasa
TERMINATE 1
START 100
```

За результатами моделювання отримуємо звіт (рис. 6.5), з якого робимо висновок про те, що коефіцієнт завантаження каси приблизно дорівнює 0,84 (стовпець **UTIL.** таблиці **FACILITY**), а загальний час обслуговування 100 покупців становить близько 70 хвилин (**END TIME**). Пояснення щодо того, як читати звіт, наводяться в § 6.12.

## GPSS World Simulation Report - Model1.1.1

Saturday, January 04, 2025 12:17:58

START TIME	END TIME	BLOCKS	FACILITIES	STORAGES
0.000	4183.558	5	1	0

NAME	VALUE
KASA	10000.000

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT	COUNT	RETRY
	1	GENERATE	100	0	0	0
	2	SEIZE	100	0	0	0
	3	ADVANCE	100	0	0	0
	4	RELEASE	100	0	0	0
	5	TERMINATE	100	0	0	0

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
KASA	100	0.841	35.167	1	0	0	0	0	0

FEC XN	PRI	BDT	ASSEM	CURRENT	NEXT	PARAMETER	VALUE
101	0	4223.466	101	0	1		

Рис. 6.5

Треба мати на увазі, що чимало характеристик у звіті залежить від випадкових чисел, а тому повторний запуск моделі командою **START 100** може дати децю інші числа. Визначити довжину черги до каси з використанням наведеної моделі неможливо. Для визначення довжини черги в моделі потрібно використовувати спеціальні блоки, які розглядаються в § 6.8. ■

Блок

PREEMPT *A,B,C,D,E*

також моделює вхід в *одноканальний пристрій* з ім'ям *A*. Обслуговування транзакта в такому пристрої може бути перерване іншим транзактом. Обов'язковим є лише операнд *A*. Якщо операнд *B* є відсутнім, то новий транзакт перериває обслуговування транзакта, який вже зайняв пристрій, при умові, що той транзакт сам не є «загарбником» (*режим переривання*). Робота «загарбника» не переривається. Якщо операндом

$B$  є системний числовий атрибут PR, то перебування транзакта в пристрої переривається лише тоді, коли новий транзакт має більш високий пріоритет (*режим пріоритетів*).

Якщо немає операндів  $C$ ,  $D$ ,  $E$ , то транзакт автоматично відновить обслуговування у пристрої після завершення обслуговування «загарбника». Час дообслуговування дорівнює часу, який залишався транзакту до завершення обслуговування в одноканальному пристрої.

Операнд  $C$  може містити мітку або номер блока, куди потрібно відправити транзакт, обслуговування якого було перерване. В операнді  $D$  може вказуватись ім'я або номер параметра транзакта, в якому зберігатиметься залишок часу, що залишався транзакту до завершення обслуговування у пристрої. Якщо операнд  $E$  містить ключове слово RE, то транзакт, обслуговування якого було перерване, не буде дообслуговуватись у пристрої. Якщо використовується операнд  $E$ , то одночасно має використовуватись операнд  $C$ . Якщо операнд  $C$  використовується без операнда  $E$ , то транзакт зберігає право на дообслуговування у пристрої. Тому цей транзакт не може бути знищений блоком TERMINATE перш, ніж транзакт увійде в блок RELEASE або RETURN для пристрою з ім'ям  $A$ .

Для моделювання виходу з одноканального пристрою, вхід в який моделювався блоком PREEMPT, має використовуватись блок RETURN  $A$  або RELEASE  $A$ .

Приклад моделювання роботи одноканального пристрою, обслуговування транзакта в якому може бути перерване транзактом з вищим пріоритетом:

```
PREEMPT pry1,PR  
ADVANCE 1  
RETURN pry1
```

Для використання *багатоканального пристрою* в моделі його спочатку потрібно описати командою опису

$N$  STORAGE  $A$ ,

де  $N$  — ім'я багатоканального пристрою,  $A$  — його ємність, тобто кількість каналів. Ім'ям може бути послідовність латинських букв, цифр і символів підкреслення, що починається з букви. Команду `STORAGE`, як і інші команди опису, зазвичай записують на початку програми.

Блок

`ENTER A,B`

моделює вхід у багатоканальний пристрій з ім'ям  $A$ , описаний командою `STORAGE`. Необов'язковий операнд  $B$  вказує кількість каналів, які займає транзакт при вході в багатоканальний пристрій. За замовчуванням транзакт претендує тільки на один канал. Якщо у багатоканальному пристрої немає достатньої кількості вільних каналів, то транзакти затримуються в попередньому блоці. Черга формується з урахуванням пріоритету транзактів. При однаковому пріоритеті нові транзакти стають у хвіст черги.

Блок

`LEAVE A,B`

моделює вихід із багатоканального пристрою з ім'ям  $A$ . Необов'язковий операнд  $B$  вказує кількість каналів, які звільняє транзакт при виході з багатоканального пристрою. За замовчуванням транзакт звільняє один канал.

Приклад моделювання роботи п'ятиканального пристрою:

```
bkr STORAGE 5
ENTER bkr
ADVANCE 2
LEAVE bkr
```

### § 6.8. Блоки і команди опису, пов'язані зі статистичною категорією

Частина результатів моделювання автоматично наводиться у статистичному звіті, який формується після завершення моделювання. Однак для збору статистики часу виконання

деякого процесу або інформації про черги доводиться використовувати спеціальні команди опису і блоки, які розглядаються у цьому параграфі.

Команда **TABLE** використовується для опису *таблиць*, які збиратимуть і відображатимуть статистичну інформацію про результати моделювання. Цю команду застосовують у наступному форматі:

$$N \text{ TABLE } A, B, C, D,$$

де  $N$  — ім'я таблиці,  $A$  — системний числовий атрибут, який визначає те, що буде відображатись у таблиці,  $B$  — верхня межа першого інтервалу таблиці,  $C$  — ширина всіх інтервалів, крім першого і останнього,  $D$  — кількість інтервалів (натуральне число). Якщо число  $D$  виявиться надто великим, то в таблиці буде відображено інформацію лише для тих інтервалів, яким реально відповідатимуть транзакти. Якщо число  $D$  буде надто малим, то в таблиці буде наведено детальну інформацію лише про частину транзактів. У таблиці відзначається, скільки транзактів (та який їхній відсоток) за деяким критерієм відповідає кожному з інтервалів  $(-\infty, B]$ ,  $(B, B + C]$ ,  $(B + C, B + 2C]$ ,  $(B + 2C, B + 3C]$ ,  $\dots$ ,  $(B + C(D - 3), B + C(D - 2)]$ ,  $(B + C(D - 2), \infty)$ , а також виводиться середнє значення і середньоквадратичне відхилення.

Найчастіше створюють таблиці, в яких відображається інформація про розподіл часу руху транзактів через модель або її частину. Системний числовий атрибут  $M1$  забезпечує доступ до часу перебування транзакта в моделі. Його використовують замість операнда  $A$  у команді **TABLE** для збору інформації в таблиці про розподіл часу руху транзактів у моделі від моменту їхнього створення до потрапляння у блок **TABULATE**  $A$ , де  $A$  — ім'я таблиці.

Системний числовий атрибут  $MP\alpha$  використовують для збору інформації в таблиці про розподіл часу руху транзактів через частину моделі від їхнього входження у блок **MARK**  $\alpha$  до потрапляння у блок **TABULATE**  $A$ , де  $A$  — ім'я таблиці.

Можна також використовувати на місці операнда  $A$  у команді TABLE системний числовий атрибут з будь-яким параметром транзакта. У цьому випадку таблиця відобразатиме частотний розподіл транзактів різного типу (відповідно до вибраного параметра).

**Приклад 1.** Доповнимо програму для прикладу з § 6.7 (с. 160) кількома операторами для збору статистики розподілу часу перебування покупців біля каси і часу обслуговування покупців касиром.

**Розв'язання.** Створимо дві таблиці — `tab1` і `tab2`. У таблиці `tab1` будемо збирати статистику розподілу часу, необхідного покупцям для перебування в черзі до каси і здійснення оплати, а в таблиці `tab2` — лише для здійснення оплати. Програма може бути, наприклад, такою:

```
tab1 TABLE M1,10,10,100
tab2 TABLE MP1,10,10,100
GENERATE 40,40
SEIZE kasa
MARK 1
ADVANCE 35,25
RELEASE kasa
TABULATE tab1
TABULATE tab2
TERMINATE 1
START 100
```

Отримані таблиці наведені на рис. 6.6. З них, зокрема, видно, що середній час перебування покупця біля каси становить 70 секунд (що на 10 секунд більше, ніж максимальний можливий час обслуговування покупця касиром), а один з покупців на перебування біля каси витратив понад три хвилини (останній рядок першої таблиці). Детальний опис таблиць наводиться в § 6.12 (с. 184). ■

TABLE	MEAN	STD.DEV.	RANGE		RETRY FREQUENCY	CUM.%
TAB1	70.000	38.643			0	
			10.000 -	20.000	8	8.00
			20.000 -	30.000	9	17.00
			30.000 -	40.000	10	27.00
			40.000 -	50.000	9	36.00
			50.000 -	60.000	10	46.00
			60.000 -	70.000	8	54.00
			70.000 -	80.000	6	60.00
			80.000 -	90.000	9	69.00
			90.000 -	100.000	10	79.00
			100.000 -	110.000	4	83.00
			110.000 -	120.000	3	86.00
			120.000 -	130.000	8	94.00
			130.000 -	140.000	2	96.00
			140.000 -	150.000	0	96.00
			150.000 -	160.000	3	99.00
			160.000 -	170.000	0	99.00
			170.000 -	180.000	0	99.00
			180.000 -	190.000	1	100.00
TAB2	35.167	15.230			0	
			10.000 -	20.000	24	24.00
			20.000 -	30.000	15	39.00
			30.000 -	40.000	17	56.00
			40.000 -	50.000	25	81.00
			50.000 -	60.000	19	100.00

Рис. 6.6

У § 6.7 говорилось про те, що при використанні одноканальних чи багатоканальних пристроїв черга створюється автоматично. Однак зі звіту для прикладу § 6.7 (с. 160) взагалі не видно черги, а з прикладу 1 § 6.8 видно, що у звіті є лише опосередкована інформація про чергу. Для того щоб побачити у звіті характеристики черги, треба використовувати спеціальні блоки.

Блок

QUEUE  $A, B$

моделює вхід у чергу з ім'ям  $A$ , при входженні транзакта в цей блок її довжина збільшується на  $B$  одиниць. Операнд  $B$  можна не вказувати, в такому разі довжина черги збільшується на одиницю.

Блок

DEPART  $A, B$

моделює вихід з черги з ім'ям  $A$ , при входженні транзакта в цей блок її довжина зменшується на  $B$  одиниць. При відсутності операнда  $B$  довжина черги зменшується на одиницю. Транзакт може знаходитись у кількох чергах одночасно. Для правильної роботи блок QUEUE зазвичай розміщують перед блоком SEIZE, PREEMPT або ENTER, а блок DEPART — після. При використанні блоків QUEUE і DEPART у звіті виводиться інформація про середню і максимальну довжину черги, середній час перебування в черзі, кількість транзактів, які очікували в черзі тощо. Для того щоб отримати детальну статистику розподілу часу, проведеного транзактами в черзі, використовують команду опису таблиці для черги

$N$  QTABLE  $A, B, C, D$ ,

де  $N$  — ім'я таблиці для черги,  $A$  — ім'я черги,  $B$  — верхня межа першого інтервалу таблиці,  $C$  — ширина всіх інтервалів, крім першого і останнього,  $D$  — кількість інтервалів. Операнди  $B$ ,  $C$  і  $D$  задаються аналогічно відповідним операндам таблиці TABLE.

Для доступу до поточної довжини черги в моделі використовується системний числовий атрибут класу Q.

**Приклад 2.** Доповнимо програму для прикладу з § 6.7 (с. 160), крім операторів для збору статистики розподілу часу перебування покупців біля каси і часу обслуговування покупців касиром (приклад 1 § 6.8, с. 165), також операторами для визначення характеристик і детальної статистики черги.

**Розв'язання.** Отримаємо модель:

```
tab1 TABLE M1,10,10,100
tab2 TABLE MP1,10,10,100
tab3 QTABLE cherga,10,10,100
GENERATE 40,40
```

```

QUEUE cherga
SEIZE kasa
DEPART cherga
MARK 1
ADVANCE 35,25
RELEASE kasa
TABULATE tab1
TABULATE tab2
TERMINATE 1
START 100

```

Отримані додаткові таблиці наведені на рис. 6.7. З них, зокрема, видно, що максимальна довжина черги становила 4, середня довжина черги наближено дорівнює 0,8, середній час перебування в черзі — 35 секунд, але один покупець перебував у черзі понад дві хвилини. ■

QUEUE	MAX	CONT.	ENTRY	ENTRY(0)	AVE.CONT.	AVE.TIME	AVE.(-0)	RETRY
CHERGA	4	0	100	31	0.833	34.833	50.483	0

TABLE	MEAN	STD.DEV.	RANGE	RETRY	FREQUENCY	CUM.%
TAB3	34.833	35.299		0		
			- -	10.000	37	37.00
			10.000 -	20.000	9	46.00
			20.000 -	30.000	5	51.00
			30.000 -	40.000	9	60.00
			40.000 -	50.000	5	65.00
			50.000 -	60.000	12	77.00
			60.000 -	70.000	7	84.00
			70.000 -	80.000	3	87.00
			80.000 -	90.000	5	92.00
			90.000 -	100.000	1	93.00
			100.000 -	110.000	1	94.00
			110.000 -	120.000	5	99.00
			120.000 -	130.000	1	100.00

Рис. 6.7

## § 6.9. Використання змінних, логічних перемикачів і комірок пам'яті

Для опису дійсної змінної з ім'ям  $N$  використовується команда

$N$  FVARIABLE  $A$ .

Операнд  $A$  — це вираз для обчислення змінної, який буде утворюватися за правилами побудови арифметичних виразів. Оточувати його круглими дужками тут необов'язково. У GPSS World команда опису VARIABLE є синонімом до команди FVARIABLE. У попередніх версіях вона була призначена для здійснення обчислень у множині цілих чисел. Для звертання до змінної використовується клас системних числових атрибутів  $V$ .

Команда опису

$N$  BVARIABLE  $A$

визначає логічну (булеву) змінну з ім'ям  $N$ . Така змінна може набувати лише значень 0 і 1. Операнд  $A$  — логічний вираз. Для звертання до логічної змінної використовується клас системних числових атрибутів  $BV$ .

Змінні дозволяють виконувати проміжні обчислення за певними, наперед заданими формулами. Під час моделювання ці формули не можуть змінюватись. Значення змінної обчислюється при звертанні до неї активного транзакта. У стандартному звіті значення змінних не виводяться.

Приклади:

aaaa FVARIABLE 10#X\$qwerty+P1 — створюється дійсна змінна з ім'ям aaaa;

bbbb BVARIABLE P1>3 & X1<3 — створюється логічна змінна з ім'ям bbbb;

V\$aaaa — звертання до дійсної змінної з ім'ям aaaa;

BV\$bbbb — звертання до логічної змінної з ім'ям bbbb.

Для збереження даних змінні не використовують, а здійснення обчислень в операндах блоків з допомогою арифмети-

чних виразів дозволяє взагалі відмовитись від використання змінних. Логічні перемикачі і комірки пам'яті, на відміну від змінних, дозволяють зберігати інформацію, яка може бути довільним чином змінена в процесі моделювання.

*Логічні перемикачі* можна трактувати як змінні, які можуть набувати лише двох значень — 0 (вимкнено) і 1 (увімкнено). Блок

LOGIC X A

використовують для перемикання логічного перемикача. Операнд  $X$  є умовним оператором і може набувати наступних значень: **S** — увімкнути перемикач, **R** — вимкнути перемикач, **I** — інвертувати значення перемикача. Операнд  $A$  має бути ім'ям логічного перемикача. Значення логічного перемикача змінюється при входженні будь-якого транзакта в блок LOGIC.

Для визначення стану, в якому перебуває логічний перемикач, використовують системний числовий атрибут перемикача. Клас цього системного числового атрибута позначають буквами **LS**.

Логічний перемикач автоматично створюється при використанні блока LOGIC або при опитуванні системного числового атрибута перемикача, причому спочатку перемикач є неувімкненим (дорівнює нулю). Якщо логічний перемикач на початку моделювання має бути увімкненим, його описують командою

INITIAL A,

де  $A$  — системний числовий атрибут перемикача.

Приклади:

INITIAL LS\$abracadabra — початкове значення логічного перемикача з ім'ям abracadabra є рівним 1;

LOGIC I 15 — при входженні транзакта в блок значення логічного перемикача з ім'ям 15 змінюється на протилежне.

*Комірки пам'яті* призначені для збереження числових або рядкових величин, їх можна трактувати як змінні і використо-

увати в обчисленнях. Блок **SAVEVALUE** змінює вміст комірки пам'яті. Його можна використовувати у трьох форматах:

**SAVEVALUE A,B** присвоює комірці пам'яті *A* значення *B* (режим фіксації);

**SAVEVALUE A+,B** збільшує значення комірки пам'яті *A* на *B* одиниць (режим збільшення);

**SAVEVALUE A-,B** зменшує значення комірки пам'яті *A* на *B* одиниць (режим зменшення).

Операндом *A* є вузьке ім'я комірки пам'яті (ним може бути натуральне число або послідовність латинських букв, цифр і символів підкреслення, яка починається з букви). Значення комірки пам'яті змінюється при входженні довільного транзакта в блок **SAVEVALUE**. Рядкові значення задають у подвійних лапках.

Клас системного числового атрибута для комірки пам'яті позначається буквою **X**. Опитують комірки пам'яті, як і інші об'єкти, з допомогою системних числових атрибутів. Комірка пам'яті автоматично створюється при використанні блока **SAVEVALUE** або при опитуванні системного числового атрибута комірки пам'яті, причому в останньому випадку її початкове значення дорівнює нулю. Якщо початкове значення комірки пам'яті має не дорівнювати нулю, то її описують командою

**INITIAL A,B,**

де *A* — системний числовий атрибут комірки пам'яті, *B* — початкове значення комірки пам'яті.

Приклади:

**SAVEVALUE 2,20** — комірці пам'яті з ім'ям 2 (номер 2) присвоюється значення 20;

**SAVEVALUE aaa+,2.3** — значення комірки пам'яті *aaa* збільшується на величину 2,3;

**SAVEVALUE xy1,"Yes"** — комірці пам'яті *xy1* присвоюється значення "Yes";

INITIAL X\$bbb,1 — початкове значення комірки пам'яті bbb дорівнює 1;

X2 — звертання до комірки пам'яті з ім'ям 2 (номер 2);

X\$aaa — звертання до комірки пам'яті aaa;

SAVEVALUE kin, (V\$хуз#P10-X12) — комірці пам'яті kin присвоюється результат обчислення арифметичного виразу (змінна хуз множиться на параметр номер 10 транзакта, який потрапив у цей блок, і від отриманого результату віднімається вміст комірки пам'яті номер 12).

Стан логічних перемикачів і вміст комірок пам'яті після завершення моделювання виводиться у статистичному звіті (§ 6.12, с. 185).

### § 6.10. Блок TRANSFER для зміни маршруту руху транзактів

Для зміни маршруту руху транзактів використовуються чотири види блоків. Спочатку розглянемо блок TRANSFER. Транзакт, що потрапляє в блок

TRANSFER A,B,C,D,

переходить у деякий інший блок відповідно до режиму переходу, визначеного операндом A. Розглянемо основні сім режимів.

1. Режим безумовного переходу. Якщо операнд A пропущено, то блок працює в режимі безумовного переходу. У цьому режимі транзакт переходить у блок, визначений міткою або номером, що міститься в операнді B. Інші операнди не використовуються. Блоки нумеруються послідовними натуральними числами, порожні рядки, рядки з командами або коментарями не нумеруються. Приклад:

TRANSFER ,mit — безумовний перехід на мітку mit.

2. Режим статистичної передачі. Якщо операнд A є числом, то блок працює в режимі статистичної передачі. Якщо

$A \in (0, 1)$ , то транзакт з імовірністю  $A$  переходить у блок, визначений міткою або номером, що міститься в операнді  $C$ , а з імовірністю  $1 - A$  — у блок, визначений міткою або номером в операнді  $B$ . Якщо ціла частина операнда  $A$  є натуральним числом, меншим від 1000, то вона інтерпретується як кількість тисячних часток імовірності передачі. Якщо  $A \leq 0$ , то транзакт переходить у блок, визначений операндом  $B$ . Якщо  $A \geq 1000$ , то транзакт переходить у блок, визначений операндом  $C$ . Якщо операнд  $B$  пропущено, то транзакт з імовірністю  $1 - A$  переходить у наступний блок. Якщо операнд  $C$  пропущено, то транзакт з імовірністю  $A$  переходить у наступний блок. Операнд  $D$  не використовується. Приклади:

TRANSFER 0.3,mit1,mit2 — транзакт з імовірністю 0,3 переходить у блок з міткою mit2, а з імовірністю 0,7 — у блок з міткою mit1;

TRANSFER 10,12 — транзакт з імовірністю 0,01 переходить у наступний блок, а з імовірністю 0,99 — у дванадцятий блок.

3. Режим BOTH. Якщо операнд  $A$  є ключовим словом BOTH, то блок працює в режимі BOTH. У цьому режимі транзакт намагається зайти в блок, визначений міткою або номером з операнда  $B$ . Якщо зайти у цей блок не вдається, бо він є входом у зайнятий одноканальний чи багатоканальний пристрій, то транзакт намагається зайти в блок, визначений міткою або номером з операнда  $C$ . Якщо не вдається зайти в жоден з двох блоків, то транзакт затримується у блоці TRANSFER до тих пір, доки він не зможе зайти в один з них. Якщо один з операндів  $B$  або  $C$  є відсутнім, то вважається, що він вказує на наступний блок. Операнд  $D$  не використовується. Приклади:

TRANSFER BOTH,rob1,rob2 — транзакт намагається зайти в один з двох блоків, визначених мітками rob1 і rob2;

TRANSFER BOTH,rob2 — транзакт намагається зайти у наступний блок або у блок, визначений міткою rob2.

4. Режим ALL. Якщо операнд  $A$  є ключовим словом ALL, то блок працює в режимі ALL. У цьому режимі перевіряється

блок, вказаний в операнді *B*. Якщо цей блок не може прийняти транзакт, то послідовно перевіряються всі блоки до тих пір, поки не буде досягнутий блок, визначений операндом *C*, або один з перевірених блоків не прийме транзакт. Розташування кожного блока, що послідовно перевіряється, обчислюється шляхом додавання до розташування останнього перевіреного блока операнда *D*. Блоки з більшим розташуванням, ніж блок, визначений операндом *C*, не розглядаються. Якщо операнд *D* не використовується, то послідовно перевіряються всі блоки між блоками, визначеними операндами *B* і *C*. Якщо заданий операнд *C*, то перевіряється лише один блок. Транзакт скеровується в перший блок, який його приймає. Якщо жоден блок не приймає транзакт, то він затримується в блоці TRANSFER до тих пір, доки він не зможе зайти в один із цих блоків. Приклад:

TRANSFER ALL,b1\_b,b1\_e,2 — послідовно перевіряються всі блоки через один, починаючи з блока з міткою b1\_b і закінчуючи блоком з міткою b1\_e або попереднім до нього (в залежності від розташування цих блоків); якщо жоден блок не приймає транзакт, то він залишається у блоці TRANSFER до тих пір, доки його не зможе прийняти один з них.

5. Режим PICK. Якщо операнд *A* є ключовим словом PICK, то блок працює в режимі PICK. У цьому режимі місце призначення вибирається випадковим чином з-поміж блоків від блока, визначеного операндом *B*, до блока, визначеного операндом *C*. Операнд *D* не використовується.

Приклад рівноймовірного переходу на одну з п'яти міток mit1, mit2, mit3, mit4, mit5:

```
TRANSFER PICK,b1_b,b1_e
b1_b TRANSFER ,mit1
TRANSFER ,mit2
TRANSFER ,mit3
TRANSFER ,mit4
b1_e TRANSFER ,mit5
```

Рівноймовірний перехід на одну з кількох міток можна реалізувати акуратним використанням меншої кількості блоків TRANSFER в режимі статистичної передачі. Тому попередній приклад можна подати коротше:

```
TRANSFER 0.2,mit1
TRANSFER 0.25,mit2
TRANSFER 0.333,mit3
TRANSFER 0.5,mit4,mit5
```

6. Підпрограмний режим. Якщо операнд  $A$  є ключовим словом SBR, то блок працює у підпрограмному режимі. У цьому режимі транзакт переходить у блок, на який вказує операнд  $B$ , а в параметрі з ім'ям, визначеним операндом  $C$ , зберігається номер поточного блока TRANSFER. Операнд  $D$  не використовується. Підпрограмний режим дозволяє викликати підпрограму. Для виходу з підпрограми слід використовувати блок TRANSFER у безумовному або параметричному режимі. Приклад:

```
TRANSFER SBR,rob,vern
...
rob ...
...
TRANSFER ,(P$vern+1)
```

7. Параметричний режим. Якщо операндом  $A$  є буква P, то блок працює в параметричному режимі. При входженні транзакта у блок TRANSFER до параметра транзакта з ім'ям, визначеним операндом  $B$ , додається вміст операнда  $C$ , якщо він заданий. Ціла частина отриманого результату визначає номер блока для переходу транзакта. Операнд  $D$  не використовується. Приклад:

TRANSFER P,param,1 — транзакт переходить у блок, розташований зразу після блока, на який вказує параметр з ім'ям param цього транзакта.

### § 6.11. Блоки TEST, GATE і LOOP для зміни маршруту руху транзактів

Для переходу транзактів при виконанні певних умов використовуються блоки TEST і GATE. Блок

TEST  $X$   $A, B, C$

порівнює операнди  $A$  і  $B$  відповідно до умовного операнда  $X$ . Операндами  $A$  і  $B$  можуть бути системні числові атрибути, числа, арифметичні або логічні вирази. Умовний операнд  $X$  може набувати наступних значень:

$E$  — значення операнда  $A$  дорівнює значенню операнда  $B$ ;

$NE$  — значення операнда  $A$  не дорівнює значенню операнда  $B$ ;

$L$  — значення операнда  $A$  менше від значення операнда  $B$ ;

$LE$  — значення операнда  $A$  менше або дорівнює значенню операнда  $B$ ;

$G$  — значення операнда  $A$  більше від значення операнда  $B$ ;

$GE$  — значення операнда  $A$  більше або дорівнює значенню операнда  $B$ .

Якщо умова виконується, то транзакт, який потрапив у блок TEST, переходить у наступний блок. Необов'язковий операнд  $C$  може містити мітку або номер блока для альтернативного переходу. Якщо умова не виконується і заданий операнд  $C$ , то транзакт, який потрапив у блок TEST, переходить у блок, визначений операндом  $C$ . Якщо умова не виконується і відсутній операнд  $C$ , то транзакт, який потрапив у блок TEST, затримується у цьому блоці до тих пір, доки умова не буде виконана.

Для перевірки виконання складної умови у блоці TEST в одному з операндів  $A$  або  $B$  розміщують логічний вираз і перевіряють його на рівність 0 чи 1.

Приклади:

TEST E P1,2,mit — якщо значення параметра номер 1 транзакта, який потрапив у цей блок, дорівнює числу 2, то

транзакт переходить у наступний блок, інакше — у блок з міткою *mit*;

TEST L  $X\$cin, (2\#(X1+P\$cin))$  — якщо вміст комірки пам'яті з ім'ям *cin* є меншим, ніж подвоєна сума вмісту комірки пам'яті номер 1 і параметра транзакта з ім'ям *cin*, то транзакт, який потрапив у цей блок, переходить у наступний блок, інакше він затримується у блоці TEST до тих пір, доки умова не буде виконана (умова виконається тільки тоді, коли якийсь інший транзакт відповідним чином змінить вміст хоч однієї з двох використаних в умові комірок пам'яті);

TEST E  $(X3 \geq 5.1 \ \& \ Q\$ch_ < 10), 1$  — якщо вміст комірки пам'яті номер 3 є більшим або рівним числу 5,1 і поточна довжина черги з ім'ям *ch\_* є меншою від 10, то транзакт, який потрапив у цей блок, переходить у наступний блок, інакше він затримується у блоці TEST до тих пір, доки умова не буде виконана.

Блок

### GATE X A,B

змінює маршрут руху транзактів залежно від стану об'єкту, заданого операндом *A*. Необов'язковий операнд *B* може містити мітку або номер блока для альтернативного переходу. Умовний операнд *X* може набувати наступних значень для перевірки відповідних умов:

U — одноканальний пристрій з ім'ям *A* є зайнятим;

NU — одноканальний пристрій з ім'ям *A* є вільним;

I — обслуговування транзакта в одноканальному пристрої з ім'ям *A* щойно було перерване;

NI — обслуговування транзакта в одноканальному пристрої з ім'ям *A* не було щойно перерване;

FV — одноканальний пристрій з ім'ям *A* є доступним;

FNV — одноканальний пристрій з ім'ям *A* є недоступним;

SF — всі канали багатоканального пристрою з ім'ям *A* є зайнятими;

SNF — принаймні один канал багатоканального пристрою з ім'ям  $A$  є вільним;

SE — всі канали багатоканального пристрою з ім'ям  $A$  є вільними;

SNE — принаймні один канал багатоканального пристрою з ім'ям  $A$  є зайнятим;

SV — багатоканальний пристрій з ім'ям  $A$  є доступним;

SNV — багатоканальний пристрій з ім'ям  $A$  є недоступним;

LS — логічний перемикач з ім'ям  $A$  увімкнений;

LR — логічний перемикач з ім'ям  $A$  вимкнений;

M — блок MATCH, на який вказує операнд  $A$ , містить транзакт, що очікує на синхронізацію з активним транзактом;

NM — блок MATCH, на який вказує операнд  $A$ , не містить транзакта, що очікує на синхронізацію з активним транзактом.

Про доступність і недоступність одноканальних і багатоканальних пристроїв ітиметься в § 7.6. При синхронізацію руху транзактів з допомогою блоків MATCH ітиметься в § 7.1.

Якщо умова виконується, то транзакт, який потрапив у блок GATE, переходить у наступний блок. Якщо умова не виконується і заданий операнд  $B$ , то транзакт, який потрапив у блок GATE, переходить у блок, визначений операндом  $B$ . Якщо умова не виконується і відсутній операнд  $B$ , то транзакт, який потрапив у блок GATE, затримується у цьому блоці до тих пір, доки умова не буде виконана.

Приклади:

GATE U pry1,mit — якщо одноканальний пристрій з ім'ям pry1 — зайнятий, то транзакт, який зайшов у цей блок, переходить у наступний блок, інакше — у блок з міткою mit;

GATE LS 11 — якщо логічний перемикач з номером 11 є ввімкненим, то транзакт, який зайшов у цей блок, переходить у наступний блок, інакше він затримується у блоці GATE до тих пір, доки умова не буде виконана (умова виконається тільки тоді, коли якийсь інший транзакт змінить стан перемикача з

номером 11).

Блок

LOOP *A,B*

призначений для організації руху транзактів у циклі. Операнд *A* має містити ім'я (номер) параметра транзакта, який визначатиме кількість проходів транзакта по циклу. Операнд *B* — мітка (номер) блока, в який переходить транзакт, якщо його параметр *A* не дорівнює нулю. При входженні транзакта в блок LOOP значення його параметра *A* зменшується на одиницю. Якщо воно є більшим від нуля, то транзакт переходить у блок, на який вказує операнд *B*. Якщо воно дорівнює нулю, то транзакт переходить у наступний блок. Якщо значення параметра *A* виявиться меншим від нуля, то виникне помилка.

Приклад створення циклу, який буде виконуватись п'ять разів (якщо в тілі циклу не змінюватиметься параметр з номером 1, який використовується як лічильник ітерацій):

```
ASSIGN 1,5  
mit ...  
...  
LOOP 1,mit
```

## § 6.12. Елементи стандартного звіту

Після завершення моделювання система автоматично створює звіт. Звіт містить загальну інформацію про результати моделювання та статистичну інформацію про конкретні об'єкти, які використовуються в моделі. Наведемо приклади і пояснення для основних елементів звіту. Роз'яснення про інші елементи звіту наводяться у §§ 7.7, 7.8.

**Рядки заголовка.** Перший рядок звіту завжди містить ім'я файлу моделі, номер трансляції і номер запуску моделі на виконання, відокремлені крапками. Другий рядок звіту містить дату і час завершення моделювання, наприклад:

GPSS World Simulation Report - Model4.32.1

Monday, January 13, 2025 16:25:48

**Загальна інформація про результати роботи моделі**

подається у вигляді таблиці, наприклад:

START TIME	END TIME	BLOCKS	FACILITIES	STORAGES
0.000	4349.501	24	2	1

**START TIME** — початковий час, тобто абсолютний модельний час у момент початку моделювання;

**END TIME** — кінцевий час, тобто значення абсолютного модельного часу в момент, коли лічильник завершень набуває значення 0;

**BLOCKS** — кількість блоків у моделі;

**FACILITIES** — кількість одноканальних пристроїв, використаних під час моделювання;

**STORAGES** — кількість багатоканальних пристроїв, використаних під час моделювання.

Тривалість моделювання (в одиницях модельного часу) дорівнює різниці кінцевого і початкового часу.

**Інформація про імена** також подається у вигляді таблиці, наприклад:

NAME	VALUE
AVBF	4.670
ВКР	10002.000
CHERGA	10003.000
ККК	10004.000
MIT1	16.000
MIT2	12.000
PMM	UNSPECIFIED
S_	10005.000
TAB1	10000.000
VAR	10006.000

**NAME** — перелік заданих користувачем імен, які містяться у програмі;

VALUE — числові значення, присвоєні іменам. Для міток числовими значеннями є номери відповідних блоків, для змінних користувача<sup>1</sup> — значення цих змінних у момент завершення моделювання. В інших випадках система надає іменам унікальні номери, починаючи з 10000. Якщо ім'я не використовувалось під час моделювання, то воно містить значення UNSPECIFIED.

**Інформація про блоки.** Наступна у звіті таблиця містить досить важливу інформацію про використані в моделі блоки, наприклад:

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT	COUNT	RETRY
	1	GENERATE	111		0	0
	2	QUEUE	111		3	0
	3	SEIZE	108		0	0
	4	DEPART	108		0	0
	5	ADVANCE	108		1	0
	6	RELEASE	107		0	0
	7	TABULATE	107		0	0
	8	GATE	107		0	0
	9	ASSIGN	67		0	0
	10	ASSIGN	67		0	0
	11	PRIORITY	67		0	0
MIT2	12	ENTER	68		0	0
	13	ADVANCE	68		1	0
	14	LEAVE	67		0	0
	15	SAVEVALUE	67		0	0
MIT1	16	LOGIC	107		0	0
	17	TRANSFER	107		4	0
	18	PREEMPT	102		1	0
	19	ADVANCE	101		1	0
	20	RELEASE	100		0	0
	21	TERMINATE	100		0	0
	22	SEIZE	0		0	0
	23	ADVANCE	0		0	0
	24	RELEASE	0		0	0

LABEL — мітка, тобто алфавітно-цифрове ім'я блока (якщо воно задане);

LOC — числовий номер позиції блока в моделі;

<sup>1</sup>Про змінні користувача йтиметься в § 7.8.

**BLOCK TYPE** — тип блока GPSS;

**ENTRY COUNT** — кількість транзактів, які увійшли до блока від початку роботи програми чи після останнього виконання команди **RESET** або **CLEAR**<sup>1</sup>;

**CURRENT COUNT** — кількість транзактів, які перебувають у блоці в момент завершення моделювання;

**RETRY** — кількість транзактів, які чекають виконання спеціальної умови, яка залежить від стану відповідного блока.

Наявність решти таблиць у звіті залежить від конкретної моделі.

### Інформація про одноканальні пристрої

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
22	108	0.876	35.285	1	108	0	0	0	3
S_	102	0.916	39.063	1	104	4	1	0	0

**FACILITY** — ім'я або номер одноканального пристрою;

**ENTRIES** — кількість транзактів, які займали пристрій (у тому числі з перериванням) від початку моделювання чи після останнього виконання команди **RESET** або **CLEAR**;

**UTIL.** — коефіцієнт використання пристрою, тобто частка часу моделювання, протягом якого пристрій був зайнятим;

**AVE. TIME** — середній час займання пристрою одним транзактом від початку моделювання чи після останнього виконання команди **RESET** або **CLEAR**;

**AVAIL.** — стан пристрою в кінці моделювання (1, якщо пристрій доступний; 0 — якщо недоступний);

**OWNER** — номер транзакта, який займає пристрій у кінці моделювання (0 — пристрій не зайнятий);

**PEND** — кількість транзактів, які очікують на обслуговування з перериванням обслуговування інших транзактів (тільки для блока **PREEMPT**, що працює в режимі переривань);

<sup>1</sup>Про команди **RESET** і **CLEAR** йтиметься в § 7.4.

**INTER** — кількість транзактів, обслуговування яких було перерване на даний момент;

**RETRY** — кількість транзактів, які чекають виконання спеціальної умови, яка залежить від стану цього пристрою;

**DELAY** — кількість транзактів, які на момент завершення моделювання очікують на входження в одноканальний пристрій (крім блока **PREEMPT**, що працює в режимі переривань).

### Інформація про багатоканальні пристрої

STORAGE	CAP.	REM.	MIN.	MAX.	ENTRIES	AVL.	AVE.C.	UTIL.	RETRY	DELAY
ВКР	6	5	0	3	68	1	0.643	0.107	0	0

**STORAGE** — ім'я або номер багатоканального пристрою;

**CAP.** — ємність багатоканального пристрою (кількість каналів), задана командою опису **STORAGE**;

**REM.** — кількість вільних каналів багатоканального пристрою на момент завершення моделювання;

**MIN.** — мінімальна кількість каналів багатоканального пристрою, які були використані під час моделювання;

**MAX.** — максимальна кількість каналів багатоканального пристрою, які були використані під час моделювання;

**ENTRIES** — кількість входжень транзактів до пристрою під час моделювання;

**AVL.** — стан пристрою в кінці моделювання (1, якщо пристрій доступний; 0 — якщо недоступний);

**AVE.C.** — середня кількість зайнятих каналів за час моделювання;

**UTIL.** — коефіцієнт використання пристрою;

**RETRY** — кількість транзактів, які чекають виконання спеціальної умови, яка залежить від стану цього пристрою;

**DELAY** — кількість транзактів, які на момент завершення моделювання очікують на входження в багатоканальний пристрій.

## Інформація про черги

QUEUE	MAX	CONT.	ENTRY	ENTRY(0)	AVE.CONT.	AVE.TIME	AVE.(-0)	RETRY
CHERGA	7	3	111	28	1.112	43.573	58.272	0

QUEUE — ім'я або номер черги;

MAX — максимальна довжина черги під час моделювання;

CONT. — поточна довжина черги в кінці процесу моделювання;

ENTRY — загальна кількість входжень транзактів у чергу протягом моделювання;

ENTRY(0) — загальна кількість входжень транзактів у чергу з нульовим часом очікування;

AVE.CONT. — середня довжина черги протягом моделювання;

AVE.TIME — середній час перебування одного транзакта в черзі з урахуванням усіх входжень до черги;

AVE.(-0) — середній час перебування одного транзакта в черзі без урахування «нульових» входжень до черги;

RETRY — кількість транзактів, які чекають виконання спеціальної умови, яка залежить від стану черги.

## Інформація про таблиці

TABLE	MEAN	STD.DEV.	RANGE	RETRY	FREQUENCY	CUM.%
TAB1	77.774	54.449		0		
			- -	20.000	8	7.48
			20.000 -	40.000	19	25.23
			40.000 -	60.000	26	49.53
			60.000 -	80.000	16	64.49
			80.000 -	100.000	9	72.90
			100.000 -	120.000	8	80.37
			120.000 -	140.000	5	85.05
			140.000 -	160.000	7	91.59
			160.000 -	180.000	1	92.52
			180.000 -	200.000	4	96.26
			200.000 -	220.000	1	97.20
			220.000 -	240.000	2	99.07
			240.000 -	260.000	1	100.00

TABLE — ім'я або номер таблиці або таблиці для черги;

MEAN — середнє арифметичне табличних значень;

STD.DEV. — середньоквадратичне відхилення табличних значень;

RANGE — нижня і верхня межі частотного класу (символ підкреслення використовується для позначення  $-\infty$  або  $+\infty$ );

RETRY — кількість транзактів, які чекають виконання спеціальної умови, яка залежить від стану цієї таблиці;

FREQUENCY — частота, тобто кількість табличних значень, які потрапили в зазначені межі частотного класу;

CUM.% — накопичена частота, подана в процентах до загальної кількості.

### Інформація про логічні перемикачі

LOGICSWITCH	VALUE	RETRY
KKK	1	0

LOGICSWITCH — ім'я або номер логічного перемикача;

VALUE — значення перемикача в кінці моделювання (1 — увімкнений; 0 — вимкнений);

RETRY — кількість транзактів, які чекають виконання спеціальної умови, яка залежить від стану цього логічного перемикача.

### Інформація про комірки пам'яті

SAVEVALUE	RETRY	VALUE
1	0	82.410

SAVEVALUE — ім'я або номер комірки пам'яті;

VALUE — значення збереженої величини на момент завершення моделювання;

RETRY — кількість транзактів, які чекають виконання спеціальної умови, яка залежить від стану даної комірки пам'яті.

### Інформація про списки поточних і майбутніх подій.

Списки поточних і майбутніх подій містять інформацію про транзакти на момент завершення моделювання, наприклад:

CEC XN	PRI	M1	ASSEM	CURRENT	NEXT	PARAMETER	VALUE
104	0	4039.240	104	18	19		
FEC XN	PRI	BDT	ASSEM	CURRENT	NEXT	PARAMETER	VALUE
112	0	4360.199	112	0	1		
108	0	4387.011	108	5	6		
107	1	4390.664	107	13	14	2	1.400
						VAR	5.000

CEC XN — номер транзакта, який міститься у списку поточних подій;

PRI — пріоритет транзакта;

M1 — час створення транзакта (для копії транзакта — час створення транзакта, який його породив, § 7.1);

ASSEM — номер сім'ї транзакта (§ 7.1);

CURRENT — номер блока, в якому перебуває транзакт у момент завершення моделювання;

NEXT — номер наступного блока, в який мав би перейти транзакт при продовженні моделювання;

PARAMETER — ім'я або номер параметра транзакта;

VALUE — значення параметра транзакта;

FEC XN — номер транзакта, який міститься у списку майбутніх подій;

BDT — значення абсолютного модельного часу, коли транзакт має залишити список майбутніх подій.

В окремих випадках (при застосуванні спеціальних блоків) у звіті можуть бути присутніми додаткові таблиці. Таблиця для списків користувача описується в § 7.7.

### § 6.13. Функції і генератори випадкових чисел

У мові GPSS World можна одночасно використовувати кілька різних генераторів випадкових чисел. Для доступу до них застосовують системні числові атрибути класу RN, наприклад: RN8, RN\$vyp\_ch. Генератори випадкових чисел створюють цілі числа, рівномірно розподілені на проміжку [0, 999].

Початкове значення нумерованого генератора випадкових чисел визначається його номером.

У блоках **GENERATE**, **ADVANCE** і **TRANSFER** за замовчуванням для генерації випадкових чисел використовується системний числовий атрибут **RN1**, але його можна замінити з допомогою вкладки **Random Numbers** вікна, яке відкривається командою меню **Edit►Settings**.

Функція в мові GPSS задається таблицею. Правила пошуку значення функції за таблицею визначає команда опису **FUNCTION**. Вона записується принаймні у два рядки, причому другий і наступні рядки опису функції мають містити список даних функції (тобто саму таблицю):

$$N \text{ FUNCTION } A, Kn \\ X_1, Y_1 / X_2, Y_2 / \dots / X_n, Y_n$$

Мітка  $N$  визначає ім'я функції. Операндом  $A$  є аргумент функції, ним може бути будь-який системний числовий атрибут або арифметичний вираз у дужках. Часто буває потрібним задати деяку випадкову залежність. Тоді на місці операнда  $A$  розміщують один із генераторів випадкових чисел, наприклад, **RN15**. У цьому випадку аргументом функції буде рівномірно розподілене випадкове число з проміжку  $[0, 1)$ .

Операнд  $Kn$  складається з однієї букви (**C**, **D**, **E**, **L** або **M**), яка вказує тип функції, і натурального числа  $n$  — кількості пар даних у таблиці значень функції. Буква і число пропуском не відокремлюються.

Пари  $X_i, Y_i$  — координати точок у списку даних функції. Вони відокремлюються символами «/». При необхідності частину пар переносять на наступні рядки, причому рядки повинні починатись з  $X_i$ . У кінці рядків символи «/» можна не ставити. Кількість пар у списку даних функції має збігатись із числом  $n$ , вказаним у операнді  $Kn$ . Рядки списку даних функції не нумеруються і не можуть містити мітки і коментарі. Величини  $X_1, X_2, \dots, X_n$  мають утворювати неспадну

послідовність чисел.

Для звертання до функції використовують системний числовий атрибут з класом FN, наприклад, FN\$asd.

Кожен тип функції має свої особливості використання. Буква C в операнді  $Kn$  означає, що функція є неперервною, тобто точки, задані парами  $X_i, Y_i$ , з'єднуються відрізками й утворюється ламана (рис. 6.8). Значення такої функції GPSS знаходить з допомогою лінійної інтерполяції. Для неперервної функції координати точок  $X_i, Y_i$  мають бути числами, причому перші координати  $X_i$  повинні бути невід'ємними. Якщо аргумент функції виходить за межі проміжку  $[X_1, X_n]$ , то функція вертає найближче крайнє значення проміжку. Якщо аргументом функції є генератор випадкових чисел, то повинні виконуватись рівності  $X_1 = 0, X_n = 1$ .

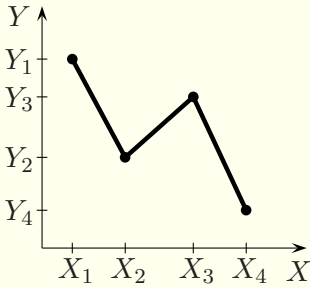


Рис. 6.8.

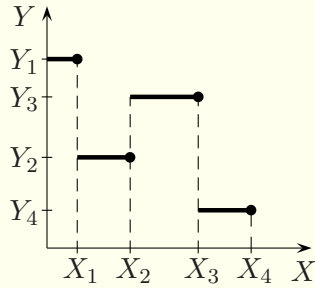


Рис. 6.9.

Приклад неперервної функції, аргументом якої є змінна zmo:

```
namef FUNCTION V$zmo, C5
0, 1/1, 2/2, 1.5/4, 1/6, -0.5
```

Буква D в операнді  $Kn$  означає, що функція є дискретною. Її графік є кусково-сталим (рис. 6.9). Дискретна функція дорівнює  $Y_i$  на цілому півінтервалі  $(X_{i-1}, X_i]$  для  $i > 1$  і дорівнює  $Y_1$  на проміжку  $(-\infty, X_1]$ . Для дискретної функції координати точок  $X_i, Y_i$  мають бути числами. При обчисленні

функції значення  $X_i$  у списку даних функції перебираються від найменшого до найбільшого. Для найменшого значення  $X_i$ , більшого або рівного значенню аргументу, вертається відповідне йому значення  $Y_i$ . Якщо таке значення  $X_i$  відсутнє, то отримуємо значення  $Y_n$ . Якщо аргументом функції є генератор випадкових чисел, то повинні виконуватись умови  $X_1 \geq 0$ ,  $X_n = 1$ .

Приклад дискретної функції з випадковим аргументом:

```
namef1 FUNCTION RN10,D5
0.1,1/0.2,2/0.6,1.5/0.8,1/1,2
```

Буква E в операнді  $Kn$  команди FUNCTION означає, що функція є дискретною, але координати  $Y_i$  у списку даних функції можуть бути системними числовими атрибутами або виразами у круглих дужках. Дробові числа тут також треба брати в круглі дужки. Функція типу E працює аналогічно функції типу D. Приклад:

```
namef2 FUNCTION RN10,E5
0.1,1/0.2,V$zmo/0.6,(1.5)/0.8,(P1+0.5)/1,2
```

Буква L в операнді  $Kn$  означає, що функція є списковою. Значення аргументу використовується для визначення позиції значення, яке буде вертатись. Значення  $X_i$  повинні бути послідовними натуральними числами, починаючи з 1. Значення  $Y_i$  повинні бути числами. Приклад:

```
namef3 FUNCTION X1,L6
1,1/2,2.1/3,0.7/4,3/5,2/6,-1.1
```

Буква M в операнді  $Kn$  означає, що функція є списковою, але координати  $Y_i$  у списку даних функції можуть бути системними числовими атрибутами або виразами у круглих дужках. Дробові числа тут також треба брати в круглі дужки. Функція типу M працює аналогічно функції типу L. Приклад:

```
namef4 FUNCTION X1,M6
1,1/2,(2.1)/3,X$zmk/4,(P2+0.5)/5,2/6,-1.1
```

**Приклад.** Час появи замовлень у двоканальній системі масового обслуговування визначається функцією з випадковим аргументом, графік якої зображено на рис. 6.10. Час обслуговування задається функцією з випадковим аргументом, графік якої наведено на рис. 6.11. Виконати моделювання обслуговування 300 замовлень у такій системі масового обслуговування.

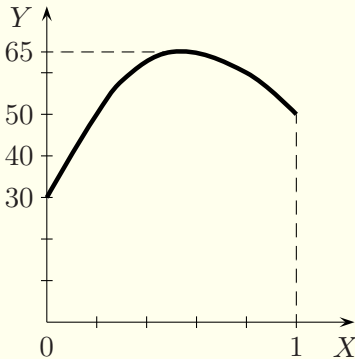


Рис. 6.10.

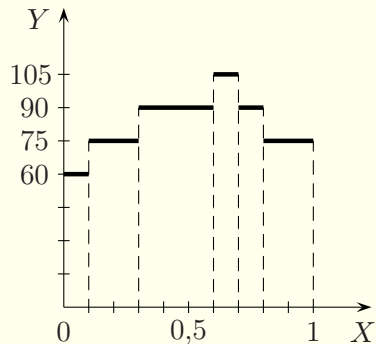


Рис. 6.11.

**Розв'язання.** Неперервну функцію, графік якої подано на рис. 6.10, назвемо `gen` і закодуємо шістьма точками. Дискретну функцію з рис. 6.11 назвемо `adv` і також закодуємо шістьма точками. Для моделювання системи масового обслуговування використаємо двоканальний пристрій `smo`. Для визначення характеристик черги і тривалості перебування замовлень в моделі застосуємо відповідні блоки і команду для опису таблиці. Наведемо текст моделі:

```
smo STORAGE 2
tab TABLE M1,10,10,100
gen FUNCTION RN10,C6
0,30/0.2,50/0.3,58/0.5,65/0.8,60/1,50
adv FUNCTION RN11,D6
0.1,60/0.3,75/0.6,90/0.7,95/0.8,90/1,75
```

```

GENERATE FN$gen
QUEUE cher
ENTER smo
DEPART cher
ADVANCE FN$adv
LEAVE smo
TABULATE tab
TERMINATE 1
START 300

```

На рис. 6.12 наведено частину звіту. З неї, зокрема, видно, що більшу частину часу черги не було, лише окремі замовлення перебували в черзі довжиною одне замовлення, коефіцієнт завантаження системи масового обслуговування наближено дорівнює 0,72, середній час перебування замовлення в моделі становить 82 одиниці модельного часу. Інші висновки зі звіту читачам пропонується зробити самостійно. ■

QUEUE	MAX	CONT.	ENTRY	ENTRY(0)	AVE.CONT.	AVE.TIME	AVE.(-0)	RETRY
CHER	1	0	301	289	0.005	0.255	6.397	0

STORAGE	CAP.	REM.	MIN.	MAX.	ENTRIES	AVL.	AVE.C.	UTIL.	RETRY	DELAY
SMO	2	1	0	2	301	1	1.447	0.723	0	0

TABLE	MEAN	STD.DEV.	RANGE		RETRY	FREQUENCY	CUM.%
TAB	82.106	10.530			0		
			50.000	-	60.000	26	8.67
			60.000	-	70.000	0	8.67
			70.000	-	80.000	121	49.00
			80.000	-	90.000	109	85.33
			90.000	-	100.000	42	99.33
			100.000	-	110.000	2	100.00

Рис. 6.12

Застосування функцій для моделювання емпірично розподілених випадкових величин розглядається в §§ 9.2–9.4.

### § 6.14. Математичні функції і функції для генерації випадкових величин

В арифметичних виразах мова GPSS World дозволяє використовувати наступні математичні функції:

Функція	Опис
ABS( $X$ )	модуль виразу $X$
SQR( $X$ )	квадратний корінь виразу $X$
SIN( $X$ )	синус виразу $X$ (в радіанах)
COS( $X$ )	косинус виразу $X$ (в радіанах)
TAN( $X$ )	тангенс виразу $X$ (в радіанах)
ATN( $X$ )	арктангенс виразу $X$ (в радіанах)
EXP( $X$ )	експонента виразу $X$
LOG( $X$ )	натуральний логарифм виразу $X$
INT( $X$ )	відкидання дробової частини виразу $X$

Приклад:

ADVANCE (2#SQR(ABS(P1))) — час затримки транзактів у блоці є подвоєним коренем квадратним від модуля параметра номер 1 транзакта, який зайшов у цей блок.

GPSS World містить 24 спеціальні функції для генерації випадкових величин з різними законами розподілу. Їх теж можна використовувати в арифметичних виразах. У всіх цих функціях першим аргументом має бути номер генератора випадкових чисел — довільне натуральне число, яке визначає послідовність випадкових чисел. Основні з функцій для генерації випадкових величин в алфавітному порядку наведені у наступній таблиці. З рештою функцій можна ознайомитись у довідковій системі GPSS World або в книзі [2].

Функція	Опис
$\text{Beta}(R, a, b, \alpha_1, \alpha_2)$	бета-розподіл на проміжку $[a, b]$ з параметрами $\alpha_1, \alpha_2$
$\text{Binomial}(R, n, p)$	біноміальний розподіл для серії $n$ незалежних випробувань з імовірністю успіху в одному випробуванні $p$
$\text{DUniform}(R, a, b)$	дискретний (цілочисловий) рівномірний розподіл на проміжку $[a, b]$
$\text{Exponential}(R, a, \bar{x})$	експоненціальний розподіл з середнім значенням $\bar{x}$ і зміщенням $a$
$\text{Gamma}(R, a, \beta, \alpha)$	гамма-розподіл з формою $\alpha$ , стисненням $\beta$ і зміщенням $a$
$\text{Geometric}(R, p)$	геометричний розподіл з імовірністю $p$ (розподіл моделює кількість невдалих спроб, див. § 8.5)
$\text{Lognormal}(R, a, \mu, \sigma)$	логнормальний розподіл з середнім значенням $\mu$ і середньоквадратичним відхиленням $\sigma$ відповідного нормального закону розподілу, зміщений на $a$ одиниць вправо
$\text{Normal}(R, \mu, \sigma)$	нормальний розподіл з середнім значенням $\mu$ і середньоквадратичним відхиленням $\sigma$
$\text{Poisson}(R, \lambda)$	розподіл Пуассона з середнім значенням $\lambda$
$\text{Triangular}(R, a, b, c)$	трикутний розподіл на проміжку $[a, b]$ з найбільш імовірним значенням $c$
$\text{Uniform}(R, a, b)$	рівномірний розподіл на проміжку $[a, b]$
$\text{Weibull}(R, a, \beta, \alpha)$	розподіл Вейбулла з формою $\alpha$ , стисненням $\beta$ і зміщенням $a$

Про функції щільності розподілів неперервних випадкових величин з попередньої таблиці та їхні графіки можна прочитати у §§ 8.6, 9.6. Зміщені розподіли розглядаються в § 9.14. Для незміщених розподілів замість параметра  $a$  у відповідних функціях використовують число 0. Вибору розподілів випадкових величин присвячено розділ 9.

Приклади:

**GENERATE** (*Exponential*(8,0,1/2.1)) — в модель вводиться стаціонарний пуассонівський потік транзактів з інтенсивністю 2,1;

**ADVANCE** (*Lognormal*(9,0,1,0.2)) — час затримки транзактів є випадковою величиною з логнормальним законом розподілу з середнім значенням 1 і середньоквадратичним відхиленням 0,2 відповідного нормального закону розподілу.

Також є функції для роботи з рядками і файлами.

### § 6.15. Приклади побудови моделей

**Приклад 1.** Телефонна система має три паралельні лінії зв'язку. Нові зовнішні виклики утворюють стаціонарний пуассонівський потік з середнім значенням 70 секунд між викликами. Якщо всі лінії зайняті, то абонент набирає номер ще раз через  $5 \pm 1$  хвилини. Тривалість розмови становить  $3 \pm 1$  хвилини. Потрібно скласти модель на мові GPSS World, яка б дозволила отримати: а) кількість часу, необхідного для здійснення 500 розмов; б) коефіцієнт завантаження телефонної системи; в) табулювання розподілу часу, який потрібен абоненту для встановлення зв'язку і здійснення розмови; г) середню і максимальну кількості абонентів, які чекають звільнення лінії; д) середню і максимальну кількості викликів, потрібні для здійснення абонентом телефонної розмови.

**Розв'язання.** Оскільки в постановці задачі для вимірювання часу використовуються як хвилини, так і секунди, потрібно весь час у хвилинах перевести в секунди або навпаки. Нехай одиницею модельного часу є секунда. У цій задачі зручно

абонентів імітувати транзактами. Для формування стаціонарного пуассонівського потоку вхідних викликів використаємо функцію **Exponential** в операнді блока **GENERATE**. Для моделювання роботи телефонної системи з трьома лініями зв'язку використаємо багатоканальний пристрій **tel** з трьома каналами. Оскільки у випадку, коли зайняті всі лінії зв'язку, абонент отримує відмову і змушений чекати  $5 \pm 1$  хвилин для здійснення повторної спроби виклику, то для перевірки зайнятості телефонної системи скористаємось блоком **GATE**, додатковим блоком затримки **ADVANCE** і блоком безумовного переходу **TRANSFER**. Таким чином, для здійснення моделювання 500 розмов і отримання відповіді на пункти а) і б) достатньо скласти таку модель:

```
tel STORAGE 3
GENERATE (Exponential(8,0,70))
again GATE SNF tel,wait
ENTER tel
ADVANCE 180,60
LEAVE tel
TERMINATE 1
wait ADVANCE 300,60
TRANSFER ,again
START 500
```

Для побудови статистики, згаданої у пункті в), створимо таблицю з ім'ям **trans** з допомогою команди опису **TABLE** і блока **TABULATE**. Часові інтервали в таблиці візьмемо рівними 60 секундам. Для відповіді на пункт г) зручно нетрадиційним способом скористатись чергою, для цього блок **ADVANCE** після мітки **wait** оточимо блоками **QUEUE** і **DEPART**.

Нарешті, для визначення середньої і максимальної кількостей спроб, потрібних для отримання зв'язку абонентом, у параметрі **вук** кожного транзакта будемо накопичувати кількість здійснених абонентом викликів. Для цього після мітки **again** розмістимо блок **ASSIGN**, який працюватиме в режимі

збільшення. Середню і максимальну кількості спроб виклику найлегше знайти з таблиці `calls`, побудованої за параметром `vyk`. Для цього скористаємось командою опису `TABLE`, першим операндом якої вкажемо системний числовий атрибут `P$vyk`, і блоком `TABULATE`. З таблиці `calls` насправді можна отримати ще більше статистичної інформації. Нижче наводиться остаточна модель:

```
tel STORAGE 3
trans TABLE M1,60,60,200
calls TABLE P$vyk,1,1,100
GENERATE (Exponential(8,0,70))
again ASSIGN vyk+,1
GATE SNF tel,wait
ENTER tel
ADVANCE 180,60
LEAVE tel
TABULATE trans
TABULATE calls
TERMINATE 1

wait QUEUE chr
ADVANCE 300,60
DEPART chr
TRANSFER ,again
START 500
```

У книзі повністю не наводиться досить об'ємний статистичний звіт (читач може отримати його самостійно), а наводяться лише його окремі частини і даються відповіді на поставлені питання. Таблиця з загальною інформацією у звіті

START TIME	END TIME	BLOCKS	FACILITIES	STORAGES
0.000	38746.003	13	0	1

дає кількість часу, необхідного для здійснення 500 розмов, — 38746 секунд або понад 10 годин. З таблиці

STORAGE	CAP.	REM.	MIN.	MAX.	ENTRIES	AVL.	AVE.C.	UTIL.	RETRY	DELAY
TEL	3	1	0	3	502	1	2.367	0.789	0	0

видно, що коефіцієнт завантаження телефонної системи приблизно становить 0,8. При зміні генераторів випадкових чисел будуть отримані трохи інші, але подібні результати.

Табулювання розподілу часу, який потрібен абоненту для встановлення зв'язку і здійснення розмови, видно з таблиці TRANS (наведена лише частково):

TABLE	MEAN	STD.DEV.	RANGE		RETRY	FREQUENCY	CUM. %
TRANS	685.330	777.028			0		
			120.000	-	180.000		
			180.000	-	240.000	91	18.20
			240.000	-	300.000	125	43.20
			300.000	-	360.000	0	43.20
			360.000	-	420.000	13	45.80
			420.000	-	480.000	42	54.20
			480.000	-	540.000	37	61.60
			540.000	-	600.000	13	64.20
			600.000	-	660.000	2	64.60
			660.000	-	720.000	12	67.00
			720.000	-	780.000	21	71.20
			780.000	-	840.000	22	75.60
			840.000	-	900.000	7	77.00
			900.000	-	960.000	5	78.00
			...		...	...	...
			6360.000	-	6420.000	0	99.80
			6420.000	-	6480.000	0	99.80
			6480.000	-	6540.000	1	100.00

Зокрема, виявляється, що 216 абонентів (понад 43 %) витратило на здійснення телефонної розмови до чотирьох хвилин (вони змогли додзвонитись з першої спроби). У 15 хвилин змогло вкластись 77 % абонентів, проте решті абонентів довелося чекати довше, зокрема, одному абоненту довелося чекати близько 1 години 48 хвилин. Середній час становить близько 11 хвилин.

Максимальну і середню кількості абонентів, які чекають звільнення лінії, знайдемо з таблиці для черги CHR:

QUEUE	MAX	CONT.	ENTRY	ENTRY(0)	AVE.CONT.	AVE.TIME	AVE.(-0)	RETRY
CHR	21	10	879	0	6.715	296.016	296.016	0

Максимальна кількість абонентів — 21, середня кількість абонентів — 6,7. Середній час очікування абонентів з цієї таблиці визначити неможливо. З таблиці

TABLE CALLS	MEAN 2.686	STD.DEV. 2.597	RANGE	RETRY FREQUENCY 0	CUM.%
			- -	1.000	216 43.20
			1.000 -	2.000	105 64.20
			2.000 -	3.000	67 77.60
			3.000 -	4.000	36 84.80
			4.000 -	5.000	25 89.80
			5.000 -	6.000	15 92.80
			6.000 -	7.000	12 95.20
			7.000 -	8.000	4 96.00
			8.000 -	9.000	3 96.60
			9.000 -	10.000	6 97.80
			10.000 -	11.000	3 98.40
			11.000 -	12.000	3 99.00
			12.000 -	13.000	2 99.40
			13.000 -	14.000	0 99.40
			14.000 -	15.000	0 99.40
			15.000 -	16.000	0 99.40
			16.000 -	17.000	1 99.60
			17.000 -	18.000	0 99.60
			18.000 -	19.000	0 99.60
			19.000 -	20.000	1 99.80
			20.000 -	21.000	0 99.80
			21.000 -	22.000	1 100.00

робимо висновок про те, що для одного абонента середня кількість викликів є трохи меншою від трьох, а максимальна — дорівнює 22. Крім того, з останньої таблиці можна побачити, наприклад, що 105 абонентів зробили по два виклики, а понад три чверті абонентів здійснили не більше трьох викликів. Читачам пропонується самостійно модифікувати модель так, щоб одиницею модельного часу була хвилина, а не секунда. ■

На практиці малоімовірно, що будь-який абонент буде близько двадцяти разів підряд телефонувати на зайняті телефонні лінії. У наступному прикладі розглядається модифікація попереднього прикладу.

**Приклад 2.** З допомогою GPSS World здійснити моделювання роботи телефонної системи з попереднього прикладу

(с. 194), якщо для абонента максимальна кількість невдалих викликів становить від трьох до шести. Визначити ймовірність того, що абонент не зможе додзвонитись.

**Розв'язання.** Будемо вважати, що максимальна кількість викликів має рівномірний розподіл серед цілих чисел на проміжку [3, 6]. Збережемо її у параметрі транзакта з ім'ям `maxv`. Для цього використаємо додатковий блок `ASSIGN` з викликом функції `DUniform` після блока `GENERATE` у моделі зі с. 196. Після мітки `wait` додамо перевірку виконання умови `P$vyk < P$maxv` (блок `TEST`). Для визначення кількості абонентів, які не змогли додзвонитись, використаємо блок `SAVEVALUE` в режимі збільшення. Для вилучення цих транзактів з моделі використаємо додатковий блок `TERMINATE` з нульовим операндом. Отримаємо модель:

```
tel STORAGE 3
trans TABLE M1,60,60,200
calls TABLE P$vyk,1,1,100
GENERATE (Exponential(8,0,70))
ASSIGN maxv,(DUniform(7,3,6))
again ASSIGN vyk+,1
GATE SNF tel,wait
ENTER tel
ADVANCE 180,60
LEAVE tel
TABULATE trans
TABULATE calls
TERMINATE 1

wait TEST 1 P$vyk,P$maxv,vidm
QUEUE chr
ADVANCE 300,60
DEPART chr
TRANSFER ,again

vidm SAVEVALUE vidmov+,1
```

```
TERMINATE 0
START 500
```

Після моделювання в комірці пам'яті `vidmov` міститься кількість абонентів, які не змогли додзвонитись. Імовірність того, що абонент не зможе додзвонитись, знаходимо за формулою

$$P_{\text{відм}} = \frac{\text{vidmov}}{500 + \text{vidmov}} = \frac{36}{536} = \frac{9}{134} \approx 0,07.$$

Насправді можна було б не додавати в модель блок `SAVEVALUE`, а кількість відмов знайти з таблиці блоків по кількості входів транзактів в останній блок `TERMINATE`. Обчислення ймовірності відмови можна було включити в модель з допомогою додаткового блока `SAVEVALUE` (зробіть самостійно!). Зі статистичного звіту можна отримати всі ті самі характеристики, що й у прикладі 1 зі с. 194 (зробіть це самостійно і порівняйте їх з аналогічними характеристиками з прикладу 1). ■

**Приклад 3.** Основна виробнича функція рекламного агентства повного циклу послуг «Фіалка» — виробництво реклами трьох видів — телереклама, друкована реклама і радіореклама. Виробничий процес в агентстві «Фіалка» організований наступним чином. Реклама трьох видів, що надходить від замовників з інтервалом, рівномірно розподіленим на проміжку [1, 100] хвилин, обробляється групою менеджерів по рекламі протягом 2–5 хвилин. Останні передають фінансові документи про оплату замовником прийнятої реклами в бухгалтерію, а самі рекламні замовлення у виробничий відділ. На це теж потрібно 2–5 хвилин. Виробничий відділ упродовж 1–3 хвилин скеровує замовлення залежно від їхнього типу в телестудію, видавництво або радіостудію рекламного агентства і здійснює контроль за їхнім виконанням. Імовірності того, що реклама виявиться телерекламою, друкованою рекламою та радіорекламою дорівнюють відповідно 0,25, 0,35 і 0,4. Тривалості виконання замовлень телестудією, видавництвом та ра-

діостудією складають відповідно 50–100, 30–90 та 15–30 хвилин, причому студії і видавництво не можуть виконувати більше одного замовлення одночасно. Готова реклама надходить від студій і видавництва у виробничий відділ і спрямовується ним у засоби масової інформації протягом 5–10 хвилин.

З допомогою системи імітаційного моделювання GPSS World здійснити моделювання роботи рекламного агентства впродовж восьмигодинного робочого дня. Проаналізувати кількість виконаних замовлень з різних видів реклами і порівняти її з заданими ймовірностями, визначити витрачений час на виконання замовлень, кількість часу, потрібного на виконання одного замовлення в середньому, відмінності між тривалостями часу виконання замовлень, наявність черг, їхні довжини, середній час перебування в чергах, кількість замовлень, виконаних без черги.

**Розв’язання.** Одиницею модельного часу є хвилина. Замовлення зручно імітувати транзактами. Телестудію, радіостудію і видавництво моделюємо з допомогою одноканальних пристроїв. Для визначення черг перед цими пристроями використовуємо блоки QUEUE і DEPART. Вважаємо, що група менеджерів по рекламі і виробничий відділ можуть виконувати кілька замовлень одночасно (про інше не сказано), тому їхню роботу можна моделювати лише з допомогою блоків ADVANCE. Вибір типу реклами зручно здійснювати з допомогою двох блоків TRANSFER, які працюють у режимі статистичної передачі. Для аналізу часу перебування замовлень у моделі використовується таблиця, описана командою TABLE, інформацію в таблицю збирає блок TABULATE. Для моделювання роботи рекламного агентства протягом восьми годин створюємо таймер із блоків GENERATE і TERMINATE, інші блоки TERMINATE використовуємо з нульовим операндом, а команду START — з одиничним операндом. Отримуємо таку модель:

```
tab TABLE M1,10,10,100  
GENERATE 480
```

```
TERMINATE 1
GENERATE 50.5,49.5
ADVANCE 3.5,1.5
ADVANCE 3.5,1.5
ADVANCE 2,1
TRANSFER 0.4,,radio
TRANSFER 0.5833,tv,druk
radio QUEUE ch_radio
SEIZE pryl_radio
DEPART ch_radio
ADVANCE 22.5,7.5
RELEASE pryl_radio
TRANSFER,zmi
tv QUEUE ch_tv
SEIZE pryl_tv
DEPART ch_tv
ADVANCE 75,25
RELEASE pryl_tv
TRANSFER,zmi
druk QUEUE ch_druk
SEIZE pryl_druk
DEPART ch_druk
ADVANCE 60,30
RELEASE pryl_druk
TRANSFER,zmi
zmi ADVANCE 7.5,2.5
TABULATE tab
TERMINATE 0
START 1
```

Читачам пропонується самостійно виконати моделювання і проаналізувати звіт. ■

**Приклад 4.** Виробництво деталей певного типу складається з тривалого процесу складання, що завершується коро-

тким періодом випалу в печі. Оскільки утримання печі обходиться досить дорого, кілька складальників використовують ту саму піч. У печі одночасно можна випалювати лише одну деталь. Якщо піч зайнята, складальник разом зі своєю заготовкою деталі чекає в черзі звільнення печі. Складальник не може розпочати процес складання нової деталі, доки не витягне готову деталь із печі. Після отримання з печі готової деталі складальник починає складати наступну деталь. Час складання деталі є рівномірно розподіленою випадковою величиною на проміжку [40, 50] хвилин. Час випалу деталі є рівномірно розподіленою випадковою величиною на проміжку [5, 10] хвилин. Зарплата складальника становить 10 грошових одиниць за годину, вартість печі — 600 грошових одиниць за добу незалежно від міри використання, вартість матеріалу — 7 грошових одиниць за деталь, вартість готового виробу — 24 грошові одиниці за деталь. Процес складання і випалу здійснюється цілодобово (у три зміни) без вихідних.

Скласти модель на мові GPSS World, здійснити моделювання протягом місяця з різними кількостями складальників і визначити оптимальну кількість складальників, що максимізує прибуток. Визначити також середню і максимальну довжини черги, середню і максимальну тривалості перебування в черзі та середню і максимальну тривалості обробки деталей.

**Розв'язання.** За одиницю модельного часу беремо хвилину. На роль транзактів у цій задачі найкраще підходять складальники. Однак після завершення виготовлення деталі складальники не припиняють роботу. Оскільки після завершення зміни їхнє місце роботи займають інші складальники, для спрощення моделі можна вважати, що складальники працюють вічно і їх достатньо створити лише на початку процесу моделювання. Отже, блок GENERATE для створення складальників слід записувати особливим чином (див. нижче текст моделі).

Для моделювання процесу випалу деталей у печі викори-

стовуємо одноканальний пристрій, а для визначення основних характеристик черги — пару блоків **QUEUE** і **DEPART**. Проте цього недостатньо для визначення максимального часу перебування заготовки деталі в черзі до печі. Знайти його можна з детальної статистики черги (команда **QTABLE**). Після завершення виготовлення деталі потрібно виконати безумовну передачу транзакта на блок початку виготовлення наступної деталі (блок **TRANSFER**). Для визначення часу обробки деталей зручно використовувати таблицю, для збору інформації в яку потрібні блоки **MARK** і **TABULATE**. Для моделювання процесу виробництва протягом тридцяти діб створюємо таймер з блоків **GENERATE** і **TERMINATE**, а команду **START** використовуємо з одиничним операндом.

Оптимальну кількість складальників доводиться шукати перебором, причому надто мала або велика кількість складальників призводить до збитків. Кількість виготовлених деталей можна визначити з таблиці блоків у звіті, а потім за елементарною формулою знайти прибуток за місяць. Однак, оскільки такі розрахунки потрібно здійснювати для різного числа складальників, легше зробити так, щоб прибуток виводився безпосередньо у статистичному звіті. Для цього достатньо двох блоків **SAVEVALUE**, в одному з них треба накопичувати кількість виготовлених деталей, а другий потрібен для обчислення прибутку і його достатньо виконати один раз перед завершенням моделювання. Крім того, щоб не змінювати кількість складальників у двох місяцях моделі, достатньо занести цю кількість один раз перед початком моделювання в комірку пам'яті, а потім використовувати її як константу у блоках **GENERATE** і **SAVEVALUE**. Наведемо текст моделі для роботи 10 складальників:

```
tabd TABLE MP1,1,1,100
tabq QTABLE qkiln,1,1,100
INITIAL X$fit,10
GENERATE ,,X$fit
```

```
assem MARK 1
ADVANCE 45,5
QUEUE qkiln
SEIZE kiln
DEPART qkiln
ADVANCE 7.5,2.5
RELEASE kiln
TABULATE tabd
SAVEVALUE det+,1
TRANSFER ,assem
GENERATE (60#24#30)
SAVEVALUE prof,(X$det#(24-7)-600#30-10#X$fit#24#30)
TERMINATE 1
START 1
```

Після завершення моделювання комірка пам'яті `prof` міститиме прибуток для заданої у третьому рядку програми кількості складальників. Пропонуємо читачам самостійно переконатись у тому, що найбільший прибуток забезпечує сім складальників, а також знайти для цього випадку всі потрібні характеристики. ■

У наступному розділі розглядаються додаткові можливості мови GPSS World. Інші приклади імітаційного моделювання з використанням GPSS World і вправи для самостійного моделювання можна знайти в посібниках [2, 5, 11].

## Питання до розділу 6

1. Як запустити процес трансляції програми, написаної на мові GPSS World?
2. Назвіть основні категорії об'єктів мови GPSS. Що таке транзакт?
3. Що таке системні числові атрибути, який вигляд вони можуть мати при використанні мови GPSS?
4. Назвіть основні елементи рядка програми на мові GPSS.

5. Які арифметичні і логічні операції можна використовувати при побудові виразів на мові GPSS World?
6. Опишіть блоки створення, знищення, затримки та зміни параметрів і пріоритету транзактів.
7. Для чого використовують блоки SEIZE, RELEASE, PREEMPT, RETURN, ENTER і LEAVE? Опишіть їх.
8. Для чого використовують блоки TABULATE, MARK, QUEUE, DEPART і команди TABLE, QTABLE? Опишіть їх.
9. Як користуватись змінними, логічними перемикачами і комірками пам'яті? Які команди і блоки використовують для роботи з ними?
10. Які блоки дозволяють змінювати маршрут руху транзактів? Опишіть їх.
11. Як створюють функції на мові GPSS?
12. Назвіть основні математичні функції і функції для генерації випадкових величин мови GPSS World.

### Вправи до розділу 6

1. Читальний зал на факультеті обслуговує читачів лише під час перерв між парами і після пар. У читальному залі працює один бібліотекар. Час між появами читачів — рівномірно розподілена випадкова величина на проміжку  $[0, 30]$  секунд. Читачі бувають двох типів, причому їхня кількість однакова. Читачі першого типу хочуть здати книжку і цей процес займає 10–20 секунд. Читачі другого типу хочуть взяти книжку. Цей процес вимагає від 20 до 120 секунд. Бібліотекар не може обслуговувати кілька читачів одночасно. Складіть модель на мові GPSS World, яка б дозволила змоделювати роботу читального залу протягом 30 хвилин перерви, визначте довжину черги і статистику розподілу часу, потрібного для обслуговування читача.
2. Змініть модель з вправи 1 так, щоб час між появами читачів був рівномірно розподіленою випадковою величиною на проміжку  $[0, 30 + h/60]$  секунд, де  $h$  — кількість секунд, які минули від відкриття читального залу (тобто на початку перерви приходиться найбільше читачів).
3. У магазині є два продавці. Якщо обидва продавці зайняті обслуговуванням покупців, то до них формується спільна черга. Вхідний потік покупців не залежить від передісторії і є стаціонарним

пуассонівським (найчастіше саме так і є). Тому час появи покупців є експоненціально розподіленою випадковою величиною з інтенсивністю 0,9 за хвилину. Час обслуговування покупця продавцем теж є експоненціально розподіленою випадковою величиною з середнім значенням 2 хвилини. Складіть модель на мові GPSS, яка б дозволила змоделювати роботу магазину протягом 12 годин, отримати статистику розподілу часу, потрібного для обслуговування покупця, і довжину черги.

4. У перукарні є  $n$  перукарів. Вхідний потік клієнтів має експоненціальний розподіл з інтенсивністю 0,21 за хвилину. Якщо один з перукарів є вільним, то він обслуговуватиме чергового клієнта. Якщо всі перукарі зайняті, то клієнти стають у спільну чергу. Якщо довжина черги досягає значення 10, то наступні клієнти не займають чергу, а йдуть шукати іншу перукарню. Час обслуговування клієнта будь-яким перукарем має експоненціальний розподіл з середнім значенням 15 хвилин. Складіть модель роботи системи на мові GPSS, шляхом моделювання визначте мінімальну кількість перукарів, необхідну для того щоб середня довжина черги була меншою від 5. Визначте ймовірність того, що клієнту доведеться шукати іншу перукарню, максимальну довжину черги, коефіцієнт завантаження перукарні та статистику розподілу часу, необхідного для обслуговування клієнтів.

5. Тривалість перебування відвідувача в картинній галереї є рівномірно розподіленою випадковою величиною на проміжку  $[0,5; 2,5]$  години. Відвідувачі приходять у картинну галерею з інтервалом, який є випадковою величиною з експоненціальним законом розподілу з середнім значенням 4 хвилини. Кількість відвідувачів, які одночасно перебувають у картинній галереї, не обмежується. Складіть модель на мові GPSS, яка б дозволила здійснити моделювання перебування відвідувачів у картинній галереї протягом десяти годин її роботи і встановити те, яка найбільша кількість відвідувачів знаходилась у картинній галереї одночасно.

6. У відділенні зв'язку є два віконечка, в яких приймають рекомендовані листи і продають марки. Час обслуговування одного клієнта є випадковою величиною, рівномірно розподіленою на проміжку  $[1, 6]$  хвилин. Якщо обидва віконечка зайняті, то клієнти займають спільну чергу, з якої переходять до будь-якого вільного віконечка. Час появи клієнтів утворює стаціонарний пуассонівський

потік з середнім значенням 1,5 хвилини. Складіть модель на мові GPSS, яка б дозволила змоделювати процес обслуговування 100 клієнтів і допомогла зробити директору відділення зв'язку висновок про необхідність взяття на роботу ще одного працівника на основі аналізу довжини черг і часу, потрібного на перебування в черзі.

7. На автозаправній станції є місця для одночасної заправки трьох автомобілів і достатня кількість працівників. Тривалість заправки кожного автомобіля становить 7–15 хвилин. Автомобілі, яким бракує пального, прибувають через кожні 4–10 хвилин. Якщо на автозаправній станції всі місця зайняті, то автомобілі їдуть на іншу автозаправку. Складіть модель на мові GPSS, яка б дозволила здійснити моделювання роботи автозаправної станції протягом обслуговування 200 автомобілів (крім тих, які у зв'язку із зайнятістю всіх місць для заправки поїхали шукати іншу автозаправну станцію). Визначте кількість і відсоток автомобілів, водії яких не захотіли стояти в черзі.

8. Час появи замовлень у одноканальній системі масового обслуговування визначається функцією з випадковим аргументом, графік якої зображено на рис. 6.13. Час обслуговування задається функцією з випадковим аргументом, графік якої наведено на рис. 6.14. Виконайте моделювання обслуговування 200 замовлень у такій системі масового обслуговування, визначте основні характеристики черги і час перебування замовлень у моделі. Функцію з рис. 6.13 закодуйте не менше, ніж п'ятьма точками.

9. Час появи замовлень у системі масового обслуговування визначається функцією з випадковим аргументом, графік якої зображено на рис. 6.15. Час обслуговування задається функцією з випадковим аргументом, графік якої наведено на рис. 6.16. Одночасно обслуговуватись може довільна кількість замовлень. Виконайте моделювання обслуговування 200 замовлень у такій системі масового обслуговування, визначте час перебування замовлень у моделі. Функцію з рис. 6.16 закодуйте не менше, ніж п'ятьма точками.

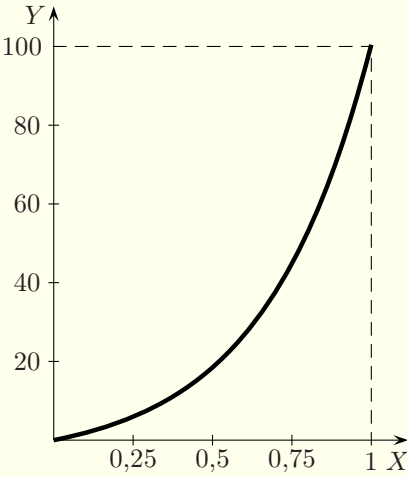


Рис. 6.13.

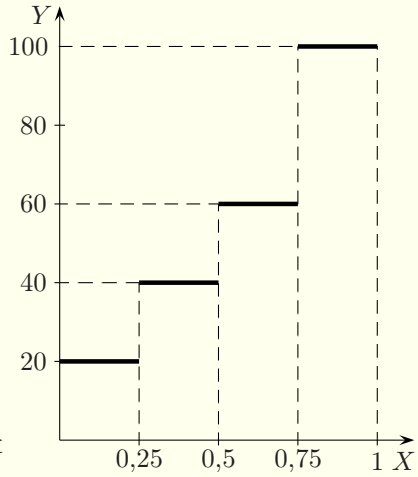


Рис. 6.14.

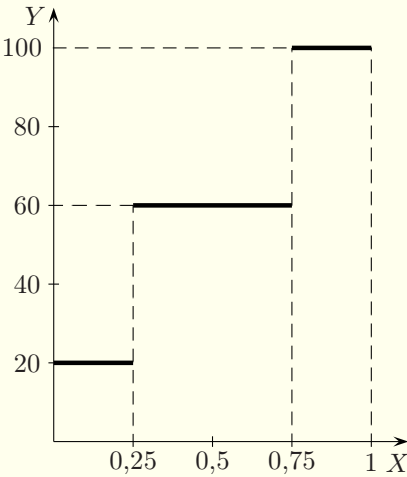


Рис. 6.15.

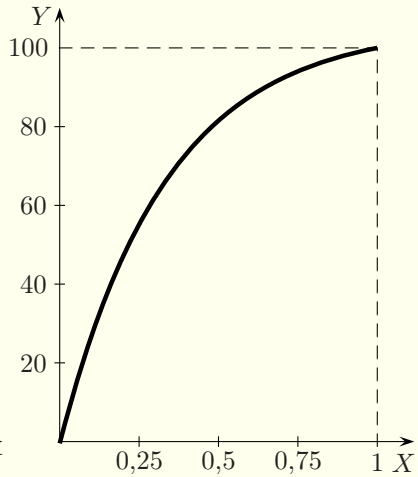


Рис. 6.16.

10. На виробничу дільницю надходять деталі. Час їхнього надходження розподілений рівномірно на проміжку [5, 15] хвилин. Спочатку вони проходять зовнішній прийом у контролера відділу технічного контролю за 2–4 хвилини. 5 % з них скеровуються на доопрацювання, яке займає 3–5 хвилин, після чого вони знову надходять на перевірку до контролера на час 1–3 хвилини. Деталі, які вже

побували на доопрацюванні і знову не проходять контроль якості, визнаються бракованими (в середньому їх буває 10 % від кількості тих, що були на доопрацюванні). Після прийому деталі надходять на складання протягом 2–7 хвилин, а потім на термозагартування на час 8–12 хвилин. Деталі, що гартувалися менше 10 хвилин або були на доопрацюванні, одержують маркування другого сорту, а якщо ці деталі були на доопрацюванні і гартування тривало менше 10 хвилин — третього.

Завдання:

а) склавши модель на мові GPSS, змоделюйте роботу дільниці протягом 22 робочих днів тривалістю вісім годин кожен;

б) визначте:

- кількість бракованих деталей;
- кількість деталей першого сорту;
- кількість деталей другого сорту;
- кількість деталей третього сорту;

в) визначте як зміняться ці дані при зменшенні браку до 2 %.

11. Станція технічного обслуговування автомобілів ремонтує машини трьох типів: *A*, *B* і *C*. Автомобілі всіх типів надходять під час робочого дня через проміжки часу, які мають експоненціальний закон розподілу, причому для машин типу *A* інтенсивність становить 0,015 за хвилину, типу *B* — 0,02 за хвилину, типу *C* — 0,011 за хвилину. На станції технічного обслуговування працюють чотири бригади слюсарів. Ремонт машини типу *A* є випадковою величиною, рівномірно розподіленою на проміжку [36, 110] хвилин, типу *B* — [30, 125] хвилин, типу *C* — [30, 150] хвилин. Стоянка перед станцією технічного обслуговування має 15 місць, а кожна бригада має свій бокс на одну машину. Машини, для яких не знайшлось місця в боксі чи на стоянці, ремонтуються в іншому місці. Станція технічного обслуговування працює в середньому 22 дні на місяць по вісім годин на день.

Завдання:

а) складіть модель на мові GPSS і змоделюйте роботу станції технічного обслуговування протягом місяця та року;

б) знайдіть розподіл часу обслуговування автомашин кожного типу, ймовірність відмови в обслуговуванні, коефіцієнт завантаження слюсарів роботою, коефіцієнт заповнення стоянки, середню кількість боксів, що працюють, та середню кількість зайнятих місць на

стоянці;

в) визначте необхідну кількість місць на стоянці для виключення випадків відмови в обслуговуванні та всі величини пункту б) у цьому випадку;

г) знайдіть необхідну місткість боксів для виключення випадків відмови в обслуговуванні та всі величини пункту б) у цьому випадку.

12. Вулиці, що виходять на чотиристороннє перехрестя, мають позначення за напрямом руху годинникової стрілки:  $A$ ,  $B$ ,  $C$ ,  $D$ . З боку вулиці  $A$  автомобілі під'їжджають до перехрестя кожні  $4 \pm 2$  секунди, причому 30 % з них повертають праворуч у напрямі  $A-D$ , а 25 % — ліворуч у напрямі  $A-B$ . Поворот наліво можливий, якщо немає руху в напрямі  $C-A$ . З боку вулиці  $C$  автомашини під'їжджають до перехрестя кожні  $7 \pm 3$  секунд, причому 60 % із них проїжджають прямо у напрямі  $C-A$ , а 40 % — направо у напрямі  $C-B$ . Поворот ліворуч у напрямі  $C-D$  заборонений. Світлофор на перехресті перемикається кожні 20 секунд (жовтий сигнал світлофора можна не враховувати, бо він визначає рух відповідно до попереднього сигналу й є лише попередженням про зміну руху). Ширина всіх вулиць допускає рух у три ряди в кожному напрямі. Машини долають перехрестя в будь-якому напрямі за 2 секунди. Автомобіль, що виїхав на перехрестя до моменту перемикавання світлофора, обов'язково продовжує рух. На перехресті може знаходитись не більше однієї машини для кожного напрямку руху.

Змодельуйте рух автомобілів через перехрестя з боку вулиць  $A$  і  $C$  протягом години. Підрахуйте кількість автомашин, що пройшли в кожному напрямі. Визначте середню і максимальну довжини черги автомобілів, середній час перебування автомобілів у черзі для кожного напрямку руху.

13. Система складається з одного бульдозера, двох самоскидів і одного механізованого навантажувача. Бульдозер згрібає землю до навантажувача. Для початку навантаження перед навантажувачем повинна лежати хоча б одна купа землі. Час, що витрачається бульдозером на підготовку купи землі, має експоненціальний розподіл з середнім значенням 10 хвилин. Крім наявності куп землі, для роботи навантажувача потрібен порожній самоскид. Про час навантаження відомо лише те, що він розподілений на відрізьку  $[10, 18]$  хвилин. Після того, як самоскид завантажений, він їде до місця

розвантаження, розвантажується і знову повертається для завантаження. Час перебування самоскида в дорозі розподілено логнормально, причому у завантаженому стані середнє значення відповідного нормального закону розподілу становить 2, а в порожньому — 1,7. Середньоквадратичне відхилення відповідного нормального закону розподілу в обох випадках дорівнює 1,5. Час розвантаження розподілено рівномірно на інтервалі від 2 до 8 хвилин. Після завантаження кожного самоскида навантажувач повинен «відпочивати» протягом 5 хвилин, а потім він знову може розпочинати роботу.

Складіть модель на мові GPSS World для моделювання роботи бульдозера, навантажувача і самоскидів протягом року. Бульдозер працює з 9 до 17 години п'ять днів на тиждень, навантажувач — з 9 до 16 години. Самоскиди в кінці робочого дня мусять завершити рейс і повернутись до навантажувача. Визначте характеристики черг партій вантажу, самоскидів і навантажувача. Дослідіть можливість зменшення черг.

## Розділ 7. Додаткові можливості мови і середовища GPSS World

### § 7.1. Створення копій транзактів і синхронізація їхнього руху

У деяких випадках виникає потреба одночасного створення кількох транзактів або моделювання одночасного обслуговування одного й того самого транзакта кількома пристроями. У цих випадках доводиться створювати копії транзактів. Для створення копій транзактів слугує блок

SPLIT *A,B,C*.

Обов'язковий операнд *A* вказує кількість створюваних копій транзакта. Необов'язковий операнд *B* може вказувати мітку або номер блока, в який потрібно відправити копії (оригінал транзакта йде в наступний блок). Необов'язковий операнд *C* може вказувати ім'я того параметра транзакта, яким будуть відрізнятися між собою копії транзакта (у кожній наступній копії він буде на одиницю більшим). Всі інші параметри транзакта повторюються у копіях без змін. Блок SPLIT 0 не створює жодної копії транзакта.

Приклад 1:

SPLIT 5 — створюється п'ять ідентичних копій транзакта, який зайшов у цей блок. Оригінал транзакта і всі його копії переходять у наступний блок.

Приклад 2:

ASSIGN 10,25

SPLIT 4,P10,10

У другому прикладі створюється чотири копії транзакта, який зайшов у блок. Оригінал транзакта переміщується у наступний блок (при цьому значення його параметра з ім'ям 10

збільшується на одиницю), його перша копія — у блок з номером 27, друга — у блок з номером 28, третя — у блок з номером 29, а четверта — у блок з номером 30.

Оригінал транзакта і всі його копії належать до однієї сукупності транзактів, яку називають *сім'єю*. Члени однієї сім'ї транзактів пов'язані між собою. Номер сім'ї транзактів визначається номером транзакта, який створив цю сім'ю. Для транзакта, створеного блоком **GENERATE**, номер сім'ї збігається з номером транзакта.

З кількох транзактів однієї сім'ї можна створити один транзакт. Для цього використовується блок **ASSEMBLE**  $A$ , де  $A$  — кількість транзактів, які об'єднуються в один транзакт. Транзакти, які входять в блок **ASSEMBLE**, затримуються в ньому до тих пір, доки у цьому блоці не назбирається потрібна кількість транзактів однієї сім'ї. Після цього всі вони об'єднуються в один транзакт, який виходить з блока **ASSEMBLE**. Транзакти, які належать різним сім'ям, не об'єднуються і можуть затримуватись у блоці **ASSEMBLE** на необмежений час. Приклад:

**ASSEMBLE 3** — у блоці об'єднується три транзакти однієї сім'ї в один транзакт.

Блок **GATHER**  $A$  накопичує  $A$  транзактів однієї сім'ї. Транзакти затримуються в блоці **GATHER** до тих пір, поки їхня кількість не стане рівною  $A$ . Після цього всі вони одночасно виходять з блока в тому порядку, в якому вони туди потрапили.

Блок **ADOPT**  $A$  переводить всі транзакти, які в нього потрапили, у сім'ю з номером  $A$  ( $A$  — натуральне число). Якщо блок **ADOPT** використовувати перед блоком **ASSEMBLE** або **GATHER**, то ці блоки будуть збирати довільні транзакти.

Блок **MATCH**  $A$  синхронізує рух двох транзактів однієї сім'ї. Операндом  $A$  є мітка або номер спряженого блока **MATCH**. *Спряженими* називають такі два блоки **MATCH**, що операнд першого блока містить мітку або номер другого блока, а опе-

ранд другого — мітку або номер першого блока. Довільний транзакт, досягнувши одного з блоків MATCH, затримується в ньому до тих пір, поки інший транзакт тієї самої сім'ї не досягне спряженого з ним блока MATCH. Після цього обидва транзакти починають синхронно рухатись по моделі. Приклад:

```
aaa MATCH bbb
...
bbb MATCH aaa
```

## § 7.2. Матриці

У мові GPSS World передбачена можливість роботи з матрицями комірок пам'яті. Для створення двовимірної матриці з ім'ям  $N$  використовується команда опису

$N$  MATRIX  $A, B, C$ .

Операнд  $A$  не використовується (залишений для сумісності зі старими версіями програми), операнд  $B$  визначає кількість рядків матриці, а операнд  $C$  — кількість стовпців матриці.

Для читання елементів матриць використовується системний числовий атрибут  $M\alpha(a, b)$ , де  $\alpha$  — ім'я матриці, а натуральні числа  $a$  і  $b$  — відповідно номер рядка і стовпця в матриці, на перетині яких розташовано елемент.

Після створення матриці командою MATRIX всі її елементи містять нульове значення. Інші початкові значення елементам матриці можуть бути присвоєні командою

INITIAL  $A, B$ ,

де  $A$  — системний числовий атрибут елемента матриці або ім'я матриці, а  $B$  — початкове значення. Приклади:

abcd MATRIX ,2,3 — створюється матриця з ім'ям abcd, яка має два рядки і три стовпці;

INITIAL abcd,10 — всі елементи матриці abcd на початку моделювання будуть містити число 10;

`INITIAL MX$abcd(1,2),5` — елементу матриці `abcd` з перетину першого рядка і другого стовпця на початку моделювання присвоюється число 5;

`MX$abcd(1,3)` — звертання до елемента матриці `abcd`, який знаходиться на перетині першого рядка і третього стовпця.

Блок `MSAVEVALUE` використовується для збереження значень в елементах матриці під час моделювання, а також для збільшення чи зменшення цих значень. Він використовується у трьох форматах:

`MSAVEVALUE A,B,C,D` присвоює елементу матриці  $A$  з рядка  $B$  і стовпця  $C$  значення  $D$  (режим фіксації);

`MSAVEVALUE A+,B,C,D` збільшує елемент матриці  $A$  з рядка  $B$  і стовпця  $C$  на величину  $D$  (режим збільшення);

`MSAVEVALUE A-,B,C,D` зменшує елемент матриці  $A$  з рядка  $B$  і стовпця  $C$  на величину  $D$  (режим зменшення).

Приклад роботи з матрицею наведено у § 7.9.

Матриці можуть мати розмірність більшу від двох (три, чотири, п'ять чи шість). Для цього в команді опису `MATRIX` використовують додаткові операнди:

$$N \text{ MATRIX } A,B,C,D,E,F,G.$$

Присвоїти значення елементу матриці з розмірністю, більшою від двох, можна з допомогою спеціальних процедур користувача (§ 7.9). Блок `MSAVEVALUE` дозволяє здійснювати присвоювання значень елементам матриці з одиницями в індексах для всіх розмірностей, вище від другої.

### § 7.3. Найважливіші системні числові атрибути

**Системні числові атрибути процесу моделювання.**

`RN $\alpha$`  — генератор випадкових чисел  $\alpha$  (генерує ціле випадкове число на проміжку  $[0, 999]$ ).

`AC1` — абсолютний модельний час.

`C1` — відносний модельний час.

**TG1** — поточне значення лічильника завершень (початкове значення лічильника задається командою **START**, значення **TG1** зменшується при вході транзакта в блок **TERMINATE** з додатним операндом).

**Z1** — поточна вільна системна пам'ять.

### **Системні числові атрибути транзактів.**

$P\alpha$  — значення параметра  $\alpha$  активного транзакта.

**PR** — пріоритет активного транзакта.

**XN1** — номер активного транзакта (натуральне число).

**M1** — час перебування в моделі активного транзакта.

$MP\alpha$  — час проходження транзактом деякої частини моделі (різниця абсолютного модельного часу і значення параметра  $\alpha$  цього транзакта:  $AC1 - P\alpha$ , це значення параметра  $\alpha$  формується блоком **MARK**  $\alpha$ , який є рівносильним до **ASSIGN**  $\alpha, AC1$ ).

**A1** — номер сім'ї активного транзакта.

### **Системні числові атрибути блоків.**

$N\alpha$  — загальна кількість входжень транзактів до блока з міткою або номером  $\alpha$ .

$W\alpha$  — поточна кількість транзактів у блоці з міткою або номером  $\alpha$ .

### **Системні числові атрибути одноканальних пристроїв.**

$F\alpha$  дорівнює 1, якщо одноканальний пристрій  $\alpha$  є зайнятим, інакше — 0.

$FI\alpha$  дорівнює 1, якщо обслуговування в одноканальному пристрої  $\alpha$  було щойно перерване, інакше — 0.

$FV\alpha$  дорівнює 1, якщо одноканальний пристрій  $\alpha$  є доступним, інакше — 0.

$FR\alpha$  — коефіцієнт використання одноканального пристрою  $\alpha$  (ціле число від 0 до 1000).

$FS\alpha$  — кількість транзактів, які займали одноканальний пристрій  $\alpha$ .

$FT\alpha$  — середній час використання одноканального пристрою  $\alpha$  одним транзактом.

**Системні числові атрибути багатоканальних пристроїв.**

$S\alpha$  — кількість зайнятих каналів у багатоканальному пристрої  $\alpha$ .

$R\alpha$  — кількість вільних каналів у багатоканальному пристрої  $\alpha$ .

$SR\alpha$  — коефіцієнт використання багатоканального пристрою  $\alpha$  (ціле число від 0 до 1000).

$SA\alpha$  — середня кількість зайнятих каналів у багатоканальному пристрої  $\alpha$ .

$SM\alpha$  — максимальна кількість зайнятих каналів у багатоканальному пристрої  $\alpha$ .

$SC\alpha$  — загальна кількість входжень транзактів до багатоканального пристрою  $\alpha$ .

$ST\alpha$  — середній час використання одного каналу багатоканального пристрою  $\alpha$ .

$SF\alpha$  дорівнює 1, якщо багатоканальний пристрій  $\alpha$  є порожнім, інакше — 0.

$SF\alpha$  дорівнює 1, якщо багатоканальний пристрій  $\alpha$  є повністю зайнятим, інакше — 0.

$SV\alpha$  дорівнює 1, якщо багатоканальний пристрій  $\alpha$  є доступним, інакше — 0.

**Системні числові атрибути черг.**

$Q\alpha$  — поточна довжина черги  $\alpha$ .

$QA\alpha$  — середня довжина черги  $\alpha$ .

$QM\alpha$  — максимальна довжина черги  $\alpha$ .

$QC\alpha$  — загальна кількість входжень транзактів до черги  $\alpha$ .

$QZ\alpha$  — кількість входжень транзактів із нульовим часом перебування їх у черзі  $\alpha$ .

$QT\alpha$  — середній час перебування транзакта в черзі  $\alpha$ .

$QX\alpha$  — середній час перебування транзакта в черзі  $\alpha$  без врахування входжень транзактів із нульовим часом перебування їх у черзі.

#### **Системні числові атрибути таблиць.**

$TV\alpha$  — середнє значення аргументу таблиці  $\alpha$ .

$TC\alpha$  — кількість введень інформації у таблицю  $\alpha$ .

$TD\alpha$  — середньоквадратичне відхилення аргументу таблиці  $\alpha$ .

#### **Системні числові атрибути комірок пам'яті, матриць, змінних, функцій і логічних перемикачів.**

$X\alpha$  — вміст комірки пам'яті  $\alpha$ .

$MX\alpha(a, b)$  — вміст елемента матриці  $\alpha$ , що знаходиться на перетині рядка  $a$  і стовпця  $b$ .

$FN\alpha$  — результат обчислення функції  $\alpha$ .

$V\alpha$  — результат обчислення дійсної змінної  $\alpha$ .

$BV\alpha$  — результат обчислення логічної змінної  $\alpha$ .

$LS\alpha$  — стан логічного перемикача  $\alpha$  (1 — увімкнений, 0 — вимкнений).

Системні числові атрибути для списків користувача наведено у § 7.7.

**Непряма адресація.** Скрізь, де може розміщуватись системний числовий атрибут, може стояти також клас системного числового атрибута й ім'я параметра транзакта, відокремлені символом «\*». Такий запис означає, що як ім'я системного числового атрибута буде використовуватись аргумент відповідного параметра конкретного транзакта. Приклади:

$X*5$  — комірка пам'яті, ім'я чи номер якої занесено до P5, тобто до параметра з номером 5 транзакта;

$Q*aska$  — черга з ім'ям, яке зберігається у параметрі  $aska$  транзакта.

Непряма адресація дозволяє скоротити обсяг програми.

## § 7.4. Команди керування

Для керування процесом моделювання і налагодження моделей можна використовувати спеціальні команди керування. Їх викликають з допомогою меню **Command**, для їхнього виконання також призначені певні комбінації клавіш.

Команда **START A**, де  $A$  — початкове значення лічильника завершень, запускає процес моделювання.

Команда **HALT** припиняє процес моделювання. На відміну від решти команд керування і команд опису ця команда не стає у чергу команд, а виконується миттєво. Клавіша виклику — **F4**.

Команда **CONTINUE** продовжує процес моделювання, якщо лічильник завершень не дорівнює нулю. Вона може використовуватись після команди **HALT**, а також після зупинки моделювання командою **STOP**. Клавіша виклику — **F2**.

Команда **CLEAR** повертає процес моделювання до початкового стану. При цьому всі пристрої звільняються, всі транзакти видаляються з моделі, знищується вся накопичена статистика, абсолютний і відносний модельний час занулюється, але генератори випадкових чисел не переводяться у початковий стан.

Команда **RESET** запроваджує новий етап збору статистики. Для цього вона знищує накопичену статистику і занулює відносний модельний час. На відміну від команди **CLEAR** всі транзакти залишаються в моделі і не знищується абсолютний модельний час.

Команда **STEP A** запускає процес моделювання заданої кількості  $A$  входжень транзактів до блоків.

Команда **STEP 1** використовується для покрокового виконання процесу моделювання. Клавіша виклику — **F5**.

Команда **STOP A,B,C** використовується для застосування або скасування умови зупинки процесу моделювання. Цю команду можна записувати в тексті моделі, як і команду **START**.

Команди **STOP** немає в меню **Command**, але в меню **Command** є пункт **Custom**, з допомогою якого можна давати будь-яку команду керування. Операнди: *A* — номер транзакта (нумеруються від одиниці), *B* — номер або мітка блока, *C* — стан умови, ним може бути **ON** (додати умову зупинки) чи **OFF** (скасувати умову зупинки), за замовчуванням **ON**. Всі операнди є необов'язковими, але принаймні один з них має бути. Якщо пропущено операнд *A*, то будь-який транзакт буде задовольняти умову зупинки. Якщо пропущено операнд *B*, то будь-який блок задовольнятиме умову зупинки. Одночасно може діяти кілька умов зупинки, створених кількома командами **STOP**.

Приклад: **STOP 13,7** — задається умова, яка приведе до зупинки процесу моделювання в той момент, коли транзакт з номером 13 намагатиметься увійти до блока з номером 7. Після створення умови зупинки застосовують команду **START** з достатнім значенням лічильника завершень. Якщо виконується умова зупинки, то процес моделювання припиняється й у вікно **Journal** передається системний час, номер активного транзакта, а також блок, в який цей транзакт намагається увійти.

Для продовження моделювання використовують команду **CONTINUE**. Вона не скасовує умови зупинки. Якщо якась з цих умов знову виконається, то процес моделювання знову зупиниться. Для скасування всіх умов зупинки використовують команду **STOP ,OFF**.

Наведемо ще кілька прикладів застосування команди **STOP**:

**STOP 15** — процес моделювання зупиниться тоді, коли транзакт з номером 15 стане активним;

**STOP 15,OFF** — скасовує умову, накладену попередньою командою;

**STOP ,mit2** — процес моделювання зупиниться тоді, коли будь-який транзакт спробує зайти у блок з міткою **mit2**;

`STOP ,mit2,OFF` — скасовує умову, накладену попередньою командою.

Команда `SHOW A` дозволяє обчислити вираз  $A$ , результат обчислення записується у вікні `Journal` і в рядку стану, при цьому процес моделювання має бути зупиненим. Команда `SHOW` може використовуватись для перегляду змінних, комірок пам'яті, логічних перемикачів, різноманітних системних числових атрибутів на будь-якому кроці моделювання. Команда `SHOW` пов'язана з командами `STEP`, `STOP` і `HALT`.

Приклади:

`SHOW XN1` — показується номер активного транзакта.

`SHOW N$mit1/N$mit2` — виводиться результат ділення кількості транзактів, що увійшли у блок з міткою `mit1`, на кількість транзактів, які увійшли у блок з міткою `mit2`.

Команда `RMULT A,B,C,D,E,F,G` задає початкові значення для генераторів випадкових чисел з номерами від 1 до 7. Перший операнд відповідає генератору випадкових чисел `RN1`, другий — `RN2` і т. д. Всі операнди є необов'язковими, але принаймні один з них має бути. За замовчуванням початкове значення генератора випадкових чисел дорівнює його номеру.

Команда `INCLUDE A` може розміщуватись у тексті програми для зчитування і трансляції додаткового файлу з командами чи блоками моделі. Операндом  $A$  має бути ім'я файлу або повний шлях доступу до файлу у подвійних лапках, наприклад: `INCLUDE "Chastyna1.txt"`. Цей файл обробляється так, ніби його вміст знаходиться на місці команди `INCLUDE`.

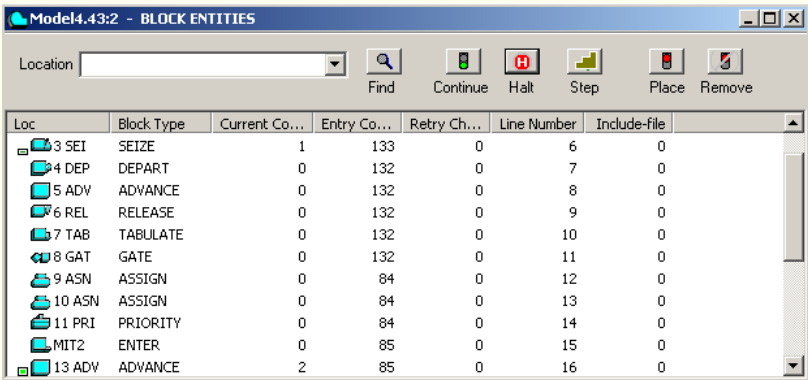
## § 7.5. Інформаційні вікна

Для спостереження за процесом моделювання на етапі тестування і верифікації, а також для отримання додаткової інформації, відсутньої у статистичному звіті, використовують десять динамічних інформаційних вікон середовища GPSS World. Доступ до них відкриває команда з меню `Window►Simulation Window`. Існують наступні вікна:

- Blocks (блоки);
- Expressions (вирази);
- Facilities (одноканальні пристрої);
- Logicswitches (логічні перемикачі);
- Matrix (матриця);
- Plots (графіки);
- Queues (черги);
- Savevalues (комірки пам'яті);
- Storages (багатоканальні пристрої);
- Table (таблиця).

Ці вікна можна переглядати як під час зупинки процесу моделювання чи після його завершення, так і під час процесу моделювання. В останньому випадку моделювання суттєво сповільнюється, але інформація у вікнах оперативного оновлюється відповідно до процесу моделювання.

Вікно **Blocks** є одним з найважливіших. Воно дає можливість візуально відстежувати пересування транзактів між блоками під час моделювання (рис. 7.1).



Loc	Block Type	Current Co...	Entry Co...	Retry Ch...	Line Number	Include-file
3 SEI	SEIZE	1	133	0	6	0
4 DEP	DEPART	0	132	0	7	0
5 ADV	ADVANCE	0	132	0	8	0
6 REL	RELEASE	0	132	0	9	0
7 TAB	TABULATE	0	132	0	10	0
8 GAT	GATE	0	132	0	11	0
9 ASN	ASSIGN	0	84	0	12	0
10 ASN	ASSIGN	0	84	0	13	0
11 PRI	PRIORITY	0	84	0	14	0
MIT2	ENTER	0	85	0	15	0
13 ADV	ADVANCE	2	85	0	16	0

Рис. 7.1

Аналогічно працюють вікна **Facilities**, **Logicswitches**, **Queues**, **Savevalues**, **Storages**, наприклад, на рис. 7.2 наведено вікно для черг.

Queue Entity	Current Content	Entry Count	Zero Entry Count	Maximum Content	Average Content	Average Time (..)
CHR	6	772	0	14	4.094	299.34

Рис. 7.2

Вікно **Expressions** призначене для спостереження за змінами будь-якої кількості виразів. Якщо команда **SHOW** обчислює лише один вираз у конкретний момент часу, то вікно **Expressions** дозволяє спостерігати зміну кількох виразів під час моделювання. Кожен вираз для обчислення і його назву вказують у полях **Expression** і **Label** вікна **Edit Expression Window** (яке з'являється після вибору з меню **Window►Simulation Window►Expression Window**), потім натискають кнопки **View** і **Memorize** (рис. 7.3). Ці кнопки дозволяють показувати значення виразів у поточному процесі моделювання та використовувати збережені вирази при повторному запуску процесу моделювання. Після введення всіх виразів натискають кнопку **OK**. Під час моделювання за зміною виразів спостерігають у вікні **Expressions** (рис. 7.4).

У вікнах **Blocks**, **Expressions**, **Facilities**, **Storages**, **Logicswitches**, **Queues** і **Savevalues** є кнопки **Continue**, **Halt** і **Step 1**, які можна використовувати для керування процесом моделювання під час перегляду цих вікон.

Вікно **Plots** дає змогу одночасно спостерігати за кількома графіками для заданих виразів. Команда меню **Window►Simulation Window►Plot Window** відкриває вікно **Edit Plot Window**, в якому для кожного графіка у полі **Label** задають назву графіка, а у полі **Expression** — вираз для його побудови, потім натискають кнопки **Plot** і **Memorize** (рис. 7.5). У полі **Title** можна вказати загальний заголовок для всіх графіків, у полі **Time Range** — часовий проміжок, графіки для

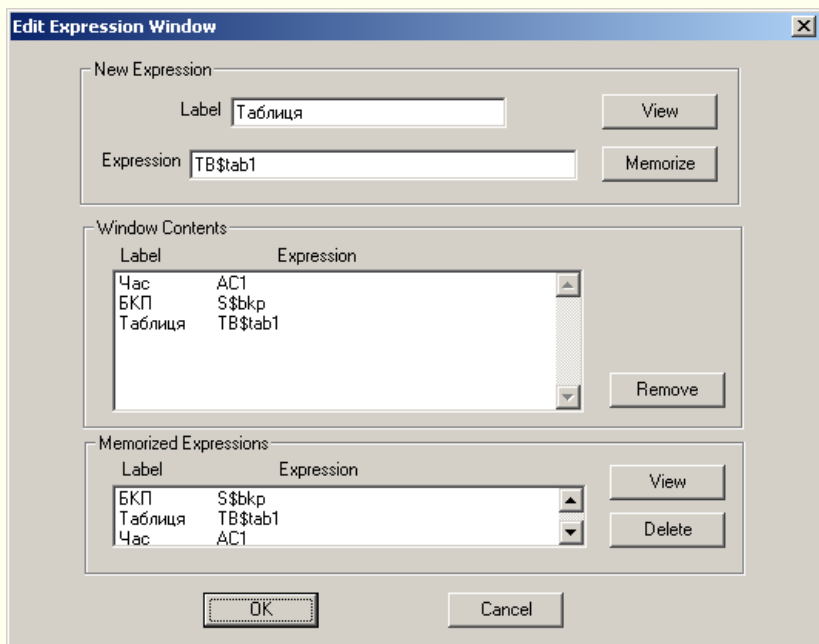


Рис. 7.3

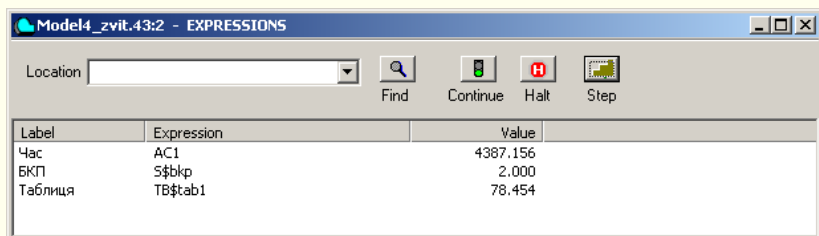


Рис. 7.4

якого будуть відображатись у вікні **Plots**, а в полях **Min Value** і **Max Value** — найменше і найбільше значення графіків для осі ординат. Після завершення введення інформації у вікні **Edit Plot Window** і натиснення на кнопку **OK** з'явиться порожнє вікно **Plots**. Для того щоб у ньому побачити графіки, треба запустити процес моделювання командою **Start** з меню **Command**. На рис. 7.6 показано графіки черги і коефіцієнта завантаження одноканального пристрою, переведеного в межі від 0 до 10. Якщо необхідно відредагувати деякі параметри відображення графіків, то використовують команду **Edit►Plot Window**.

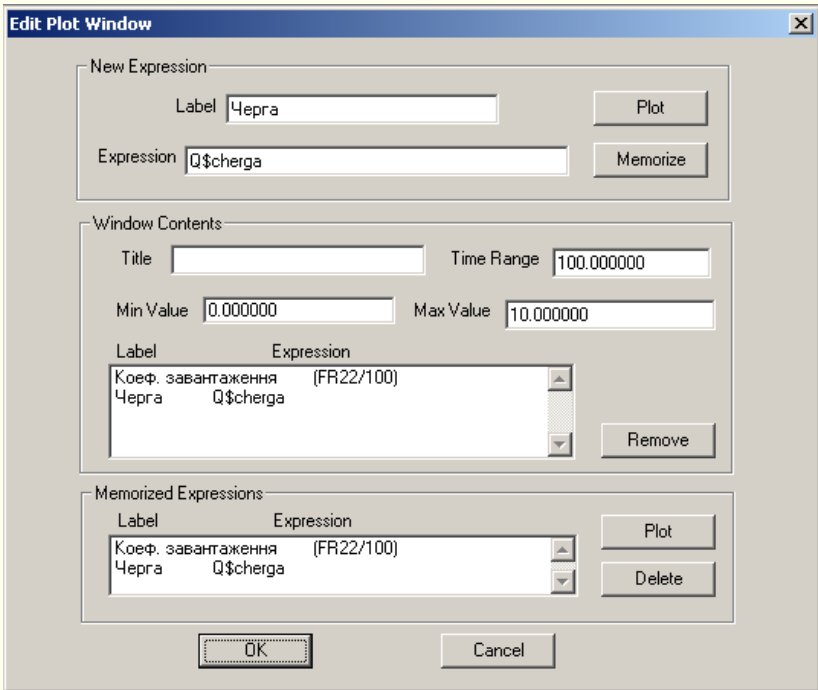


Рис. 7.5

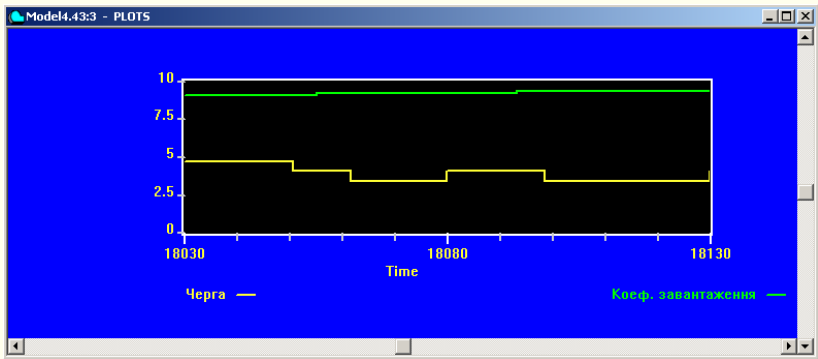


Рис. 7.6

Вікно Matrix використовується для відображення матриць (рис. 7.7).

TRANSACTS	
Dim 1	Dim 2
	44
4	22.360
5	24.839
6	28.407
7	24.515
8	20.904
9	17.549
10	21.470
11	30.659
12	13.114

Рис. 7.7

Вікно Table є динамічною гістограмою для таблиці (рис. 7.8). Для акуратного відображення гістограми потрібно добре підібрати всі операнди відповідної команди TABLE або QTABLE, зокрема кількість інтервалів у команді має відповідати кількості інтервалів у таблиці.

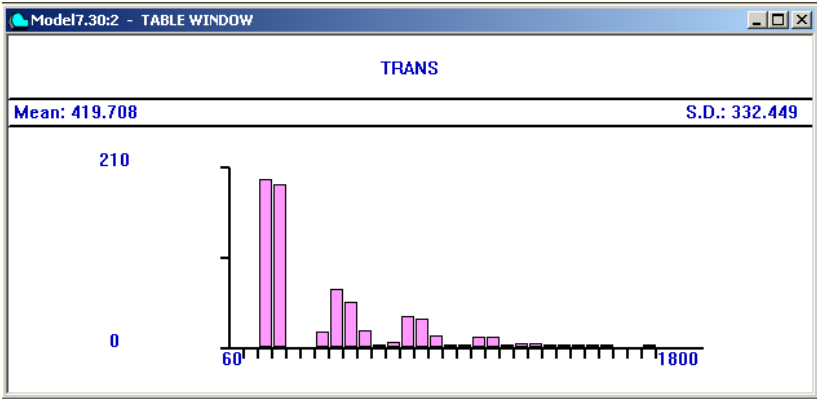


Рис. 7.8

Крім перегляду динамічних вікон можна робити кадри стану (Window►Simulation Snapshot) для:

- CEC (списку поточних подій);
- FEC (списку майбутніх подій);
- Transaction (окремого транзакта);
- Numeric Groups (числових груп);
- Transactions Groups (груп транзактів);
- Userchains (списків користувача).

На рис. 7.9 наведено приклад вікна для списку майбутніх подій.

### § 7.6. Імітаційне моделювання виходу з ладу обладнання

Для імітації виходу з ладу одноканального пристрою найкраще використовувати блок PREEMPT з пріоритетами, а імітувати несправність і її усунення (ремонт) — з допомогою спеціального транзакта з високим пріоритетом.

Для імітації виходу з ладу одного чи кількох каналів багатоканального пристрою можна замість багатоканального пристрою використовувати кілька одноканальних пристроїв, входи в які треба моделювати блоками PREEMPT з пріоритетами,

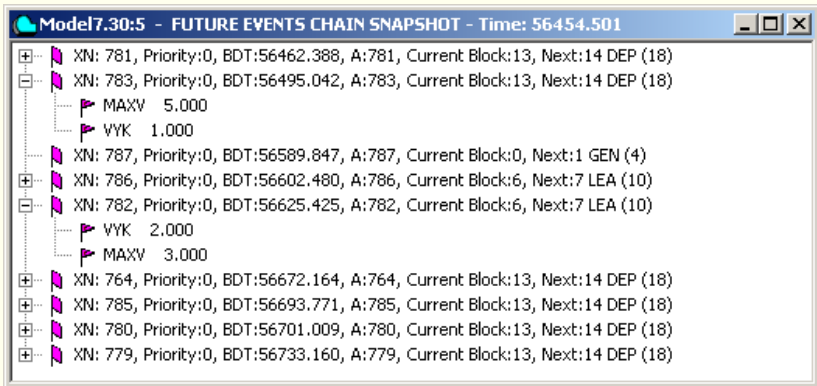


Рис. 7.9

а несправність і її усунення потрібно імітувати спеціальним транзактом (транзактами) з високим пріоритетом.

**Приклад 1.** На виробничій дільниці є два верстати. Якщо один з них виходить з ладу, то деталь буде чекати завершення ремонту або звільнення другого верстату. Потік надходження деталей є стаціонарним пуассонівським з середнім значенням 3. Час обслуговування — випадкова величина з логнормальним законом розподілу з середнім значенням 1 і середньоквадратичним відхиленням 1 відповідного нормального розподілу. Вихід з ладу верстату моделюється логнормальним законом розподілу з середнім значенням 6 і середньоквадратичним відхиленням 2 відповідного нормального розподілу. Час ремонту — випадкова величина, рівномірно розподілена на відріжку [1, 3]. Скласти модель роботи виробничої дільниці.

**Розв’язання.** Два верстати будемо моделювати двома блоками PREEMPT з пріоритетами. Нижче наводиться можливий лістинг моделі.

```
tab TABLE M1,1,1,100
GENERATE (Exponential(10,0,3))
ASSIGN 1,(Lognormal(11,0,1,1))
mit1 QUEUE cherga
```

```

TEST NE (F$prod1+F$prod2),2; чекати,
; доки не звільниться хоч один пристрій
DEPART cherga
GATE NU prod1,mit3; якщо зайнятий пристрій 1, то
; перейти до пристрою 2
mit2 PREEMPT prod1,PR,mit1,1,RE
ADVANCE P1
RETURN prod1
TRANSFER ,mit4
mit3 PREEMPT prod2,PR,mit1,1,RE
ADVANCE P1
RETURN prod2
mit4 TEST E PR,0,mit5
TABULATE tab
mit5 TERMINATE 1
GENERATE (Lognormal(12,0,6,2)); генерування виходу
; з ладу пристрою
PRIORITY 1
ASSIGN 1,(Uniform(13,1,3))
TRANSFER 0.5,mit2,mit3
START 1000

```

Недоліком цієї моделі є те, що транзакт, обслуговування якого було перерване, стає в кінець черги. Щоб він розміщувався на початку черги, для нього можна використовувати проміжний пріоритет (зробіть самостійно!) або створити спеціальну чергу з сортуванням транзактів по часу їхнього перебування в моделі (про такі черги буде йти мова в § 7.7). ■

Якщо у блоках PREEMPT з попередньої моделі вилучити по три останні операнди, то транзакт, обслуговування якого було перерване, ставатиме на початок черги, але лише до того самого пристрою, навіть якщо другий пристрій вільний і немає ніяких черг. Імітація виходу з ладу одного з більшої кількості каналів здійснюється аналогічно.

Для моделювання виходу з ладу всього багатоканально-го пристрою нам будуть потрібні кілька спеціальних блоків. Розглянемо спочатку їх.

Блок

DISPLACE  $A, B, C, D$

призначений для відшукування в моделі потрібного транзакта і переміщення його до вказаного блока. Операнд  $A$  — номер транзакта, який потрібно перемістити. Транзакти нумеруються послідовними натуральними числами, номер активного транзакта зберігається в системному числовому атрибуті  $XN1$ . Операнд  $B$  — мітка або номер блока, до якого передається транзакт, вказаний операндом  $A$ . Операнд  $C$  — номер чи ім'я параметра цього транзакта, до якого записується час, що залишився до завершення його обслуговування у блоці **ADVANCE**, якщо транзакт перебував у ньому. Блок **DISPLACE** спрацьовує при потраплянні до нього активного транзакта. Операнд  $D$  може містити мітку альтернативного блока для переходу транзакта — ініціатора виведення, якщо шуканий транзакт відсутній у моделі. Операнди  $C$  і  $D$  — необов'язкові. При відсутності операнда  $D$  активний транзакт переходить до наступного блока. Переміщення транзакта за межі приладу не звільняє прилад, транзакт продовжує його займати.

Недоступність багатоканального пристрою моделює блок **SUNAVAIL  $A$** , де  $A$  — ім'я багатоканального пристрою. Якщо у пристрої обслуговувались транзакти, то вони продовжуватимуть обслуговуватись, поки не вийде час. Інші транзакти в недоступний пристрій зайти не можуть. Зняття недоступності цього пристрою моделюється блоком **SAVAIL  $A$** .

Блоки **FUNAVAIL  $A$**  і **FAVAIL  $A$**  призначені відповідно для моделювання недоступності і доступності одноканального пристрою.

Імітацію виходу з ладу всього багатоканального пристрою здійснюють у ручному режимі з допомогою блоків **SUNAVAIL**, **SAVAIL** і **DISPLACE**.

**Приклад 2.** Вхідний потік вимог обслуговується у багато-канальному пристрої. Пристрій може вийти з ладу повністю. У випадку виходу з ладу багатоканального пристрою всі недо-обслужені вимоги ставляться у чергу для подальшого обслу-говування у багатоканальному пристрої після його ремонту. Ємність пристрою — 10 каналів. Вхідний потік вимог є стаціо-нарним пуассонівським з середнім значенням 2. Час обслуго-вування — випадкова величина з логнормальним законом роз-поділу з середнім значенням 1 і середньоквадратичним відхи-ленням 1 відповідного нормального закону розподілу. Вихід з ладу багатоканального пристрою відбувається за логнормаль-ним законом розподілу з середнім значенням 6 і середньква-дратичним відхиленням 2 відповідного нормального закону розподілу. Час ремонту багатоканального пристрою — випад-кова величина, рівномірно розподілена на проміжку [100, 200]. Виконати моделювання роботи пристрою.

**Розв'язання.** Номери транзактів, які знаходяться в бага-токанальному пристрої, будемо зберігати в комірках пам'яті з номерами від 1 до 10. Для доступу до них використаємо непряму адресацію в циклі. Нижче наводиться можливий лі-стинг програми.

```

bkr STORAGE 10
tab TABLE M1,10,10,100
GENERATE (Exponential(10,0,2))
ASSIGN 1,(Lognormal(11,0,1,1))
mit1 QUEUE cherga
ENTER bkr,1
DEPART cherga
ASSIGN zmc,10
mitc1 TEST E X*zmc,0,mitac1
SAVEVALUE P$zmc,XN1
TRANSFER ,mitvc1
mitac1 LOOP zmc,mitc1
mitvc1 ADVANCE P1

```

```
ASSIGN zmc,10
mitc2 TEST E X*zmc,XN1,mitac2
SAVEVALUE P$zmc,0
TRANSFER ,mitvc2
mitac2 LOOP zmc,mitc2
mitvc2 LEAVE bkp,1
TABULATE tab
TERMINATE 1
GENERATE (Lognormal(12,200,6,2))
SUNAVAIL bkp
ASSIGN zmc,10
mitc3 TEST NE X*zmc,0,mitac3
DISPLACE X*zmc,mit2,1
SAVEVALUE P$zmc,0
mitac3 LOOP zmc,mitc3
ADVANCE 150,50
SAVAIL bkp
TERMINATE 1
mit2 LEAVE bkp,1
TRANSFER ,mit1
START 1000
```

Читачам пропонується самостійно виконати моделювання, користуючись наведеною програмою, проаналізувати статистичний звіт і побудувати графік черги. ■

### § 7.7. Списки користувача

При використанні одноканальних і багатоканальних пристроїв у моделі автоматично можуть виникати черги, які функціонують за принципом FIFO («першим прийшов — першим обслуговують») з врахуванням пріоритетів. Для створення черг з іншою дисципліною обслуговування потрібно використовувати списки користувача. Крім того, черги, які формуються з допомогою списків користувача, у складних моделях працюють швидше.

Список користувача є деяким буфером, до якого можуть тимчасово заноситись транзакти, причому вони вводяться до списку користувача і виводяться з нього відповідно до логіки моделі з допомогою спеціальних блоків.

Для введення транзактів до списку користувача слугує блок

LINK *A, B*.

Операнд *A* задає ім'я або номер списку користувача, до якого заноситься транзакт після входу в блок LINK. Операнд *B* визначає порядок перебування транзактів у списку користувача. Допустимі значення:

FIFO — транзакт передається в кінець списку;

LIFO — транзакт передається на початок списку;

$R\alpha$  — транзакти розташовуються в порядку зростання значення параметра  $\alpha$  транзактів;

довільний системний числовий атрибут, крім  $R\alpha$ , — транзакти розташовуються в порядку спадання значення цього системного числового атрибута.

Найчастіше використовують на місці операнда *B* наступні системні числові атрибути:

PR — транзакти впорядковуються за спаданням пріоритету;

M1 — транзакти впорядковуються за спаданням часу перебування їх у моделі;

$R\alpha$  — транзакти впорядковуються за зростанням параметра  $\alpha$  транзактів.

Наприклад, блок

LINK 3, LIFO

заносить транзакти в початок списку користувача з номером 3, а блок

LINK buf, P\$br

заносить транзакти в список користувача buf, впорядковуючи їх за зростанням значення параметра з ім'ям br.

Умови, внаслідок виконання яких транзакт заноситься до списку користувача, перевіряються засобами, передбаченими розробником моделі. Наприклад:

```
GATE NU obr,wait
SEIZE obr
...
wait LINK buf,LIFO
```

Тут, якщо одноканальний пристрій з ім'ям `obr` зайнятий, то блок `GATE` не допускає транзакт до блока `SEIZE`, а скеровує його на мітку `wait` до блока `LINK`, і транзакт заноситься на початок списку користувача з ім'ям `buf`.

Кожен список користувача має такі системні числові атрибути:

$CH\alpha$  — поточна довжина списку користувача  $\alpha$ ;

$CA\alpha$  — середня довжина списку  $\alpha$ ;

$CM\alpha$  — максимальна довжина списку  $\alpha$ ;

$CC\alpha$  — загальна кількість транзактів, які потрапили до списку  $\alpha$ ;

$CT\alpha$  — середній час перебування транзактів у списку  $\alpha$ .

У попередньому прикладі припускалось, що список користувача необмежений. Список користувача можна використовувати також для моделювання, наприклад, вхідного накопичувача з обмеженою ємністю. Це можна реалізувати так:

```
GATE NU obr,wait
SEIZE obr
...
wait TEST L CH$buf,5,vidm
LINK buf,LIFO
...
vidm TERMINATE 0
```

У цьому прикладі довжина черги обмежується п'ятьма позиціями і транзакти, які надходять при заповненій черзі, втрачатимуться.

Для виведення одного чи кількох транзактів зі списку користувача використовується блок

UNLINK  $X$   $A, B, C, D, E, F$ .

Обов'язковими тут є лише операнди  $A$  і  $B$ . Операнд  $A$  задає ім'я або номер списку користувача. Операнд  $B$  — мітка блока, до якого переходять виведені зі списку користувача транзакти. Операнд  $C$  вказує кількість транзактів, які виводяться, або ключове слово ALL для виведення всіх транзактів, що є в списку. Якщо не використовується операнд  $C$ , то виводяться всі транзакти.

Якщо операнд  $D$  відсутній, то не використовуються також операнди  $X$  і  $E$ . Операнд  $D$  може бути: логічною змінною, ключовим словом BACK, ім'ям чи номером параметра транзакта.

Якщо операнд  $D$  є логічною змінною, то операнди  $X$  і  $E$  не використовуються. Логічна змінна обчислюється відносно транзакта зі списку користувача. Якщо результат її обчислення не є нулем, тобто умова виведення виконується, то транзакт виводиться. Кількість транзактів, які виводяться, визначає операнд  $C$ . Але насправді може бути виведено менше транзактів, ніж вказано операндом  $C$ : їхня кількість дорівнює кількості результатів обчислення логічної змінної, які не дорівнюють нулю. Крім того, транзактів у списку користувача може бути менше, ніж зазначено операндом  $C$ .

Якщо операндом  $D$  є ключове слово BACK, то операнди  $X$  і  $E$  також не використовуються, а транзакти виводяться не з початку, а з кінця списку, в кількості, яка визначається операндом  $C$ .

Як умовний операнд  $X$  використовуються ті самі значення, що й у випадку блока TEST: E, NE, L, LE, G, GE (§ 6.11). Якщо операнд  $X$  пропущений, то вважається, що він рівний  $E$ . Якщо операнд  $D$  є ім'ям чи номером параметра транзакта і заданий операнд  $E$ , то операнди  $D$  і  $E$  порівнюються відпо-

відно до умовного операнда  $X$ . Якщо умова виконується, то транзакт виводиться зі списку користувача. Операнд  $D$  обчислюється відносно транзакта зі списку користувача.

Якщо операнд  $D$  задає параметр, а операнд  $E$  не використовується, то значення параметра транзакта зі списку користувача порівнюється зі значенням такого самого параметра транзакта, який ініціює виведення. Якщо вони рівні, то транзакт виводиться зі списку користувача.

Операндом  $F$  вказується мітка блока, куди переходить транзакт, який ініціює виведення, якщо зі списку користувача не було виведено жодного транзакта.

**Приклад.** В одноканальній системі масового обслуговування організувати таку дисципліну обслуговування, коли пріоритет віддається замовленням з найменшим часом обслуговування. Вхідний потік — стаціонарний пуассонівський з середнім часом 1. Час обслуговування — рівномірно розподілена випадкова величина на проміжку  $[0,5; 2]$ .

**Розв'язання.** Для моделювання черги використаємо список користувача. Нижче наводиться модель:

```
GENERATE (Exponential(5,0,1))
ASSIGN 1,(Uniform(6,0.5,2))
GATE NU pry1,wait
mit SEIZE pry1
ADVANCE P1
RELEASE pry1
UNLINK cherga,mit,1
TERMINATE 1
wait LINK cherga,P1
START 1000
```



При використанні списків користувача немає потреби вводити блоки для збору статистики про чергу, бо всю необхідну статистику можна отримати у стандартному звіті з інформа-

ції про списки користувача. Зокрема, для останнього прикладу маємо:

USER CHAIN	SIZE	RETRY	AVE.CONT	ENTRIES	MAX	AVE.TIME
CHERGA	146	0	80.151	1145	147	80.754

**USER CHAIN** — ім'я або номер списку користувача;

**SIZE** — кількість транзактів у списку користувача в кінці часу моделювання;

**RETRY** — кількість транзактів, які чекають виконання спеціальної умови, що залежить від стану цього списку користувача;

**AVE.CONT.** — середня довжина списку користувача протягом часу моделювання;

**ENTRIES** — загальна кількість транзактів, які потрапили до списку користувача протягом часу моделювання;

**MAX** — максимальна довжина списку користувача протягом часу моделювання;

**AVE.TIME** — середній час перебування транзакта у списку користувача.

## § 7.8. Інтегрування звичайних диференціальних рівнянь

Мова GPSS World дозволяє поєднувати імітаційне й аналітичне моделювання. Це поєднання полягає у тому, що паралельно з процесом руху транзактів через модель може відбуватись процес інтегрування звичайних диференціальних рівнянь, причому обидва процеси дозволяють впливати один на одного.

Для інтегрування звичайних диференціальних рівнянь використовують *змінні користувача*, які створюють командами опису EQU. Формат опису змінної користувача є наступним:

$N$  EQU  $A$ ,

де  $N$  — це ім'я змінної користувача, а  $A$  — її значення. Приклад:

`abcd EQU 2.31` — оголошується змінна користувача з ім'ям `abcd` і значенням `2,31`.

Значення змінних користувача неможливо змінювати з допомогою блоків, їх можна використовувати у блоках лише як константи. Однак значення змінних користувача можна змінювати з допомогою PLUS-процедур (§ 7.9).

У GPSS World передбачена можливість наближеного числового відшукування розв'язків початкових задач для звичайних автономних диференціальних рівнянь першого порядку та їхніх систем. Незалежна змінна у цих рівняннях має бути часом. Для інтегрування таких рівнянь використовуються команди опису `INTEGRATE` і `EQU`. Формат команди `INTEGRATE`:

$$N \text{ INTEGRATE } A, B, C, D, E,$$

де  $N$  — шукана функція  $y(x)$  у рівнянні  $y' = f(y)$ , за її ім'я беруть оголошену вище змінну користувача; операнд  $A$  — права частина диференціального рівняння, тобто значення похідної змінної користувача за часом;  $B$  — перше порогове значення шуканої функції;  $C$  — мітка або номер блока, в який переходить згенерований транзакт у момент, коли шукана функція проходить через своє перше порогове значення у будь-якому напрямку;  $D$  — друге порогове значення шуканої функції;  $E$  — мітка (номер) блока, в який переходить згенерований транзакт у момент, коли шукана функція проходить через своє друге порогове значення у будь-якому напрямку. Операнди  $B$ ,  $C$ ,  $D$  і  $E$  не є обов'язковими.

Початкова умова задається оголошенням початкового значення шуканої функції командою `EQU`. При цьому початкова умова задається в точці 0 (у початковий момент часу). Початкову умову в більш пізній момент часу можна задати лише з використанням PLUS-процедури (див. § 7.9).

Для інтегрування системи диференціальних рівнянь потрібно використовувати кілька команд INTEGRATE і кілька команд EQU для відповідних початкових умов. Звичайне диференціальне рівняння  $n$ -го порядку, розв'язане відносно старшої похідної, зводиться до системи диференціальних рівнянь першого порядку наступним чином:

$$y^{(n)} = F(x, y, y', \dots, y^{(n-1)}) \Rightarrow \begin{cases} y_1' = y_2, \\ y_2' = y_3, \\ \dots \\ y_{n-1}' = y_n, \\ y_n' = F(x, y_1, y_2, \dots, y_n). \end{cases}$$

Неавтономні рівняння і системи зводяться до автономних систем збільшенням кількості функцій і рівнянь, наприклад:

$$y' = f(x, y) \Rightarrow \begin{cases} y_1' = 1, \\ y_2' = f(y_1, y_2). \end{cases}$$

Розв'язок початкової задачі знаходиться числовим методом Рунге–Кутти–Фельберга п'ятого порядку. У кожен момент модельного часу змінна користувача, відповідна шуканій функції, містить числовий розв'язок початкової задачі в цей момент часу. Знайдене значення розв'язку можна використовувати у блоках і в обчисленнях. Після завершення моделювання у стандартному звіті вказується значення, набуте змінною користувача в кінці моделювання.

**Приклад 1.** У чан, що в початковий момент часу містить 10 л чистої води, неперервно надходить зі швидкістю 2 л за хвилину розчин, у кожному літрі якого міститься 0,3 кг солі. Цей розчин переміщується з водою і суміш витікає з чана з тією самою швидкістю. Скільки солі буде в чані через 10 хвилин?

**Розв'язання.** Нехай  $y(t)$  — кількість солі в чані в момент часу  $t$ , час  $t$  вимірюється у хвилинах, кількість солі  $y(t)$  — у

кілограмах. Тоді

$$y(t + \Delta t) - y(t) = \left( 2 \cdot 0,3 - 2 \cdot \frac{y(t + \alpha)}{10} \right) \Delta t$$

— приріст солі за час  $\Delta t$  ( $0 < \alpha < \Delta t$ ). Поділивши на  $\Delta t$ , спрямувавши  $\Delta t$  до нуля і врахувавши, що  $\alpha \rightarrow 0$  для  $\Delta t \rightarrow 0$ , отримаємо:

$$y' = 0,6 - 0,2y.$$

Крім того, маємо  $y(0) = 0$ . Для моделювання придатна модель:

```
y_ EQU 0
y_ INTEGRATE (0.6-0.2#y_)
GENERATE 10
TERMINATE 1
START 1
```

Круглі дужки у другому рядку цієї програми обов'язкові. У звіті вказується значення  $y_$  на кінець моделювання, тобто для часу  $t = 10$  (рис. 7.10). Оскільки воно дорівнює 2,594, то через 10 хвилин у чані буде 2,594 кг солі. ■

NAME	VALUE
$y_$	2.594

Рис. 7.10

**Приклад 2.** Знайти час, коли в чані з попередньої задачі опиниться 2,9 кг солі.

**Розв'язання.** Модифікуємо модель:

```
y_ EQU 0
y_ INTEGRATE (0.6-0.2#y_),2.9,mit
GENERATE 1000
mit TERMINATE 1
START 1
```

Час моделювання є відповіддю на поставлену задачу. З фрагменту звіту (рис. 7.11) видно, що час моделювання становить 17,012 хвилин. ■

START TIME	END TIME	BLOCKS	FACILITIES	STORAGES
0.000	17.012	2	0	0

Рис. 7.11

**Приклад 3.** Знайти розв’язок початкової задачі

$$\begin{cases} \frac{dx}{dt} = -xy, \\ \frac{dy}{dt} = 2x - 6y \end{cases} \quad x(0) = 1, \quad y(0) = 2$$

в точці  $t = 5$ .

**Розв’язання.** Складаємо модель:

```
x_ EQU 1
y_ EQU 2
x_ INTEGRATE (-x_#y_)
y_ INTEGRATE (2#x_-6#y_)
GENERATE 5
TERMINATE 1
START 1
```

Зі звіту можна побачити, що  $x(5) = 0,326$ ,  $y(5) = 0,111$ . ■

Для припинення чи продовження інтегрування диференціального рівняння використовують блок `INTEGRATION A,B`, де  $A$  — змінна користувача, описана командою `INTEGRATE`, а  $B$  є ключовим словом `OFF` чи `ON`. Значення `OFF` означає «припинити інтегрування», а значення `ON` — «відновити інтегрування». Припинення інтегрування приводить до того, що змінна користувача, описана командою `INTEGRATE`, буде містити розв’язок рівняння, відповідний часу входу будь-якого транзакта в блок `INTEGRATION` з цією змінною і словом `OFF`, і в майбутні моменти часу.

## § 7.9. PLUS-процедури

У GPSS World передбачена можливість створення PLUS-процедур, які можна використовувати в арифметичних виразах у блоках як, наприклад, стандартні процедури — `Normal`

(генерування нормально розподіленої випадкової величини) чи `sin` (синус числа). Оскільки ці процедури вертають значення, їх краще було б, по аналогії з багатьма мовами програмування, називати функціями.

Описується PLUS-процедура ключовим словом **PROCEDURE**, після якого ставиться ім'я створюваної процедури (послідовність букв, цифр, символів підкреслення), а далі у круглих дужках задаються формальні параметри. Якщо параметрів не потрібно, то круглі дужки все одно треба ставити.

Після заголовку процедури розміщується її тіло, яке обмежується ключовими словами **BEGIN** і **END**, причому після **END** ставиться крапка з комою (;). Замість великих букв скрізь можна вживати маленькі. Чітких обмежень на структуру рядка в процедурі немає, як і в багатьох мовах програмування. В рядку може бути і кілька операторів. Кожен оператор у тілі процедури закінчується крапкою з комою. Коментарів всередині процедури не повинно бути.

В тілі процедури можна використовувати змінні користувача, параметри процедури і системні числові атрибути. Крім того, можна створити локальні змінні процедури командою **TEMPORARY**, після якої всі ці змінні перераховуються через кому. В кінці їх, як після кожного оператора, ставиться крапка з комою. Команда **TEMPORARY** розміщується в тілі процедури (після ключового слова **BEGIN**). Локальна змінна, як і вузьке ім'я системного числового атрибута та змінна користувача, — це послідовність букв, цифр і символів підкреслення, перший символ імені — буква. Ім'я локальної змінної має відрізнятися від ключових слів і класів системних числових атрибутів. Приклад:

```
TEMPORARY z, z2, ac1_;
```

Командою **TEMPORARY MATRIX** можна оголошувати локальні матриці. Це робиться аналогічно до локальних змінних, але у квадратних дужках після імені кожної матриці через кому перелічується кількість елементів матриці за кожною розмір-

ністю, наприклад:

```
TEMPORARY MATRIX abcd[3,3,5];
```

Оператор присвоювання записується у формі:

```
змінна=вираз;
```

Присвоювати значення можна лише локальним змінним процедури, змінним користувача та елементам матриць. Системним числовим атрибутам, зокрема, коміркам пам'яті чи параметрам транзакта, присвоювати значення не можна. Системні числові атрибути можуть використовуватись тільки у правій частині оператора присвоювання. Вирази будуються за правилами побудови арифметичних виразів у мові GPSS World, зокрема знак «#» позначає множення. У круглі дужки брати вирази у даному випадку необов'язково. У виразах можна викликати інші процедури і функції, наприклад:

```
z=ABS(X3+FN$fbck)-2;
```

Для присвоєння значень елементам матриць у лівій (як і в правій) частині оператора присвоювання записується лише вузьке ім'я матриці (без класу системного числового атрибута *MX*), а елементи матриці вказуються у квадратних дужках через кому, наприклад:

```
abcd[2,3,4]=3+abcd[1,1,2];
```

Оператор розгалуження будується так:

```
IF (логічний вираз) THEN оператор;
```

або

```
IF (логічний вираз) THEN оператор; ELSE оператор;
```

Логічний вираз будується за правилами побудови логічних виразів у мові GPSS World і береться у круглі дужки. Замість одного оператора після ключового слова *THEN* чи *ELSE* між ключовими словами *BEGIN* і *END* можна записувати послідовність операторів, кожен з яких закінчується крапкою з комою, наприклад:

```
IF (z<0) THEN BEGIN z=0; z2=1; END; ELSE z2=0;
```

Оператор циклу має вигляд:

```
WHILE (логічний вираз) DO оператор;
```

Тут також замість одного оператора може бути кілька операторів між ключовими словами BEGIN і END. Приклад:

```
WHILE ((z<10) & (X3=1)) DO  
BEGIN z2=z2+RN10; z=z+1; END;
```

Оператор переходу на мітку всередині процедури записується у вигляді:

```
GOTO мітка;
```

Мітка — це послідовність букв і цифр, що починається з букви. Мітка відокремлюється від оператора, на який потрібно перейти, символом двокрапки. Приклад:

```
GOTO sob1;  
...  
sob1:t=12;
```

Оператор повернення значення має вигляд:

```
RETURN вираз;
```

Обчислене значення виразу вертається як результат роботи процедури, всі оператори після оператора RETURN ігноруються, відбувається вихід з процедури. Якщо написати просто

```
RETURN;
```

то результатом буде 0. Якщо оператор RETURN відсутній, то процедура вертає останнє обчислене значення.

Виклик PLUS-процедури використовують для обчислення виразу у потрібному блоці. При цьому ім'я процедури разом з її фактичними параметрами у круглих дужках трактується як арифметичний вираз, який береться в круглі дужки. Якщо важливим є не саме значення, яке обчислює процедура, а якісь інші «побічні» результати її роботи, то результат роботи процедури можна присвоїти деякому параметру транзакта чи комірці пам'яті, які потім не використовуватимуться.

Існує також спеціальний блок PLUS  $A$ , який викликає процедуру, що ставиться у круглих дужках замість операнда  $A$ , але ігнорує значення, яке вона вертає.

**Приклад 1.** Знайти розв'язок початкової задачі  $y' = 0,6 - 0,2y$ ,  $y(1) = 0,5$  в точці  $t = 10$ .

**Розв'язання.** Для розв'язування цієї задачі засобами GPSS World потрібно скласти PLUS-процедуру, яка б змінювала початкове значення функції  $y$ . Будемо модель:

```

у_ EQU 0
у_ INTEGRATE (0.6-0.2#у_)
PROCEDURE newint(valuei)
BEGIN
у_=valuei;
END;
GENERATE 1,,1
PLUS (newint(0.5))
ADVANCE 9
TERMINATE 1
START 1

```

У момент часу, рівний одиниці, генерується один транзакт, який запускає PLUS-процедуру `newint` зміни значення змінної користувача  $у_$ . Процес інтегрування диференціального рівняння продовжується до часу  $t = 10$ , коли знищується транзакт і припиняється моделювання. Зі звіту можна побачити, що розв'язок початкової задачі в точці  $t = 10$  дорівнює 2,587. ■

**Приклад 2.** Потік надходження замовлень у систему масового обслуговування є стаціонарним пуассонівським з середнім значенням 2. Час обслуговування замовлення в одноканальному пристрої має логнормальний розподіл з середнім значенням 0,4 і середньоквадратичним відхиленням 0,3 відповідного нормального розподілу. Виконати моделювання об-

слуговування 2000 замовлень і визначити мінімальну, максимальну і середню тривалості перебування замовлень у системі, а також медіану і показник асиметрії для цих тривалостей.

**Розв'язання.** Медіану і показник асиметрії неможливо знайти зі стандартного статистичного звіту, їхнє обчислення потрібно запрограмувати. Нехай  $x_1, x_2, \dots, x_n$  — тривалості перебування замовлень у системі. Середнє арифметичне значення  $\bar{x}$  і показник асиметрії  $\nu$  обчислюють за формулами:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i, \quad \nu = \frac{1}{n \cdot D^{3/2}} \sum_{i=1}^n (x_i - \bar{x})^3, \quad D = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2.$$

Для відшукування медіани відсортуємо тривалості  $x_i$  за зростанням і перепозначимо їх через  $x_{(i)}$ :

$$x_{(1)} \leq x_{(2)} \leq \dots \leq x_{(n)}.$$

Тоді медіану  $x_{0,5}$  знаходять за формулою:

$$x_{0,5} = \begin{cases} x_{(\frac{n+1}{2})}, & n - \text{непарне,} \\ \frac{x_{(\frac{n}{2})} + x_{(\frac{n}{2}+1)}}{2}, & n - \text{парне.} \end{cases}$$

Для зберігання тривалостей перебування замовлень у системі зручно використовувати матрицю, а для обчислення медіани, показника асиметрії, мінімального, максимального і середнього арифметичного значень доцільно створити PLUS-процедуру. Модель на мові GPSS World може бути, наприклад, такою:

```

minim EQU 1000000
maxim EQU 0
x_ EQU 0
x_05 EQU 0
nuas EQU 0
zam EQU 2000

```

```
trans MATRIX ,2000,1
PROCEDURE calc()
BEGIN TEMPORARY ii,j,sw,d;
  ii=1;
  WHILE (ii<=zam) DO
    BEGIN
      x_=x_+trans[ii,1];
      IF (trans[ii,1]<minim) THEN minim=trans[ii,1];
      IF (trans[ii,1]>maxim) THEN maxim=trans[ii,1];
      ii=ii+1;
    END;
  x_=x_/zam; ii=1; d=0;
  WHILE (ii<=zam) DO
    BEGIN
      nuas=nuas+(trans[ii,1]-x_)#(trans[ii,1]-x_)#
        (trans[ii,1]-x_);
      d=d+(trans[ii,1]-x_)#(trans[ii,1]-x_);
      ii=ii+1;
    END;
  d=d/(zam-1); nuas=nuas/(zam#sqr(d#d#d)); ii=1;
  WHILE (ii<=zam-1) DO
    BEGIN
      j=ii+1;
      WHILE (j<=zam) DO
        BEGIN
          IF (trans[ii,1]>trans[j,1]) THEN
            BEGIN
              sw=trans[ii,1]; trans[ii,1]=trans[j,1];
              trans[j,1]=sw;
            END;
          j=j+1;
        END;
      ii=ii+1;
    END;
```

```

IF (2#int(zam/2)=zam) THEN
  x_05=(trans[zam/2,1]+trans[zam/2+1,1])/2;
ELSE x_05=trans[(zam+1)/2,1];
END;
GENERATE (Exponential(5,0,2))
QUEUE wait
SEIZE okp
DEPART wait
ADVANCE (Lognormal(6,0,0.4,0.3))
RELEASE okp
SAVEVALUE 1+,1
MSAVEVALUE trans,X1,1,M1
TEST E X1,zam,term
PLUS (calc())
term TERMINATE 1
START 2000

```

Ця модель і її PLUS-процедура `calc` розраховані на до-  
вільну кількість замовлень, збережену в змінній користувача  
`zam` і задану в команді опису матриці `trans` та команді керу-  
вання `START`. Для пришвидшення моделювання читач може  
запрограмувати швидке сортування. У наступній таблиці на-  
ведені змінні користувача, в яких PLUS-процедура `calc` збе-  
рігає знайдені значення, та самі значення, отримані зі звіту.

Назва	Змінна	Значення
Мінімум	<code>minim</code>	0,595
Максимум	<code>maxim</code>	17,759
Середнє	<code>x_</code>	4,775
Медіана	<code>x_05</code>	3,739
Асиметрія	<code>nuas</code>	1,273

Інші приклади застосування PLUS-процедур наведені в  
§ 9.19 і § 9.21.

### § 7.10. Опитування неактивних транзактів

Вище розглядалися методи опитування лише активних транзактів, які потрапили у певний блок. Для доступу до них (у тому числі у PLUS-процедурах) використовуються системні числові атрибути транзактів (§ 7.3). Іноді може виникнути потреба аналізу параметрів чи інших характеристик неактивних транзактів. Існує кілька спеціальних функцій, які дозволяють це робити. Вони наведені у наступній таблиці.

Функція	Опис
<code>QueryXNExist(TN)</code>	вертає 1, якщо транзакт з номером $TN$ існує в моделі, інакше — 0
<code>QueryXNParameter(TN,Par)</code>	визначає значення параметра $Par$ транзакта з номером $TN$
<code>QueryXNPriority(TN)</code>	визначає пріоритет транзакта з номером $TN$
<code>QueryXNM1(TN)</code>	визначає час перебування в моделі транзакта з номером $TN$
<code>QueryXNAssemblySet(TN)</code>	визначає кількість транзактів у групі, в яку входить транзакт з номером $TN$

У всіх цих функціях, крім першої, виникне помилка часу виконання, якщо транзакта з заданим номером не існує в моделі. Отже, необхідно спочатку робити перевірку факту існування транзакта з заданим номером.

### § 7.11. Виконання блока

При потраплянні довільного транзакта в блок

EXECUTE A

цей транзакт виконує блок з міткою або номером  $A$ . Після цього зазвичай транзакт переходить у наступний блок після блока EXECUTE. Винятком є випадок, коли блок з міткою або номером  $A$  змінює маршрут руху транзакта або знищує його. Приклад: EXECUTE mit — транзакт виконує блок з міткою mit у програмі.

### Питання до розділу 7

1. Як створювати копії транзактів? Як об'єднувати транзакти? Які блоки використовують для синхронізації руху транзактів?
2. Як створювати матриці і працювати з ними у GPSS World?
3. Назвіть основні системні числові атрибути процесу моделювання, транзактів, блоків, пристроїв, черг, таблиць, комірок пам'яті, змінних, функцій і логічних перемикачів.
4. Які команди використовують для керування процесом моделювання у GPSS World? Опишіть їх.
5. Які інформаційні вікна можна використовувати для спостереження за процесом і результатами моделювання у середовищі GPSS World? Як будувати графіки у GPSS World?
6. Як імітувати вихід з ладу одноканального пристрою або одного каналу багатоканального пристрою?
7. Які блоки використовують для імітації виходу з ладу всього багатоканального пристрою?
8. Як створювати спеціальні черги у GPSS?
9. Як інтегрувати звичайні диференціальні рівняння засобами мови GPSS World?
10. Назвіть особливості роботи з PLUS-процедурами. Як задаються оператори розгалуження, циклу, присвоєння, переходу на мітку у PLUS-процедурах?
11. Як визначати характеристики неактивних транзактів?

### Вправи до розділу 7

1. Вхідний потік вимог задається неперервною функцією зі значеннями у вузлах  $(0; 0)$ ,  $(0,2; 1)$ ,  $(0,5; 2)$ ,  $(0,7; 3)$ ,  $(0,8; 4)$ ,  $(1; 7)$ . Час обслуговування є рівномірно розподіленою на відрізку  $[1, 5]$  випадковою величиною. Одночасно не може обслуговуватись більше однієї вимоги. На мові GPSS World напишіть модель, яка заповнить

матрицю за наступним правилом: у перший рядок  $j$ -го стовпця матриці буде заноситись час появи  $j$ -ї вимоги в моделі, а в другий рядок  $j$ -го стовпця — час перебування цієї вимоги в моделі. Здійсніть моделювання обслуговування 100 вимог. Перегляньте створену матрицю.

2. Вхідний потік вимог має логнормальний закон розподілу з середнім значенням 2 і середньоквадратичним відхиленням 1 відповідного нормального розподілу. Час обслуговування вимоги в одноканальному пристрої є випадковою величиною, розподіленою за експоненціальним законом з інтенсивністю 0,5. Вихід з ладу пристрою відбувається через інтервали часу, що мають розподіл Вейбулла з параметрами стиснення  $\beta = 600$  і форми  $\alpha = 10$  (задаються саме в цьому порядку у функції Weibull мови GPSS World). Час ремонту є рівномірно розподіленою випадковою величиною на проміжку  $[10, 100]$ . При виході з ладу пристрою недообслужені вимоги чекають завершення обслуговування після закінчення ремонту. За наявності черги першими мають обслуговуватись вимоги, обслуговування яких було перерване виходом з ладу пристрою.

Складіть модель на мові GPSS World для моделювання роботи пристрою і його ремонту. Здійсніть моделювання обслуговування 1000 вимог. Визначте середню і максимальну довжини черги. Проаналізуйте таблицю часу перебування вимог у моделі.

Створіть точку зупинки моделювання при виході з ладу пристрою. Визначте системний час, коли відбулась ця подія. Перегляньте списки поточних і майбутніх подій після досягнення цієї точки. Після цього по крокам виконайте моделювання і проаналізуйте модель у вікні блоків.

Створіть і перегляньте графік залежності поточної довжини черги від часу.

3. Вхідний потік вимог є стаціонарним пуассонівським з середнім значенням 0,3. Час обслуговування вимоги в одному каналі чотиріканального пристрою є випадковою величиною з логнормальним законом розподілу з середнім значенням 0,1 і середньоквадратичним відхиленням 0,5 відповідного нормального розподілу. Вихід з ладу одного з каналів пристрою відбувається через інтервали часу, що мають логнормальний розподіл з середнім значенням 6 і середньоквадратичним відхиленням 2 відповідного нормального розподілу. Час ремонту є рівномірно розподіленою випадковою ве-

личною на проміжку [4, 30]. При виході з ладу каналу пристрою недообслужені вимоги залишають систему.

Складіть модель на мові GPSS World для моделювання роботи пристрою і його ремонту. Здійсніть моделювання обслуговування 10000 вимог. Визначте середню і максимальну довжини черги. Проаналізуйте таблицю часу перебування вимог у моделі.

Під час моделювання перегляньте зміну інформації у вікнах блоків, пристроїв, черг, таблиці. У вікні виразів під час моделювання перегляньте зміну абсолютного системного часу, тривалості обслуговування вимоги, поточної довжини черги.

Створіть точку зупинки моделювання при виході з ладу пристрою. Визначте системний час, коли відбулась ця подія. Перегляньте списки поточних і майбутніх подій після досягнення цієї точки. Після цього по крокам виконайте моделювання і проаналізуйте модель у вікні блоків.

Перегляньте графік залежності поточної довжини черги від часу.

4. Вхідний потік вимог є стаціонарним пуассонівським з середнім значенням 5. Час обслуговування вимоги в одному каналі дванадцятиканального пристрою є випадковою величиною з логнормальним законом розподілу з середнім значенням 2 і середньоквадратичним відхиленням 2 відповідного нормального розподілу. Вихід з ладу всього пристрою відбувається через інтервали часу, що мають логнормальний розподіл з середнім значенням 5 і середньоквадратичним відхиленням 3 відповідного нормального розподілу. Час ремонту є рівномірно розподіленою величиною на проміжку [50, 300]. При виході з ладу пристрою недообслужені вимоги чекають завершення ремонту. За наявності черги першими мають обслуговуватись вимоги, обслуговування яких було перерване виходом з ладу пристрою.

Складіть модель на мові GPSS World для моделювання роботи пристрою і його ремонту. Здійсніть моделювання обслуговування 1000 вимог. Визначте середню і максимальну довжини черги. Проаналізуйте таблицю часу перебування вимог у моделі.

Під час моделювання перегляньте зміну інформації у вікнах блоків, пристроїв, таблиці.

Створіть точку зупинки моделювання при виході з ладу пристрою. Визначте системний час, коли відбулась ця подія. Пере-

гляньте списки поточних і майбутніх подій після досягнення цієї точки. Після цього по крокам виконайте моделювання і проаналізуйте модель у вікні блоків.

Перегляньте графік залежності поточної довжини черги від часу.

5. Вхідний потік вимог є стаціонарним пуассонівським з середнім значенням 5. Час обслуговування вимоги в одноканальному пристрої є випадковою величиною з рівномірним законом розподілу на проміжку [2, 7]. За наявності черги першими мають обслуговуватись вимоги, які прийшли останніми.

Складіть модель на мові GPSS World для моделювання роботи пристрою. Здійсніть моделювання обслуговування 1000 вимог. Визначте середню і максимальну довжини черги, середній час перебування вимог у черзі. Проаналізуйте таблицю часу перебування вимог у моделі.

Під час моделювання перегляньте зміну інформації у вікнах блоків, пристроїв, таблиці. Створіть і перегляньте графік залежності поточної довжини черги від часу.

6. Вхідний потік вимог є стаціонарним пуассонівським з інтенсивністю 1. Час обслуговування вимоги в одноканальному пристрої є випадковою величиною з рівномірним законом розподілу на проміжку [0,3; 1,5]. За наявності черги першими мають обслуговуватись вимоги з найменшим часом обслуговування. Черга обмежена десятьма позиціями. Вимоги, для яких немає місця в черзі, залишають модель необслуженими.

Складіть модель на мові GPSS World для моделювання роботи пристрою. Здійсніть моделювання обслуговування 1000 вимог. Визначте середню і максимальну довжини черги, середній час перебування вимог у черзі. Проаналізуйте таблицю часу перебування вимог у моделі.

Під час моделювання перегляньте зміну інформації у вікнах блоків, пристроїв, таблиці. Створіть і перегляньте графік залежності поточної довжини черги від часу.

7. З допомогою GPSS World знайдіть розв'язок початкової задачі  $y' = 5 + 4\sqrt{y}$ ,  $y(0) = 22$  у точці  $x_1 = 33$ . З допомогою GPSS World знайдіть таке значення  $x_2$ , щоб розв'язок цієї початкової задачі  $y(x_2) = 100$ .

8. Складіть на мові GPSS World модель п'ятиканальної системи

масового обслуговування з відмовами, передбачивши можливість числового інтегрування системи диференціальних рівнянь для ймовірностей станів і порівняння результатів, отриманих з допомогою аналітичної та імітаційної моделей. Інтенсивність стаціонарного пуассонівського потоку замовлень —  $\lambda = 1/3$ , час обслуговування — випадкова величина з експоненціальним законом розподілу й інтенсивністю  $\mu = 1/16$ . Порівняйте результати.

Побудуйте одночасно графіки всіх ймовірностей станів з аналітичної моделі для часу з проміжку  $[0, 100]$ . Побудуйте одночасно графіки аналітичної й імітаційної ймовірностей обслуговування для часу з проміжку  $[0, 100]$ . Побудуйте одночасно графіки аналітичної й імітаційної середніх кількостей зайнятих каналів обслуговування для часу з проміжку  $[0, 100]$ . Побудуйте одночасно графіки аналітичного й імітаційного коефіцієнтів зайнятості системи масового обслуговування для часу з проміжку  $[0, 100]$ .

Вказівка. Система диференціальних рівнянь Колмогорова для ймовірностей станів має вигляд:

$$\begin{cases} p_0'(t) = \mu p_1(t) - \lambda p_0(t), \\ p_1'(t) = \lambda p_0(t) + 2\mu p_2(t) - (\lambda + \mu)p_1(t), \\ p_2'(t) = \lambda p_1(t) + 3\mu p_3(t) - (\lambda + 2\mu)p_2(t), \\ p_3'(t) = \lambda p_2(t) + 4\mu p_4(t) - (\lambda + 3\mu)p_3(t), \\ p_4'(t) = \lambda p_3(t) + 5\mu p_5(t) - (\lambda + 4\mu)p_4(t), \\ p_5'(t) = \lambda p_4(t) - 5\mu p_5(t). \end{cases}$$

9. Вхідний потік вимог є стаціонарним пуассонівським з середнім значенням 2. Час обслуговування в одноканальному пристрої є випадковою величиною з експоненціальним законом розподілу, середнє значення якого задається функцією

$$f(x) = \begin{cases} 3 - x, & x \leq 2, \\ \frac{1}{x-1}, & x > 2, \end{cases}$$

де  $x$  — довжина черги до цього пристрою.

Складіть модель на мові GPSS World для моделювання роботи системи, створивши PLUS-процедуру для обчислення значення функції  $f(x)$ . Здійсніть моделювання обслуговування 1000 вимог. Визначте середню і максимальну довжини черги та побудуйте графік залежності поточної довжини черги від часу.

## Розділ 8. Моделювання випадкових величин

### § 8.1. Поняття про метод Монте-Карло

Метод Монте-Карло називають ще методом статистичних випробувань або ймовірнісним моделюванням. Він є одним з основних методів імітаційного моделювання.

Імітаційною моделлю при використанні методу Монте-Карло є комп'ютерна програма, яка відтворює процес функціонування реальної системи. Моделювання на основі методу Монте-Карло ґрунтується на багаторазовій імітації досліджуваного явища, тобто отриманні великої кількості конкретних реалізацій цього явища. Вплив випадкових факторів на систему моделюється з допомогою випадкових чисел. Отриманий набір реалізацій обробляється потім з метою визначення потрібних характеристик. Конкретний алгоритм розв'язування задачі методом Монте-Карло повністю залежить від постановки задачі.

На практиці метод Монте-Карло застосовують тоді, коли розв'язати задачу цим методом легше, ніж будь-яким іншим. Метод Монте-Карло може застосовуватись для розв'язування задач моделювання систем різних типів — неперервно-детермінованих, дискретно-детермінованих, неперервно-ймовірнісних, дискретно-ймовірнісних. Крім того, цей метод може успішно використовуватись для розв'язування детермінованих задач, безпосередньо не пов'язаних з моделюванням, та задач оптимізації. В останніх випадках велика кількість спроб дозволяє отримати наближене значення точного чи оптимального розв'язку.

Розв'язування задач як моделювання, так і оптимізації методом Монте-Карло ґрунтується на використанні комп'ютерів і, отже, складанні програм. Програми можна складати, використовуючи як універсальні мови програмування високого рівня, такі як Python, C++, Java та інші, так і спеціальні мови імітаційного моделювання. Можна користуватись та-

кож математичними пакетами, такими як Maple, Mathematica і MATLAB. Для імітаційного моделювання систем масового обслуговування зручно застосовувати спеціальні мови і середовища моделювання. Одну з таких мов — мову GPSS World — було розглянуто в розділах 6 і 7. Ця мова при моделюванні автоматично реалізує метод Монте-Карло.

**Приклад 1.** Необхідно знайти площу фігури, обмежену функцією  $y = f(x)$  ( $f(x) > 0$  на відрізку  $[a, b]$ ), вертикальними відрізками  $x = a$  і  $x = b$  та віссю абсцис (рис. 8.1), тобто треба знайти

$$S = \int_a^b f(x) dx.$$

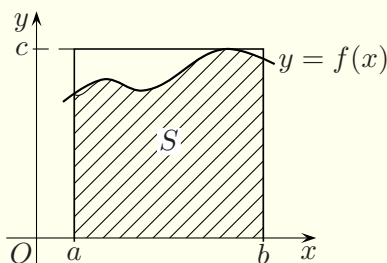


Рис. 8.1.

**Розв'язання.** Зрозуміло, що якщо інтеграл береться в елементарних функціях, то площу можна знайти аналітично. У будь-якому випадку існують методи числового інтегрування, які дозволяють отримати наближене значення площі. Для розв'язування цієї задачі можна також застосувати метод Монте-Карло.

Нехай

$$c = \max_{x \in [a, b]} f(x).$$

Фігура, площу якої шукаємо, міститься в прямокутнику  $[a, b] \times [0, c]$ , площа якого дорівнює  $c(b - a)$ . Нехай  $\xi$  — де-

яка випадкова величина, рівномірно розподілена на проміжку  $[0, 1]$ . Замість відрізка  $[0, 1]$  можна брати півінтервал  $[0, 1)$ , бо ймовірність набуття неперервною випадковою величиною конкретного значення дорівнює нулю. Тоді  $\eta = (b - a)\xi + a$  і  $\zeta = c\xi$  — випадкові величини, рівномірно розподілені на проміжках  $[a, b]$  і  $[0, c]$ . Будемо використовувати випадкову величину  $\eta$  для визначення координати  $x_i$ , а випадкову величину  $\zeta$  для визначення координати  $y_i$ .

Проведемо  $N$  випробувань. Випробування вважатимемо успішним, якщо точка з координатами  $(x_i, y_i)$  потрапить в область під кривою  $y = f(x)$  або на неї. Підрахуємо кількість успішних випробувань, позначимо її через  $m$  і знайдемо величину  $\frac{m}{N}$ . Згідно з геометричним означенням імовірності при великих  $N$  ця величина є ймовірністю  $P$  потрапляння точки в заштриховану область на рис. 8.1.

$$P = \frac{S}{c(b-a)} \quad \Rightarrow \quad S \approx c(b-a) \frac{m}{N}. \quad \blacksquare$$

На практиці цей метод для обчислення одинарних інтегралів не використовують, бо він вимагає великої кількості обчислень. Але його ефективно можна застосовувати для обчислення кратних інтегралів, обмежених складними областями, бо кількість  $N$  випробувань, які потрібно провести для обчислення інтеграла, не залежить від його кратності.

**Приклад 2.** Розглянемо складнішу задачу, яку розв'язати аналітично доволі складно. Нехай є деяка ціль довільної форми загальною площею  $S$ , на яку бомбардувальники скидають  $n$  бомб. Область враження кожної бомби — це круг з радіусом  $r$ . Ціль вважається враженою, якщо зруйновано не менше, ніж  $K$  відсотків її площі  $S$ . Необхідно знайти ймовірність враження цілі, якщо бомбардувальники скидають бомби всередині круга з центром приблизно у центрі цілі і радіусом  $R$  (рис. 8.2).

**Розв'язання.** Зауважимо, що окремі бомби можуть взагалі не влучити в ціль або лише її зачепити, бомби можуть падати так, що області руйнування ними цілі перекриватимуться.

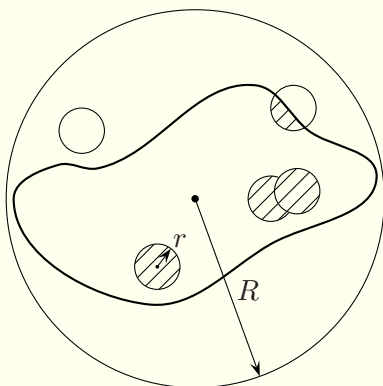


Рис. 8.2.

Проведемо  $N$  випробувань, моделюючи кожного разу координати точок падіння бомб, і підрахуємо кількість випробувань  $m$ , під час яких ціль було вражено. Якщо площа враження становить  $K$  (або більше) відсотків загальної площі цілі  $S$ , то ціль вважається враженою, а випробування — успішним. Імовірність враження цілі визначається як  $\frac{m}{N}$  (згідно з класичним означенням імовірності). ■

## § 8.2. Генератори випадкових чисел

Генератори випадкових чисел, як правило, створюють випадкові числа, рівномірно розподілені на проміжку  $[0, 1)$ . На основі цих чисел можна отримати випадкові числа практично будь-яких розподілів. Треба мати на увазі, що комп'ютер використовує лише раціональні випадкові числа, десяткові дроби яких є неперіодичними, тобто розподіл насправді є майже рівномірним, бо більша частина чисел (ірраціональні і більшість раціональних) пропущені.

Є три групи методів генерування випадкових чисел.

1. Апаратні методи генерування випадкових чисел базуються на використанні деяких фізичних явищ (наприклад, шумів електронних приладів). Вбудовані у комп'ютери апаратні генератори випадкових чисел використовуються в системах захисту інформації. Зокрема, в деякі моделі процесорів інтегровано пристрій PadLock — апаратний генератор випадкових чисел для шифрування інформації. Використання шумів електронних приладів ґрунтується на тому, що якщо на підсилювач не подавати ніякого сигналу й увімкнути його на повну потужність, то буде чути шум, який можна перетворити в послідовність випадкових чисел. Недоліком апаратних методів є неможливість повторного відтворення тієї самої послідовності випадкових чисел, що може бути необхідним на етапі налагодження моделі.

2. Табличні методи. Існують готові таблиці випадкових чисел, які містять мільйони згенерованих апаратними методами випадкових чисел. Такі таблиці можна тримати в оперативній пам'яті комп'ютера, але недоліком цього є використання значних обсягів оперативної пам'яті для зберігання випадкових чисел.

3. Найчастіше використовують програмні генератори випадкових чисел. Їх називають генераторами псевдовипадкових чисел, бо вони формують послідовність випадкових чисел, яка насправді має період, хоч і дуже великий. У більшості генераторів псевдовипадкових чисел використовується рекурентна процедура  $x_{i+1} = f(x_i)$  (наступне значення обчислюється на основі попереднього).

Лише спеціалізовані мови моделювання мають кілька (чи навіть багато) генераторів випадкових чисел. Прикладом є мова імітаційного моделювання GPSS World, розглянута у [шостому](#) і [сьомому](#) розділах цієї книги. Універсальні мови програмування мають лише один генератор випадкових чисел. Використання одного генератора для генерування випад-

кових чисел з різною метою є небажаним, бо може вийти так, що як випадкове число використовуватиметься лише кожне, скажімо, п'яте число. Виникає питання, чи кожне п'яте число буде задовольняти всім критеріям, які мають висуватись до рівномірно розподілених випадкових чисел.

Найпростішим ефективним генератором випадкових чисел є *лінійний конгруентний генератор*, який працює за формулами Лемера

$$x_{i+1} = (ax_i + c) \bmod m, \quad r_{i+1} = \frac{x_{i+1}}{m}, \quad (8.1)$$

де  $r_{i+1}$  — випадкове число,  $i = 0, 1, 2, \dots$ , числа  $a$ ,  $c$  і  $m$  — невід'ємні цілі константи,  $\bmod m$  — остача від ділення на число  $m$ . Цей генератор може створити не більше, ніж  $m$  різних випадкових чисел, а саме:  $0, \frac{1}{m}, \frac{2}{m}, \dots, \frac{m-1}{m}$ . Замість  $m$  для невід'ємних чотирьохбайтових цілих чисел логічно використовувати число  $2^{32}$ , тоді операцію визначення остачі від ділення числа на  $m$  здійснювати не треба (при переповненні розрядної сітки процесора старші цифри втрачаються і залишається лише остача). Число  $m = 2^{32} = 4\,294\,967\,296$  є максимальною можливою довжиною періоду в послідовності випадкових чисел, проте період може виявитись меншим.

**Теорема.** Лінійний конгруентний генератор (8.1) для будь-якого початкового значення  $x_0$  дає послідовність випадкових чисел з періодом  $m$  тоді і тільки тоді, коли:

- 1) числа  $c$  і  $m$  є взаємно простими;
- 2) число  $a - 1$  є кратним  $q_i$ , де  $q_i$  — прості дільники числа  $m$ ;
- 3) число  $a - 1$  є кратним 4, якщо  $m$  є кратним 4.

Ознайомитись з доведенням цієї теореми можна, наприклад, у книзі [17].

### § 8.3. Моделювання випадкових подій

**Незалежні випадкові події.** Припустимо, що ймовірність настання деякої елементарної випадкової події  $A$  в одному випробуванні становить  $P(A) = p$ . Вважається, що умови проведення кожного випробування однакові і його можна повторити нескінченну кількість разів. Якщо  $r$  — значення рівномірно розподіленої на півінтервалі  $[0, 1)$  випадкової величини, то на основі формули  $P(r < p) = p = P(A)$  можна стверджувати, що за умови  $r < p$  відбудеться подія  $A$ , а якщо  $r \geq p$ , то відбудеться протилежна подія  $\bar{A}$  (рис. 8.3).

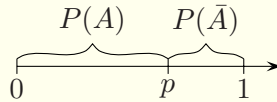


Рис. 8.3.

**Група несумісних подій.** Нехай задано групу несумісних подій  $A_1, A_2, \dots, A_n$  з відомими ймовірностями настання цих подій  $p_1 = P(A_1), p_2 = P(A_2), \dots, p_n = P(A_n)$ . Внаслідок несумісності подій сума ймовірностей їхнього настання дорівнює одиниці:  $p_1 + p_2 + \dots + p_n = 1$ . Припустимо, що  $p_0 = 0$ . На відрізку  $[0, 1]$  числової осі відкладемо значення цих ймовірностей (рис. 8.4).

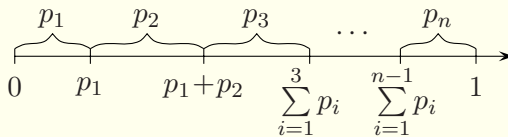


Рис. 8.4.

Нехай  $r$  — рівномірно розподілене на півінтервалі  $[0, 1)$  випадкове число. Якщо воно задовольняє подвійну нерівність

$$\sum_{i=0}^{j-1} p_i \leq r < \sum_{i=0}^j p_i, \quad (8.2)$$

де  $p_0 = 0$ , то вважаємо, що відбулась подія  $A_j$ . Таку процедуру називають визначенням результату випробування за жеребом. Вона ґрунтується на формулі

$$P\left(\sum_{i=0}^{j-1} p_i \leq r < \sum_{i=0}^j p_i\right) = p_j = P(A_j).$$

**Умовні події.** Умовна подія  $A$  — це подія, яка відбувається з імовірністю  $P(A|B)$  тільки за умови, що настала подія  $B$ . Імовірність настання події  $B$  позначимо через  $P(B)$ . Моделювання настання умовної події  $A$  здійснюється наступним чином. Генерується випадкове число  $r_1$ , рівномірно розподілене на півінтервалі  $[0, 1)$ . Подія  $B$  настає в тому випадку, якщо справджується нерівність  $r_1 < P(B)$ . Настання події  $A$  моделюється з допомогою ще одного рівномірно розподіленого на півінтервалі  $[0, 1)$  випадкового числа  $r_2$ . Для нього перевіряється умова  $r_2 < P(A|B)$ , за виконання якої приймається рішення про те, що подія  $A$  відбулась. Якщо подія  $B$  не відбулась (тобто настала подія  $\bar{B}$ ), то настання події  $A$  моделювати не потрібно.

## § 8.4. Метод оберненої функції

У спеціалізованих мовах моделювання, таких як GPSS World, і в математичних пакетах, таких як Maple, MathCad, є готові процедури чи функції для генерування випадкових величин. При використанні звичайних мов програмування випадкові величини, здебільшого, доводиться моделювати самому програмісту.

Метод оберненої функції використовують для моделювання як дискретних, так і неперервних випадкових величин. Його застосовують тоді, коли відомою є функція розподілу випадкової величини  $F(x)$  і вдається знайти обернену до неї функцію  $F^{-1}(r)$ , де  $r$  — випадкове число, рівномірно розподілене на півінтервалі  $[0, 1)$ .

В основі методу оберненої функції лежить той факт, що для будь-якої функції розподілу  $F(x)$  випадкова величина  $\eta = F(\xi)$  має рівномірний розподіл на проміжку  $[0, 1]$ . Доведемо це. Нехай випадкові числа  $\xi_1, \xi_2, \dots, \xi_n$  мають закон розподілу з функцією розподілу

$$F(x) = P(\xi < x). \quad (8.3)$$

Утворимо нову послідовність випадкових чисел  $\eta_1 = F(\xi_1)$ ,  $\eta_2 = F(\xi_2)$ ,  $\dots$ ,  $\eta_n = F(\xi_n)$ . Оскільки значення функції розподілу належать проміжку  $[0, 1]$ , то  $\eta \in [0, 1]$ . Знайдемо функцію розподілу  $G(y)$  випадкової величини  $\eta$ :

$$\begin{aligned} G(y) &= P(\eta < y) = P(F(\xi) < y) = P(F^{-1}(F(\xi)) < F^{-1}(y)) = \\ &= P(\xi < F^{-1}(y)), \end{aligned}$$

де  $F^{-1}(y)$  — обернена функція до функції розподілу  $F(x)$ .

Звідси, враховуючи (8.3), маємо:

$$G(y) = P(\xi < F^{-1}(y)) = F(F^{-1}(y)) = y.$$

Оскільки неперервна випадкова величина з рівномірним законом розподілу на проміжку  $[0, 1]$  має функцію розподілу вигляду  $G(y) = y$ , то випадкова величина  $\eta = F(\xi)$  має рівномірний розподіл на проміжку  $[0, 1]$ .

Таким чином, для моделювання неперервної випадкової величини з функцією розподілу  $F(x)$  методом оберненої функції потрібно розв'язати рівняння  $F(x) = r$ , де  $r$  — випадкова величина, рівномірно розподілена на проміжку  $[0, 1]$ . Тоді  $x = F^{-1}(r)$  — випадкова величина з потрібним законом розподілу, де  $F^{-1}(r)$  — обернена функція до функції розподілу  $F(x)$ . Якщо функція  $F^{-1}(r)$  є неоднозначною, то для кожного проміжку значень змінної  $r$  вибирають потрібну вітку функції  $F^{-1}(r)$ .

**Приклад.** Вивести формули для моделювання неперервної випадкової величини, щільність розподілу ймовірностей

якої має вигляд

$$f(x) = \begin{cases} 0, & x < 2, \\ 0,1, & 2 \leq x < 5, \\ 0,15, & 5 \leq x < 8, \\ 0,05, & 8 \leq x < 13, \\ 0, & x \geq 13. \end{cases}$$

**Розв'язання.** Знайдемо функцію розподілу  $F(x)$  за формулою

$$F(x) = \int_{-\infty}^x f(x) dx.$$

Тоді

$$F(x) = \begin{cases} \int_{-\infty}^x 0 dx, & x < 2, \\ \int_{-\infty}^2 0 dx + \int_2^x 0,1 dx, & 2 \leq x < 5, \\ \int_{-\infty}^2 0 dx + \int_2^5 0,1 dx + \int_5^x 0,15 dx, & 5 \leq x < 8, \\ \int_{-\infty}^2 0 dx + \int_2^5 0,1 dx + \int_5^8 0,15 dx + \int_8^x 0,05 dx, & 8 \leq x < 13, \\ 1, & x \geq 13 \end{cases} \Rightarrow$$

$$F(x) = \begin{cases} 0, & x < 2, \\ 0,1x - 0,2, & 2 \leq x < 5, \\ 0,15x - 0,45, & 5 \leq x < 8, \\ 0,05x + 0,35, & 8 \leq x < 13, \\ 1, & x \geq 13. \end{cases}$$

Розглянемо рівняння  $F(x) = r$ :

$$\begin{aligned} 0,1x - 0,2 &= r, & 2 \leq x < 5, \\ 0,15x - 0,45 &= r, & 5 \leq x < 8, \\ 0,05x + 0,35 &= r, & 8 \leq x < 13. \end{aligned}$$

Знайдемо його розв'язок  $x = F^{-1}(r)$ :

$$x = \begin{cases} 10r + 2, & 0 \leq r < 0,3, \\ \frac{20}{3}r + 3, & 0,3 \leq r < 0,75, \\ 20r - 7, & 0,75 \leq r < 1. \end{cases}$$

Межі діапазонів значень  $r$  отримані підстановкою відповідних значень  $x$  у рівняння  $F(x) = r$ . ■

Аналогічно метод оберненої функції використовують для моделювання дискретних випадкових величин. Припустимо, що дискретна випадкова величина  $\xi$  може набувати значень  $x_1, x_2, \dots$ , для яких  $x_1 < x_2 < \dots$ , а її функція розподілу ймовірностей має вигляд

$$F(x) = P(\xi < x) = \sum_{x_i < x} p(x_i),$$

де  $p(x_i)$  — ймовірнісна міра

$$p(x_i) = P(\xi = x_i).$$

Для моделювання випадкової величини методом оберненої функції генеруємо рівномірно розподілене випадкове число  $r \in [0, 1)$  і знаходимо таке найменше ціле число  $k$ , для якого  $r < F(x_k)$ . Тоді число  $x_k$  буде значенням випадкової величини з потрібним законом розподілу.

Зокрема, формули (8.2) для моделювання однієї події з групи незалежних подій можна вивести також методом оберненої функції.

Якщо функція розподілу  $F(x)$  не існує в елементарних функціях (як це є, наприклад, у випадку нормального розподілу) або рівняння  $F(x) = r$  не розв'язується аналітично, а лише чисельно, то метод оберненої функції застосувати неможливо.

Розвитком методу оберненої функції є табличний метод. Він полягає в тому, що функцію розподілу замінюють таблицею значень функції в окремих точках, а значення функції

між цими точками задають лінійною інтерполяцією. Отриману кусково-лінійну функцію можна моделювати методом оберненої функції. Однак треба мати на увазі, що табличний метод моделювання є наближеним.

### § 8.5. Моделювання дискретних випадкових величин

Розглянемо моделювання кількох найбільш вживаних дискретних випадкових величин.

**Цілочислова рівномірно розподілена випадкова величина.** Для генерування цілочислової рівномірно розподіленої на проміжку  $[a, b]$  випадкової величини використовують формулу

$$k = [a + (b - a + 1)r], \quad (8.4)$$

де квадратні дужки означають цілу частину числа, а  $r$  — випадкове число з рівномірним розподілом на проміжку  $[0, 1)$ .

**Геометричний розподіл.** Прикладом випадкової величини з геометричним розподілом може бути загальна кількість випробувань, які потрібно провести до першого успішного випробування, наприклад, кількість пострілів, які потрібно виконати до першого влучення в ціль.

Імовірність того, що випадкова величина з геометричним розподілом набуває значення  $k$ , визначається за формулою

$$P(k) = p(1 - p)^{k-1}, \quad k = 1, 2, \dots, \quad (8.5)$$

де  $p \in (0, 1)$  — ймовірність успіху в одному випробуванні. Зокрема,  $P(1) = p$ ,  $P(2) = p(1 - p)$  і т. д.

Для отримання значення випадкової величини з геометричним розподілом використовують формулу

$$k = \left[ \frac{\ln r}{\ln(1 - p)} \right] + 1, \quad (8.6)$$

де  $r$  — рівномірно розподілена на проміжку  $[0, 1)$  випадкова величина, а квадратні дужки означають цілу частину числа.

Доведемо формулу (8.6):

$$\begin{aligned} P\left(k \leq \frac{\ln r}{\ln(1-p)} + 1 < k+1\right) &= P\left(k-1 \leq \frac{\ln r}{\ln(1-p)} < k\right) = \\ &= P(k \ln(1-p) < \ln r \leq (k-1) \ln(1-p)) = \\ &= P\left((1-p)^k < r \leq (1-p)^{k-1}\right) = (1-p)^{k-1} - (1-p)^k = \\ &= (1-p)^{k-1}(1-1+p) = p(1-p)^{k-1}. \end{aligned}$$

Треба мати на увазі, що для  $r = 0$  спроба обчислити  $k$  за формулою (8.6) призведе до повідомлення про помилку. У цьому випадку потрібно згенерувати найбільше для вжитого типу даних ціле число.

У книгах багатьох іноземних авторів формула (8.5) є трохи іншою:  $P(k) = p(1-p)^k$ , де  $k$  — кількість невдалих спроб. Тому в них формула (8.6) не містить додавання одиниці.

**Біноміальний розподіл.** Біноміальний розподіл задає ймовірність  $k$  успіхів у серії  $n$  незалежних випробувань, якщо відомою є ймовірність  $p$  успіху в одному випробуванні. Формула для ймовірності має вигляд

$$P_n(k) = \frac{n!}{(n-k)!k!} p^k (1-p)^{n-k},$$

а функція розподілу ймовірностей —

$$F(k) = \sum_{j=0}^k \frac{n!}{(n-j)!j!} p^j (1-p)^{n-j}, \quad k = 0, 1, \dots, n.$$

Залежно від значення  $n$  можна вибрати один з двох способів моделювання випадкової величини з біноміальним законом розподілу. За невеликих  $n$  значення біноміально розподіленої випадкової величини визначають як кількість менших від значення  $p$  чисел у послідовності  $\{r_i\}$  з  $n$  чисел. Цей метод

ґрунтується на самому сенсі біноміально розподіленої випадкової величини. Припустимо, наприклад, що потрібно отримати значення випадкової величини, відповідної біноміальному розподілу з параметрами  $n = 6$  і  $p = 0,35$ . Для цього спочатку генеруємо послідовність із шести рівномірно розподілених випадкових чисел  $r_i \in [0, 1)$ , наприклад: 0,0871, 0,5691, 0,3275, 0,1298, 0,8105, 0,6801, а потім рахуємо ті з них, які є меншими від  $p$ . У даному випадку три числа менші від 0,35. Таким чином, знайдене значення випадкової величини, розподіленої за біноміальним законом, дорівнює 3.

Для великих значень  $n$  краще використовувати інший спосіб моделювання випадкової величини з біноміальним законом розподілу. Він ґрунтується на використанні функції розподілу  $F(k)$  і методу оберненої функції. У цьому випадку генерують одне рівномірно розподілене випадкове число  $r \in [0, 1)$  і визначають таке найменше ціле число  $k$ , щоб виконувалась нерівність

$$r < F(k),$$

тобто

$$r < \sum_{j=0}^k \frac{n!}{(n-j)!j!} p^j (1-p)^{n-j}$$

для заданих  $n$  і  $p$ . Знайдене  $k$  є випадковим числом, яке має біноміальний розподіл. Для реалізації цього методу можна використовувати наступний алгоритм.

**Крок 1.** Генеруємо рівномірно розподілене випадкове число  $r \in [0, 1)$  і виконуємо присвоєння  $b := (1-p)^n$ ,  $s := k$ ,  $k := 0$ .

**Крок 2.** Якщо  $r < s$ , то виводимо  $k$  і закінчуємо роботу алгоритму, інакше переходимо до кроку 3.

**Крок 3.** Обчислюємо  $k := k+1$ ,  $b := b \cdot (n-k+1) \cdot p / (1-p) / k$ ,  $s := s + b$ . Переходимо до кроку 2.

**Розподіл Пуассона.** Пуассонівська випадкова величина набуває значень  $k = 0, 1, 2, \dots$  з імовірностями

$$P(k) = \frac{\lambda^k}{k!} e^{-\lambda},$$

де  $\lambda$  — середнє значення. Закону розподілу Пуассона відповідає та кількість подій у стаціонарному пуассонівському потоці, яка відбувається за певний проміжок часу  $t$ . У цьому випадку  $\lambda = \Lambda t$ , де  $\Lambda$  — інтенсивність пуассонівського потоку (тобто середня кількість подій, що відбувається за одиницю часу). Функція розподілу ймовірностей для закону розподілу Пуассона має вигляд:

$$F(k) = \sum_{j=0}^k \frac{\lambda^j}{j!} e^{-\lambda}.$$

Найпростіший спосіб створення випадкової величини з пуассонівським законом розподілу ґрунтується на методі оберненої функції: генерується рівномірно розподілене випадкове число  $r \in [0, 1)$  і визначається таке найменше ціле число  $k$ , щоб виконувалась нерівність

$$r < F(k),$$

тобто

$$r e^{\lambda} < \sum_{j=0}^k \frac{\lambda^j}{j!}.$$

Для реалізації цього методу можна використовувати наступний алгоритм.

**Крок 1.** Генеруємо рівномірно розподілене випадкове число  $r \in [0, 1)$  і виконуємо присвоєння  $r1 := r \cdot e^{\lambda}$ ,  $s := 1$ ,  $b := 1$ ,  $k := 0$ .

**Крок 2.** Якщо  $r1 < s$ , то виводимо  $k$  і закінчуємо роботу алгоритму, інакше переходимо до кроку 3.

**Крок 3.** Обчислюємо  $k := k + 1$ ,  $b := b \cdot \lambda/k$ ,  $s := s + b$ .  
Переходимо до кроку 2.

Існують також більш швидкі алгоритми моделювання випадкової величини з розподілом Пуассона, які базуються на інших формулах (див. [13, 18]).

Про способи моделювання інших дискретних випадкових величин можна дізнатись з книги [18].

### § 8.6. Моделювання неперервних випадкових величин

Цей параграф присвячено моделюванню найважливіших неперервних випадкових величин.

**Рівномірний розподіл.** Для генерування рівномірно розподіленої на проміжку  $[a, b]$  неперервної випадкової величини використовують формулу

$$x = a + (b - a)r, \quad (8.7)$$

де  $r$  — випадкове число з рівномірним розподілом на проміжку  $[0, 1)$ . Значення  $b$  при цьому ніколи не буде згенероване, але у цьому немає нічого страшного, бо ймовірність набуття конкретного значення з проміжку дорівнює нулю.

Якщо про випадкову величину відомо лише те, що вона змінюється на проміжку  $[a, b]$ , то для її моделювання використовують рівномірний закон розподілу на цьому відрізку.

**Експоненціальний розподіл.** Щільність експоненціального розподілу подається формулою

$$f(x) = \begin{cases} 0, & x < 0, \\ \lambda e^{-\lambda x}, & x \geq 0, \end{cases}$$

а функція розподілу має вигляд

$$F(x) = \begin{cases} 0, & x < 0, \\ 1 - e^{-\lambda x}, & x \geq 0, \end{cases} \quad (8.8)$$

де параметр  $\lambda$  — обернена величина до середнього значення  $\bar{x}$  експоненціально розподіленої випадкової величини. Для моделювання такої випадкової величини скористаємось методом оберненої функції. Розв'язком рівняння

$$1 - e^{-\lambda x} = r$$

є функція

$$x = -\frac{1}{\lambda} \ln(1 - r), \quad (8.9)$$

де  $r \in [0, 1)$  — рівномірно розподілене випадкове число. Формулу (8.9) використовують для моделювання випадкової величини з експоненціальним законом розподілу. Замість  $\frac{1}{\lambda}$  у формулі (8.9) можна використовувати середнє значення  $\bar{x}$ . Оскільки число  $1 - r$  має той самий розподіл, що й  $r$ , для створення експоненціально розподіленої випадкової величини можна було б використовувати формулу

$$x = -\frac{1}{\lambda} \ln r.$$

Однак в останньому випадку потрібно додатково виконувати перевірку виконання рівності  $r = 0$ , бо спроба обчислити логарифм нуля призведе до повідомлення про помилку. Заміна арифметичної операції на логічну лише сповільнює роботу алгоритму.

**Розподіл Ерланга.** Оскільки випадкова величина Ерланга  $k$ -го порядку з параметром  $\lambda$  є сумою  $k$  експоненціально розподілених випадкових величин з параметром  $\lambda$ , то її моделювання здійснюють за формулою

$$x = -\frac{1}{\lambda} \sum_{i=1}^k \ln r_i \quad \Rightarrow \quad x = -\frac{1}{\lambda} \ln \prod_{i=1}^k r_i, \quad (8.10)$$

де  $r_i \in [0, 1)$  — рівномірно розподілені випадкові числа. У формулі (8.10) потрібно враховувати можливість перетворення в нуль добутку випадкових чисел.

**Нормальний розподіл.** Щільність нормального розподілу подається формулою

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}},$$

а його функція розподілу має вигляд

$$F(x) = \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^x e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx,$$

де  $\mu$  — середнє значення, а  $\sigma$  — середньоквадратичне відхилення.

Для нормального розподілу метод оберненої функції можна застосувати лише наближено, замінивши функцію  $f(x)$  деякою іншою функцією.

Інший метод полягає у використанні центральної граничної теореми, яку можна сформулювати таким чином.

**Теорема.** Якщо взаємно незалежні випадкові величини  $\xi_1, \xi_2, \dots, \xi_n$  є однаково розподіленими і мають математичне сподівання  $a$  і дисперсію  $\sigma^2$ , то при  $n \rightarrow \infty$  рівномірно по  $x$

$$P\left(\frac{\sum_{k=1}^n \xi_k - na}{\sigma\sqrt{n}} < x\right) \rightarrow \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{x^2}{2}} dx.$$

Математичне сподівання рівномірно розподіленої на проміжку  $[c, d]$  випадкової величини визначається формулою  $\frac{c+d}{2}$ , а дисперсія — формулою  $\frac{(d-c)^2}{12}$ .

Отже, достатньо взяти 12 рівномірно розподілених на проміжку  $[0, 1]$  випадкових чисел  $r_i$ , для яких  $a = \frac{0+1}{2} = \frac{1}{2}$ ,  $\sigma^2 = \frac{(1-0)^2}{12} = \frac{1}{12}$ . Тоді формула

$$z = \sum_{i=1}^{12} r_i - 6$$

дає нормально розподілену випадкову величину з середнім значенням 0 і середньоквадратичним відхиленням 1, тобто випадкову величину зі стандартним нормальним законом розподілу. Щоб отримати з неї нормально розподілену випадкову величину з довільними значеннями  $\mu$  і  $\sigma$ , треба скористатись формулою  $x = \mu + \sigma z$ . Таким чином, ми приходимо до остаточної формули

$$x = \mu + \sigma \left( \sum_{i=1}^{12} r_i - 6 \right).$$

Останній розглянутий метод також є наближеним. Збільшення кількості доданків дозволяє збільшити точність цього методу, але одночасно сповільнює його.

Точним і значно швидшим методом є метод полярних координат (метод Марсальї–Брея). Для його застосування використовують наступний алгоритм [18].

**Крок 1.** Генеруємо два рівномірно розподілених випадкових числа  $r_1, r_2 \in [0, 1)$  і виконуємо обчислення  $v_1 := 2r_1 - 1$ ,  $v_2 := 2r_2 - 1$ ,  $W := v_1^2 + v_2^2$ .

**Крок 2.** Якщо  $W > 1$ , то переходимо до кроку 1, інакше — до кроку 3.

**Крок 3.** Обчислюємо  $y := \sqrt{-\frac{2 \ln W}{W}}$  і знаходимо два нормально розподілених числа:  $x_1 := \mu + \sigma v_1 y$ ,  $x_2 := \mu + \sigma v_2 y$ .

**Логнормальний розподіл.** Випадкова величина, розподілена за логнормальним законом розподілу, є експонентою випадкової величини, розподіленої за нормальним законом розподілу. Функція щільності логнормального розподілу подається формулою

$$f(x) = \begin{cases} 0, & x \leq 0, \\ \frac{1}{x\sqrt{2\pi}\sigma} e^{-\frac{(\ln x - \mu)^2}{2\sigma^2}}, & x > 0, \end{cases}$$

де  $\mu$  і  $\sigma$  — середнє значення і середньоквадратичне відхилення відповідного нормального закону розподілу.

З самого означення логнормально розподіленої випадкової величини видно, що її генерують за формулою

$$x = e^y,$$

де  $y$  — нормально розподілена випадкова величина з середнім значенням  $\mu$  і середньоквадратичним відхиленням  $\sigma$ .

**Розподіл Вейбулла.** Щільність імовірностей для розподілу Вейбулла подається формулою

$$f(x) = \begin{cases} 0, & x \leq 0, \\ \frac{\alpha}{\beta} \left(\frac{x}{\beta}\right)^{\alpha-1} e^{-\left(\frac{x}{\beta}\right)^\alpha}, & x > 0, \end{cases}$$

а функція розподілу має вигляд

$$F(x) = \begin{cases} 0, & x \leq 0, \\ 1 - e^{-\left(\frac{x}{\beta}\right)^\alpha}, & x > 0, \end{cases}$$

де  $\alpha$  — параметр форми, а  $\beta$  — коефіцієнт стиснення,  $\alpha > 0$ ,  $\beta > 0$ . Для моделювання такої випадкової величини скористаємось методом оберненої функції. Розв'язок рівняння

$$1 - e^{-\left(\frac{x}{\beta}\right)^\alpha} = r$$

дає формулу

$$x = \beta \left( -\ln(1 - r) \right)^{1/\alpha}$$

для моделювання випадкової величини з розподілом Вейбулла, де  $r \in [0, 1)$  — рівномірно розподілене випадкове число.

**Трикутний розподіл.** Графік функції щільності трикутного розподілу разом з віссю абсцис утворює трикутник одиничної площі (рис. 8.5).

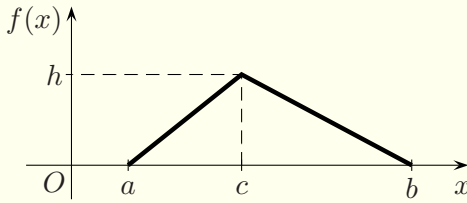


Рис. 8.5.

Тоді

$$\frac{1}{2}(b-a)h = 1 \quad \Rightarrow \quad h = \frac{2}{b-a}.$$

Будуємо рівняння прямих, які проходять через точки  $(a, 0)$ ,  $(c, 2/(b-a))$  і  $(b, 0)$ ,  $(c, 2/(b-a))$ :

$$\frac{y}{2/(b-a)} = \frac{x-a}{c-a} \quad \Rightarrow \quad y = \frac{2(x-a)}{(b-a)(c-a)}.$$

$$\frac{y}{2/(b-a)} = \frac{x-b}{c-b} \quad \Rightarrow \quad y = \frac{2(x-b)}{(b-a)(c-b)}.$$

Отже, у випадку  $a < c < b$  одержуємо функцію щільності

$$f(x) = \begin{cases} \frac{2(x-a)}{(b-a)(c-a)}, & a \leq x < c, \\ \frac{2(x-b)}{(b-a)(c-b)}, & c \leq x \leq b, \\ 0, & x < a \text{ або } x > b, \end{cases}$$

зінтегрувавши яку, знаходимо функцію розподілу

$$F(x) = \begin{cases} 0, & x < a, \\ \frac{(x-a)^2}{(b-a)(c-a)}, & a \leq x < c, \\ 1 + \frac{(x-b)^2}{(b-a)(c-b)}, & c \leq x \leq b, \\ 1, & x > b. \end{cases}$$

Розв'язуючи рівняння  $F(x) = r$ , отримуємо формули для моделювання трикутного розподілу:

$$x = a + \sqrt{r(b-a)(c-a)}, \quad r < \frac{c-a}{b-a}, \quad (8.11)$$

$$x = b - \sqrt{(1-r)(b-a)(b-c)}, \quad r \geq \frac{c-a}{b-a}, \quad (8.12)$$

де  $r \in [0, 1)$  — рівномірно розподілене випадкове число.

Якщо  $a = c$  або  $b = c$ , то ми маємо лівий або правий трикутний закон розподілу. Формули для їхнього моделювання виводяться аналогічно.

Моделювання емпіричних, зміщених, зрізаних і багатовимірних розподілів розглядається в розділі 9. Про способи моделювання інших неперервних випадкових величин можна дізнатись із книги [18].

### § 8.7. Приклади моделювання

Розглянемо кілька прикладів розв'язування задач з допомогою методу Монте-Карло, для яких немає потреби використовувати мову GPSS World.

**Приклад 1.** Торговий заклад щодня купує у видавництва 1000 примірників деякої газети по ціні 5,5 грошових одиниць і продає по 7 грошових одиниць. З досвіду роботи закладу відомо, що в ранковий час попит на газети зазвичай становить від 200 до 500 примірників. У денний час попит зазвичай становить від 300 до 600 примірників. Попит у вечірній час залежить від того, чи були надруковані в газеті матеріали, що викликали підвищений інтерес. Якщо в газеті були опубліковані такі матеріали, то попит становить від 300 до 600 примірників; якщо таких матеріалів немає, то попит становить від 100 до 300 примірників. Матеріали, що викликають підвищений інтерес, друкуються в газеті в середньому два рази на тиждень. Газети, не продані протягом дня, не можуть бути продані в наступні дні.

Розробити програму для імітації роботи торгового закладу на основі методу Монте-Карло з метою визначення середнього прибутку закладу за день, середньої кількості непроданих за день газет, середньої кількості газет за день, яких не вистачило для задоволення попиту, і ймовірності того, що попит на газети не буде повністю задоволено.

**Розв’язання.** Для моделювання попиту на газети будемо використовувати формулу (8.4), а для моделювання появи в газеті особливо цікавих матеріалів — перевірку умови  $r < 2/7$ , де  $r$  — рівномірно розподілене на проміжку  $[0, 1)$  випадкове число. Для визначення шуканих характеристик достатньо здійснити імітацію роботи торгового закладу протягом 1000 днів. Нижче наводиться програма на мові C++. Для того, щоб отримати в C++ рівномірно розподілене на проміжку  $[0, 1)$  випадкове число, використовують наступний фрагмент коду: `(double)rand()/RAND_MAX`. У програмі змінна `N` містить кількість днів, доки триває моделювання, змінна `Ng` — кількість газет, отриманих кожного дня, змінна `k` — кількість проданих кожного дня газет. У змінній `nvg` накопичується кількість газет, яких не вистачило для задоволення попиту. У змінній `nvd` накопичується кількість днів, коли не вистачило газет. У змінній `npg` накопичується кількість непроданих газет. У змінній `Pr` накопичується прибуток.

```
#include <iostream>
#include <cmath>
#include <cstdlib>
using namespace std;
int main()
{
    int N=1000,Ng=1000;
    int k,nvg=0,nvd=0,npg=0;
    float Pr=0;
    for (int i=0;i<N;i++)
    {
```

```

k=200+rand()%301+300+rand()%301;
if ((double)rand()/RAND_MAX<2.0/7)
    k+=300+rand()%301; else k+=100+rand()%201;
if (k>Ng) {nvg+=k-Ng; nvd++; Pr+=Ng*(7-5.5);}
    else {npg+=Ng-k; Pr+=k*7-Ng*5.5;}
}
cout<<"Прибуток "<<Pr/N<<endl;
cout<<"Не продано газет "<<round(float(npg)/N)
    <<endl;
cout<<"Не вистачило газет "<<round(float(nvg)/N)
    <<endl;
cout<<"Імовірність незадоволення попиту "
    <<(float)nvd/N<<endl;
}

```

Внаслідок роботи програми було отримано такі результати: середній прибуток за день становить близько 1200 грошових одиниць, середня кількість непроданих за день газет — 40, середня кількість газет за день, яких не вистачило для задоволення попиту, — 115, імовірність того, що попит на газети не буде повністю задоволено, — 0,65. ■

**Приклад 2.** Лінія електропередачі (ЛЕП) з'єднує електростанцію з містом. Довжина ЛЕП (відстань між містом і електростанцією) — 200 км. Засоби автоматичного контролю, якими обладнана ЛЕП, дозволяють у випадку аварії зразу встановити місце, де вона відбулась, і оцінити її серйозність. Приблизно у 85 % випадків аварія є звичайною, а у 15 % — серйозною. При звичайній аварії для її усунення скеровується ремонтна бригада з пункту (міста чи електростанції), ближчого до місця аварії. Для усунення серйозної аварії скеровується спеціальна ремонтна бригада з міста, незалежно від того, де відбулась аварія.

Для виїзду звичайної ремонтної бригади потрібно 5–10 хвилин, а для виїзду спеціальної бригади — 10–20 хвилин.

Відстань до місця аварії по дорозі в середньому є на 10 % більшою, ніж по лінії електропередачі. Швидкість руху ремонтної бригади до місця аварії лежить у межах від 30 до 60 км/год (залежно від якості дороги, якою можна дістатись безпосередньо до місця аварії), найімовірніша швидкість руху — 50 км/год. Час, необхідний для ремонту ЛЕП — випадкова величина, яку можна вважати розподіленою за експоненціальним законом розподілу. Ремонт ЛЕП у випадку звичайної аварії потребує в середньому 4 години, а серйозної — 8 годин.

Розробити програму для імітації ремонту ЛЕП на основі методу Монте-Карло і визначення середнього часу, необхідного для усунення аварії (включно з часом, необхідним для прибуття ремонтної бригади).

**Розв’язання.** Для моделювання відстані від міста до місця аварії на ЛЕП і часу, необхідного для виїзду ремонтної бригади, будемо використовувати неперервно розподілену випадкову величину на заданому проміжку, яка задається формулою (8.7). Для моделювання несерйозності аварії застосуємо перевірку умови  $r < 0,85$ , де  $r$  — рівномірно розподілене на проміжку  $[0, 1)$  випадкове число. Для визначення швидкості руху ремонтної бригади використаємо випадкову величину з трикутним законом розподілу з межами 30 і 60 та найімовірнішим значенням (точкою  $c$ ), рівним 50. Для моделювання такої випадкової величини призначені формули (8.11)–(8.12). Час руху ремонтної бригади до місця аварії визначимо діленням відстані на швидкість. Для моделювання часу ремонту ЛЕП будемо використовувати формулу (8.9) з середнім значенням замість  $1/\lambda$ . Для визначення середнього часу, необхідного для усунення аварії, достатньо здійснити імітацію усунення 5000 аварій на ЛЕП. Нижче наводиться програма на мові C++. У програмі час вимірюється у годинах і накопичується у змінній  $t$ . У змінній  $v_{\text{шт}}$  зберігається відстань від міста у кілометрах, а в змінній  $v$  — швидкість.

```
#include <iostream>
```

```
#include <cmath>
#include <cstdlib>
using namespace std;
int main()
{
    int N=5000;
    double t=0,vvm,v,r;
    for (int i=0; i<N; i++)
    {
        vvm=200*(double)rand()/RAND_MAX;
        r=(double)rand()/RAND_MAX;
        if (r<2.0/3) v=30+sqrt(r*600);
        else v=60-sqrt((1-r)*300); // трикутний розподіл
        if ((double)rand()/RAND_MAX<0.85)
        {
            t+=1.0/12+(double)rand()/RAND_MAX/12;
            if (vvm<=100) t+=vvm*1.2/v;
            else t+=(200-vvm)*1.2/v;
            t+=-4*log(1-(double)rand()/RAND_MAX);
        }
        else t+=1.0/6+(double)rand()/RAND_MAX/6+vvm*1.2/v-
            8*log(1-(double)rand()/RAND_MAX);
    }
    cout<<t/N<<endl;
}
```

Робота програми показала, що середній час усунення аварії приблизно дорівнює 6,3 години. ■

## Питання до розділу 8

1. У чому полягає метод Монте-Карло? Для розв'язування яких задач його можна застосовувати?

2. Назвіть методи генерації рівномірно розподілених випадкових чисел. У якому випадку лінійний конгруентний генератор для

будь-якого початкового значення дає послідовність випадкових чисел з максимально можливим періодом?

3. Як при застосуванні методу Монте-Карло моделюють настання випадкової події? Як моделюють одну подію з групи несумісних подій? Як моделюють умовну подію?

4. У чому полягає метод оберненої функції?

5. За якими формулами моделюють цілочислові рівномірно розподілені випадкові величини і випадкові величини з геометричним законом розподілу? У чому полягають методи моделювання випадкових величин з біноміальним розподілом і розподілом Пуассона?

6. Як моделюють випадкові величини з рівномірним законом розподілу на проміжку  $[a, b]$  (експоненціальним, нормальним, логнормальним, трикутним розподілом, розподілом Ерланга, розподілом Вейбулла)?

### Вправи до розділу 8

1. Підприємство по ремонту побутової електроніки виконує ремонт холодильників, телевізорів, порохотягів і телефонів. Відомо, що приблизно 10 % замовлень, які надходять на підприємство, становлять замовлення на ремонт холодильників, 35 % — телевізорів, 25 % — порохотягів, 30 % — телефонів. По кожному замовленню потрібен ремонт лише одного виробу. Середній прибуток підприємства від ремонту одного холодильника становить 65 грошових одиниць, телевізора — 60 грошових одиниць, порохотяга — 30 грошових одиниць, телефону — 40 грошових одиниць. Розробіть програму для імітації роботи ремонтного підприємства на основі методу Монте-Карло і визначення прибутку підприємства від виконання 100 ремонтів побутової електроніки.

2. Підприємство виготовляє деякі вироби. Зі спостережень відомо, що 4 % виробів є бракованими. Виготовлення придатного виробу приносить підприємству прибуток у розмірі 10 грошових одиниць, а виготовлення дефектного виробу — збиток у розмірі 15 грошових одиниць. Створіть програму імітації виготовлення виробів на основі методу Монте-Карло для визначення середнього прибутку підприємства від виготовлення одного виробу.

3. Підприємство виготовляє електроприлади, які складаються з трьох блоків  $A$ ,  $B$  і  $C$ . Якщо хоч один із блоків виходить з ладу, то

прилад працювати не буде. З досвіду відомо, що ймовірність безвідмовної роботи блока  $A$  протягом гарантійного терміну становить 95 %, блока  $B$  — 97 %, блока  $C$  — 98 %. Витрати підприємства на виготовлення одного приладу наступні: вартість кожного блока — 10 грошових одиниць, інші витрати — 15 грошових одиниць. Прилади продаються по ціні 60 грошових одиниць. У випадку відмови приладу до закінчення гарантійного терміну підприємство безкоштовно виконує його ремонт. Витрати підприємства, пов'язані з гарантійним ремонтом, можуть бути різними в залежності від складності ремонту, витрат на доставку і т. п. Відомо, що приблизно у 60 % випадків витрати на гарантійний ремонт становлять 20 грошових одиниць, у 35 % випадків — 30 грошових одиниць, у 5 % випадків — 45 грошових одиниць.

Потрібно скласти алгоритм і програму на основі методу Монте-Карло для знаходження: а) ймовірності відмови приладу до завершення гарантійного терміну; б) середнього прибутку підприємства від випуску одного приладу. Визначте, чи вигідно для підприємства використовувати в конструкції приладу новий, більш надійний блок  $A$ , вартість якого становить 13 грошових одиниць, а ймовірність безвідмовної роботи протягом гарантійного терміну — 99 % (при цьому підвищення ціни на прилади не планується).

4. Підприємство виготовляє книжкові шафи на замовлення. Для великих замовлень передбачена знижка: за виготовлення однієї шафи підприємство бере 30 грошових одиниць, якщо замовлення становить від однієї до п'яти шаф, і 25 грошових одиниць — якщо замовлення перевищує п'ять шаф. Витрати підприємства на виготовлення однієї шафи — 20 грошових одиниць. З досвіду роботи підприємства відомо, що замовлення зазвичай становлять від однієї до тридцяти шаф, причому ця величина є рівномірно розподіленою на цілому проміжку. Замовлення на виготовлення понад тридцяти шаф зустрічаються надзвичайно рідко і ними можна знехтувати. Директор підприємства пропонує скасувати знижку з метою збільшення прибутку. Однак у цьому випадку можна припускати, що кількість великих замовлень зменшиться, і замовлення будуть передбачати виготовлення в середньому від однієї до п'ятнадцяти шаф. Потрібно скласти програму для імітації виконання замовлень і визначити, чи вигідним для підприємства є скасування знижок.

5. У ремонтній службі підприємства виконується налагоджен-

ня і ремонт деяких пристроїв. Кожен пристрій складається з п'яти схем. Кожна з них може виявитись несправною з імовірністю 0,1 (у 10 % випадків). Налагодження і ремонт пристрою включає в себе наступні операції: перевірку, заміну несправної схеми і налагодження пристрою. Для перевірки кожної схеми потрібно витратити від 3 до 6 хвилин. Час заміни однієї схеми є випадковою величиною з логнормальним законом розподілу з середнім значенням відповідного нормального закону розподілу 1,8 і середньоквадратичним відхиленням 0,2. Якщо схеми не замінювались, то достатньо дрібного налагодження пристрою. Якщо було замінено хоч одну схему, то необхідне повне налагодження пристрою. Час налагодження є випадковою величиною, розподіленою за експоненціальним законом. Дрібне налагодження потребує в середньому 10 хвилин, а повне — 15 хвилин.

Потрібно скласти алгоритм і програму імітації налагодження і ремонту пристроїв на основі методу Монте-Карло. Програма має дати можливість визначити: середній час налагодження і ремонту одного пристрою; процент випадків, коли відбувся ремонт, тобто була замінена хоч одна схема; середню кількість схем, замінених в одному пристрої.

6. Деякі вироби виготовляються шляхом накладання двох пластин. Номінальна товщина першої пластини — 9 мм, а другої — 6 мм. Оскільки у виробничому процесі неминучі відхилення, фактична товщина першої пластини є нормально розподіленою випадковою величиною з середнім значенням 9 мм і середньоквадратичним відхиленням 0,15 мм. Товщина другої пластини — теж нормально розподілена випадкова величина з середнім значенням 6 мм і середньоквадратичним відхиленням 0,1 мм. Готовий виріб вважається придатним, якщо його товщина становить від 14,8 до 15,2 мм. Створіть алгоритм і програму для імітації товщини виготовленого виробу на основі методу Монте-Карло. Знайдіть середню товщину готового виробу і ймовірність випуску придатної продукції.

7. Фірма виконує переклади з двадцяти п'яти іноземних мов на українську мову. Обсяг перекладу — випадкова величина з експоненціальним законом розподілу. Середній обсяг перекладу — 7 сторінок. Сторінкою вважається обсяг тексту, що містить 1860 друкованих знаків з пропусками.

З усіх перекладів переклади з англійської, німецької, французь-

кої, італійської та іспанської мов становлять 85 %, з польської мови — 5 %, з інших мов — 10 %. Крім того, відомо, що з усіх замовлень 50 % становлять замовлення на стандартні переклади, 30 % — економічні і юридичні, 20 % — художні, медичні і технічні. Кожне замовлення належить лише до одного виду (тобто переклад не може бути, наприклад, одночасно економічним і медичним).

Плата, яку бере фірма з замовника за одну сторінку стандартного перекладу, наступна: з англійської, німецької, французької, італійської та іспанської мов — 45 грошових одиниць, з польської мови — 40 грошових одиниць, з інших мов — 65 грошових одиниць. За економічний або юридичний переклад плата підвищується на 30 %, а за художній, медичний або технічний — на 50 %. Якщо обсяг перекладу становить менше, ніж півсторінки, то плата береться за півсторінки. Якщо обсяг перекладу перевищує 50 сторінок, то застосовується знижка у розмірі 5 %.

Потрібно скласти програму на основі методу Монте-Карло для визначення середньої вартості виконання одного перекладу.

8. В автоматизованій системі керування технологічним процесом передаються сигнали від виробничого обладнання (об'єкта керування) до комп'ютера, який керує процесом. Тривалість передачі сигналу — випадкова величина, розподілена за експоненціальним законом із середнім значенням 3 мс. У каналі зв'язку можуть виникати перешкоди. Інтервали між перешкодами — випадкові величини, розподілені за експоненціальним законом. Перешкоди виникають у середньому 30 разів за секунду, але є короткочасними. Якщо під час передачі сигналу виникне хоч одна перешкода, то сигнал спотвориться. Розробіть алгоритм і програму для імітації передачі сигналів. Визначте ймовірність передачі сигналу без спотворень.

Вказівка. Оскільки інтервали часу між перешкодами є експоненціально розподіленими випадковими величинами, то кількість перешкод за деякий інтервал часу  $t$  є випадковою величиною, розподіленою за законом розподілу Пуассона з параметром  $\lambda = \Lambda t$ , де  $\Lambda$  — інтенсивність потоку перешкод. У нашому випадку  $\Lambda = 30$  перешкод/с = 0,03 перешкоди/мс.

9. Виведіть формули для генерування випадкової величини, яка має задану щільність розподілу ймовірностей:

$$\begin{array}{l}
 \text{а) } f(x) = \begin{cases} 0, & x < 1, \\ \frac{x-1}{2}, & 1 \leq x < 2, \\ \frac{1}{2}, & 2 \leq x < 3, \\ 2 - \frac{x}{2}, & 3 \leq x < 4, \\ 0, & x \geq 4; \end{cases} & \text{б) } f(x) = \begin{cases} 0, & x < 0, \\ \frac{x^2}{10}, & 0 \leq x < 2, \\ \frac{1}{3}, & 2 \leq x < 3, \\ 1 - \frac{x}{5}, & 3 \leq x < 5, \\ 0, & x \geq 5; \end{cases} \\
 \\
 \text{в) } f(x) = \begin{cases} 0, & x < 2, \\ \frac{x-2}{3}, & 2 \leq x < 3, \\ \frac{1}{2}, & 3 \leq x < 4, \\ \frac{1}{3}, & 4 \leq x < 5, \\ 0, & x \geq 5; \end{cases} & \text{г) } f(x) = \begin{cases} 0, & x < -1, \\ \frac{13}{60}, & -1 \leq x < 0, \\ \frac{1}{4}, & 0 \leq x < 2, \\ \frac{1}{5}, & 2 \leq x < 3, \\ \frac{1}{12}, & 3 \leq x < 4, \\ 0, & x \geq 4. \end{cases}
 \end{array}$$

10. Виведіть формули для генерування випадкової величини з лівим трикутним законом розподілу.

11. Виведіть формули для генерування випадкової величини з правим трикутним законом розподілу.

## Розділ 9. Вибір розподілів імовірностей

### § 9.1. Методи використання даних спостережень для моделювання випадкових процесів

Для того щоб здійснити імітаційне моделювання будь-якої ймовірнісної системи, потрібно знати розподіли випадкових величин, які задають випадкові фактори у системі. У виробничій системі випадковими факторами можуть бути час обробки, час безвідмовної роботи, час ремонту обладнання, інтервали часу між надходженням деталей, імовірність відхилення деталей від стандарту, імовірність браку. У системі зв'язку — час між надходженням повідомлень, тип повідомлень, їхня довжина, можливість спотворення повідомлень тощо.

Тому спочатку треба здійснити спостереження за випадковими процесами і отримати дані (вибірку) спостережень, наприклад, моменти появи клієнтів у банку, час їхнього обслуговування і т. п. Звичайно, для отримання даних спостережень потрібно мати реальну систему і можливість здійснення спостережень за нею. Якщо отримати вибірку шляхом спостереження неможливо (наприклад, не існує реалізації самої системи), то доводиться застосовувати спеціальні методи, які розглядаються у § 9.16. Проте точність моделювання у цьому випадку буде невисокою.

Ще раз підкреслимо, що потрібно мати саме вибірку, а не середнє значення випадкової величини, її дисперсію або інші статистичні характеристики, бо є багато зовсім різних розподілів випадкової величини з тими самими середнім значенням, дисперсією чи іншими параметрами. Використання непридатного (неправильного) розподілу, хоч і з потрібними середнім значенням чи деякими іншими характеристиками, може призвести до суттєвих відхилень результатів моделювання від реального стану речей. Наприклад, довжина черги може відрізнятись у кілька разів.

Якщо є можливість зібрати дані спостережень (отримати вибірку) для потрібної випадкової величини, то їх можна застосовувати для моделювання одним із трьох методів. Методи перерахуємо в порядку зростання переваги.

1. Значення даних спостережень використовуються безпосередньо при моделюванні. Наприклад, якщо дані спостережень є часом обслуговування, то кожного разу при імітації обслуговування вибирається одне зі значень із вибірки.

2. Значення даних спостережень використовуються для визначення функції емпіричного розподілу. При моделюванні системи генерується випадкова величина з заданим емпіричним законом розподілу.

3. Методами математичної статистики підбирається форма теоретичного розподілу, визначаються його параметри і перевіряється відповідність знайденого теоретичного розподілу даним спостережень. Для моделювання випадкової величини використовується знайдений теоретичний розподіл.

Перший метод має два недоліки. По-перше, при його використанні може відтворюватись лише те, що відбувалось раніше під час спостереження. По-друге, часто даних спостережень буває недостатньо для виконання моделювання. Два інші методи позбавлені цих недоліків. Принаймні у випадку неперервної випадкової величини ці методи дозволяють генерувати будь-яке значення між точками мінімуму і максимуму даних спостережень або навіть на ширшому проміжку.

Перший метод використовують для перевірки адекватності моделі реальній системі або для порівняння кількох моделей.

Якщо для даних спостережень можна добре підібрати теоретичний розподіл, то часто буває зручнішим і кращим використовувати саме його, а не емпіричний розподіл. Перерахуємо переваги теоретичного розподілу.

1. У функції емпіричного розподілу можуть бути певні спотворення, особливо, якщо розподіл будується за невели-

кою кількістю даних. Теоретичний розподіл згладжує дані і містить інформацію про характер розподілу.

2. Теоретичні розподіли на відміну від емпіричних дозволяють генерувати значення з-поза меж проміжку даних спостережень.

3. У деяких випадках на основі ролі випадкової величини в системі можна зробити висновок про необхідність застосування певної форми теоретичного розподілу як вдалої моделі цієї випадкової величини. Цьому питанню присвячено § 9.8.

4. Теоретичний розподіл простіше змінити. Наприклад, якщо є потреба подивитись, що станеться при збільшенні інтенсивності вхідного потоку на 10 %, то при використанні теоретичного розподілу достатньо змінити лише одне число. Емпіричний розподіл у цьому випадку довелося би повністю переробляти.

5. Якщо в наборі даних спостережень є  $n$  значень, то для моделювання даних з емпіричним законом розподілу потрібно запрограмувати використання  $2n$  чисел, що при великих  $n$  тягне за собою очевидні технічні труднощі. Теоретичний розподіл позбавлений цього недоліку і його особливо зручно використовувати при великій кількості даних спостережень.

Однак існують ситуації, коли неможливо підібрати жоден теоретичний розподіл, який би відповідав даним спостережень. У цьому випадку рекомендується використовувати емпіричний розподіл. Крім того, при використанні багатьох теоретичних розподілів можуть генеруватись як завгодно великі значення, хоча й з малою ймовірністю. Якщо відомо, що випадкова величина не може набувати значень, більших або менших від деякої величини, то потрібно використовувати емпіричні або зрізані чи зміщені теоретичні розподіли.

Емпіричні розподіли та їхнє моделювання розглядаються у §§ 9.2–9.4, вибір теоретичних розподілів — у §§ 9.6–9.13, зміщені і зрізані теоретичні розподіли — у §§ 9.14, 9.15, а спільні

і двовимірні теоретичні розподіли — у §§ 9.18–9.20.

### § 9.2. Емпіричні розподіли неперервних випадкових величин та їхнє моделювання у випадку, коли відомі дані окремих спостережень

Є різні способи створення емпіричних розподілів, які залежать від відповідних способів задання даних.

Нехай відомими є значення окремих спостережень  $x_1, x_2, \dots, x_n$  неперервної випадкової величини. Спочатку відсортуємо величини  $x_i$  за зростанням. Через  $x_{(i)}$  позначимо  $i$ -те найменше значення серед величин  $x_j$  таке, що

$$x_{(1)} \leq x_{(2)} \leq \dots \leq x_{(n)}.$$

Побудуємо неперервну кусково-лінійну функцію розподілу  $F(x)$  так, щоб  $F(x) = 0$  для  $x < x_{(1)}$ ,  $F(x) = 1$  для  $x \geq x_{(n)}$  і

$$F(x_{(i)}) = \frac{i-1}{n-1}, \quad i = 1, 2, \dots, n.$$

Для визначення проміжних значень знайдемо рівняння прямих, які з'єднують точки  $(x_{(i)}, F(x_{(i)}))$  і  $(x_{(i+1)}, F(x_{(i+1)}))$ ,  $x_{(i)} \neq x_{(i+1)}$ ,  $i = 1, 2, \dots, n-1$ :

$$\begin{aligned} \frac{y - F(x_{(i)})}{F(x_{(i+1)}) - F(x_{(i)})} &= \frac{x - x_{(i)}}{x_{(i+1)} - x_{(i)}} \quad \Rightarrow \\ y &= F(x_{(i)}) + \frac{x - x_{(i)}}{x_{(i+1)} - x_{(i)}} (F(x_{(i+1)}) - F(x_{(i)})) \quad \Rightarrow \\ y &= \frac{i-1}{n-1} + \frac{x - x_{(i)}}{(n-1)(x_{(i+1)} - x_{(i)})}. \end{aligned}$$

Отже, функція розподілу  $F(x)$  задається у вигляді

$$F(x) = \begin{cases} 0, & x < x_{(1)}, \\ \frac{i-1}{n-1} + \frac{x - x_{(i)}}{(n-1)(x_{(i+1)} - x_{(i)})}, & x_{(i)} \leq x < x_{(i+1)}, \\ 1, & x \geq x_{(n)}. \end{cases} \quad (9.1)$$

Якщо для деякого значення  $i$  виконується рівність  $x_{(i)} = x_{(i+1)}$ , то формула (9.1) залишається правильною, бо в цьому випадку проміжок  $[x_{(i)}, x_{(i+1)})$  не містить жодного значення.

На рис. 9.1 наведено приклад графіка функції розподілу  $F(x)$  для  $n = 5$ .

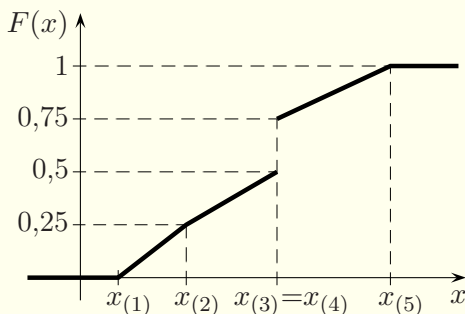


Рис. 9.1.

Для моделювання емпірично розподілених випадкових величин використовують метод оберненої функції (§ 8.4). Таким чином, для відшукування  $x$  потрібно розв'язати рівняння  $F(x) = r$ , де  $r$  — випадкове число, рівномірно розподілене на півінтервалі  $[0, 1)$ . Для функції розподілу  $F(x)$ , яка подається формулою (9.1), маємо:

$$\frac{i-1}{n-1} + \frac{x-x_{(i)}}{(n-1)(x_{(i+1)}-x_{(i)})} = r, \quad x_{(i)} \leq x < x_{(i+1)},$$

$$i = 1, 2, \dots, n-1.$$

Тоді

$$x = x_{(i)} + ((n-1)r - i + 1)(x_{(i+1)} - x_{(i)}), \quad (9.2)$$

причому

$$0 \leq (n-1)r - i + 1 < 1 \Rightarrow i \leq (n-1)r + 1 < i + 1 \Rightarrow$$

$$i = [(n-1)r] + 1, \quad (9.3)$$

де квадратні дужки позначають цілу частину числа.

Отже, для моделювання неперервної випадкової величини, функція розподілу якої подається формулою (9.1), генерують випадкове число  $r$ , рівномірно розподілене на півінтервалі  $[0, 1)$ , а потім знаходять число  $i$  за формулою (9.3) і число  $x$  за формулою (9.2).

**Приклад 1.** Обчислити два числових значення неперервної випадкової величини з емпіричним розподілом, що відповідає вибірці: 12,32; 10,71; 14,5; 13,16; 10,39; 13,74; 10,71; 14,22. Вважати, що генератор випадкових чисел генерує значення 0,869 і 0,236.

**Розв'язання.** Кількість чисел у вибірці  $n = 8$ . Відсортуємо вибірку за зростанням: 10,39; 10,71; 10,71; 12,32; 13,16; 13,74; 14,22; 14,5. Виконуємо обчислення за формулами (9.3), (9.2). Для  $r = 0,869$  маємо:

$$i = [7 \cdot 0,869] + 1 = 7,$$

$$x = 14,22 + (7 \cdot 0,869 - 7 + 1)(14,5 - 14,22) = 14,24324.$$

Для  $r = 0,236$  маємо:

$$i = [7 \cdot 0,236] + 1 = 2,$$

$$x = 10,71 + (7 \cdot 0,236 - 2 + 1)(10,71 - 10,71) = 10,71.$$

**Відповідь:** 14,24324, 10,71.

При використанні мови імітаційного моделювання GPSS обчислення за формулами (9.3), (9.2) здійснювати не потрібно, бо ці дії виконуються автоматично при використанні функції, яка описується командою FUNCTION (§ 6.13). У цьому випадку її записують так:

$$N \text{ FUNCTION RN}m,Cn$$

$$0, x_{(1)}/\frac{1}{n-1}, x_{(2)}/\frac{2}{n-1}, x_{(3)}/\dots/\frac{n-2}{n-1}, x_{(n-1)}/1, x_{(n)}$$

Замість  $N$  вказують ім'я функції, число  $n$  — кількість чисел у вибірці даних спостережень,  $m$  — номер генератора випадкових чисел, а  $x_{(1)}, x_{(2)}, \dots, x_{(n)}$  — відсортовані за зростанням дані спостережень. Усі дробни при визначенні функції мають бути обчисленими.

**Приклад 2.** На мові імітаційного моделювання GPSS створити неперервну функцію для моделювання емпіричного розподілу, що відповідає вибірці: 12,32; 10,71; 14,5; 13,16; 10,39; 13,74; 10,71; 14,22.

**Розв'язання.** Кількість чисел у вибірці  $n = 8$ . Відсортуємо вибірку за зростанням: 10,39; 10,71; 10,71; 12,32; 13,16; 13,74; 14,22; 14,5. Функцію записуємо, наприклад, так:

```
abcd FUNCTION RN15,C8
0,10.39/1/7,10.71/2/7,10.71/3/7,12.32/4/7,13.16/5/7,13.74/
6/7,14.22/1,14.5
```

Після наближеного обчислення дробів маємо:

```
abcd FUNCTION RN15,C8
0,10.39/0.143,10.71/0.286,10.71/0.429,12.32/
0.571,13.16/0.714,13.74/0.857,14.22/1,14.5 ■
```

### § 9.3. Емпіричні розподіли дискретних випадкових величин та їхнє моделювання у випадку, коли відомі дані окремих спостережень

Припустимо, що відомими є значення окремих спостережень  $x_1, x_2, \dots, x_n$  дискретної випадкової величини. Відсортуємо їх за зростанням і перепозначимо через  $x_{(i)}$ :

$$x_{(1)} \leq x_{(2)} \leq \dots \leq x_{(n)}.$$

З курсу теорії ймовірностей і математичної статистики відомою є формула для відповідної функції розподілу:

$$F(x) = \begin{cases} 0, & x < x_{(1)}, \\ \frac{i}{n}, & x_{(i)} \leq x < x_{(i+1)}, \quad i = 1, 2, \dots, n-1, \\ 1, & x \geq x_{(n)}. \end{cases} \quad (9.4)$$

На рис. 9.2 наведено приклад графіка функції розподілу  $F(x)$  для  $n = 5$ .

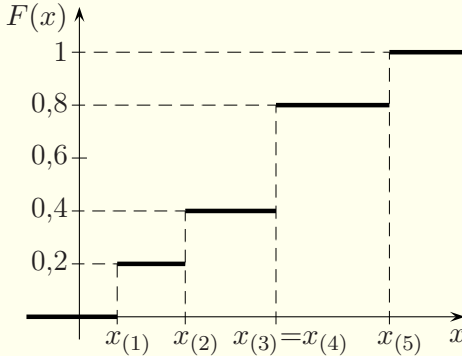


Рис. 9.2.

Для моделювання цієї випадкової величини скористаємось методом оберненої функції (§ 8.4). Згідно з цим методом потрібно знайти таке число  $i$ , для якого функція розподілу  $F(x)$  задовольняє формулу  $F(x_{(i-1)}) \leq r < F(x_{(i)})$ , де  $r \in [0, 1)$ ,  $F(x_{(0)}) = 0$ . Якщо функція  $F(x)$  подається формулою (9.4), то ми маємо:

$$\begin{aligned} \frac{i-1}{n} \leq r < \frac{i}{n} &\Rightarrow i \leq rn + 1 < i + 1 \Rightarrow \\ i &= [rn] + 1, \end{aligned} \quad (9.5)$$

де квадратні дужки позначають цілу частину числа.

Отже, для моделювання дискретної випадкової величини, функція розподілу якої подається формулою (9.4), спочатку генерують випадкове число  $r$ , рівномірно розподілене на півінтервалі  $[0, 1)$ , а потім виводять  $x_{(i)}$ , де число  $i$  обчислюється за формулою (9.5).

**Приклад 1.** Обчислити два числових значення дискретної випадкової величини з емпіричним розподілом, що відповідає вибірці 6, 5, 8, 4, 3, 4, 7, 3, 4, якщо генератор випадкових чисел генерує значення 0,869 і 0,236.

**Розв'язання.** Кількість чисел у вибірці  $n = 9$ . Відсортуємо вибірку за зростанням: 3, 3, 4, 4, 5, 6, 7, 8. Виконуємо обчислення за формулою (9.5). Для  $r = 0,869$  маємо:

$$i = [9 \cdot 0,869] + 1 = 8, \quad x = 7.$$

Для  $r = 0,236$  маємо:

$$i = [9 \cdot 0,236] + 1 = 3, \quad x = 4.$$

**Відповідь:** 7, 4.

При використанні мови імітаційного моделювання GPSS обчислення за формулою (9.5) здійснювати не потрібно, бо дії виконуються автоматично при використанні функції, яка описується командою FUNCTION (§ 6.13). У дискретному випадку її записують так:

$N$  FUNCTION RN $m$ ,D $n$

$$\frac{1}{n}, x_{(1)}/\frac{2}{n}, x_{(2)}/\frac{3}{n}, x_{(3)}/\dots/\frac{n-1}{n}, x_{(n-1)}/1, x_{(n)}$$

де  $N$  — ім'я функції,  $n$  — кількість чисел у вибірці даних спостережень,  $m$  — номер генератора випадкових чисел, а  $x_{(1)}$ ,  $x_{(2)}$ , ...,  $x_{(n)}$  — відсортовані за зростанням дані спостережень. Усі дробни при визначенні функції мають бути обчисленими.

**Приклад 2.** На мові імітаційного моделювання GPSS створити дискретну функцію для моделювання емпіричного розподілу, що відповідає вибірці 6, 5, 8, 4, 3, 4, 7, 3, 4.

**Розв'язання.** Кількість чисел у вибірці  $n = 9$ . Відсортуємо вибірку за зростанням: 3, 3, 4, 4, 5, 6, 7, 8. Функцію запишемо, наприклад, так:

ddd FUNCTION RN13,D9

$$\frac{1}{9}, 3/\frac{2}{9}, 3/\frac{3}{9}, 4/\frac{4}{9}, 4/\frac{5}{9}, 4/\frac{6}{9}, 5/\frac{7}{9}, 6/\frac{8}{9}, 7/1, 8$$

Якщо в дискретному розподілі зустрічаються пари з однаковими другими елементами, то перші з них можна пропускати. Тому після наближеного обчислення дробів маємо:

ddd FUNCTION RN13,D6

$$0.222, 3/0.556, 4/0.667, 5/0.778, 6/0.889, 7/1, 8 \quad \blacksquare$$

### § 9.4. Емпірично розподілені випадкові величини та їхнє моделювання у випадку згрупованих даних

Нехай випадкова величина є неперервною, а дані — згрупованими, тобто ми не знаємо значень окремих величин  $x_j$ . Припустимо, що  $n$  величин  $x_j$  згруповані в  $k$  суміжних півінтервалів  $[a_0, a_1)$ ,  $[a_1, a_2)$ ,  $[a_2, a_3)$ ,  $\dots$ ,  $[a_{k-1}, a_k)$ , причому  $i$ -й півінтервал ( $i = 1, 2, \dots, k$ ) містить  $n_i$  спостережень, а  $n_1 + n_2 + \dots + n_k = n$ .

Визначимо неперервну кусково-лінійну функцію розподілу  $G(x)$  так, щоб

$$G(a_0) = 0, \quad G(a_i) = \frac{n_1 + n_2 + \dots + n_i}{n}, \quad i = 1, 2, \dots, k. \quad (9.6)$$

Знайдемо рівняння прямих, які проходять через точки  $(a_i, G(a_i))$  і  $(a_{i+1}, G(a_{i+1}))$ ,  $i = 0, 1, 2, \dots, k - 1$ :

$$\begin{aligned} \frac{y - G(a_i)}{G(a_{i+1}) - G(a_i)} &= \frac{x - a_i}{a_{i+1} - a_i} \Rightarrow \\ y &= G(a_i) + \frac{x - a_i}{a_{i+1} - a_i} (G(a_{i+1}) - G(a_i)). \end{aligned}$$

Тоді

$$G(x) = \begin{cases} 0, & x < a_0, \\ G(a_i) + \frac{x - a_i}{a_{i+1} - a_i} (G(a_{i+1}) - G(a_i)), & a_i \leq x < a_{i+1}, \\ 1, & x \geq a_k. \end{cases} \quad i = 0, 1, \dots, k - 1, \quad (9.7)$$

Формули для моделювання цього емпіричного розподілу також виведемо методом оберненої функції. З рівності  $G(x) = r$ ,  $r \in [0, 1)$ , маємо:

$$\begin{aligned} G(a_i) + \frac{x - a_i}{a_{i+1} - a_i} (G(a_{i+1}) - G(a_i)) &= r, \quad a_i \leq x < a_{i+1}, \\ i &= 0, 1, \dots, k - 1. \end{aligned}$$

Звідси

$$x = a_i + \frac{r - G(a_i)}{G(a_{i+1}) - G(a_i)}(a_{i+1} - a_i), \quad (9.8)$$

причому число  $i$  повинно бути таким, щоб виконувалась подвійна нерівність

$$G(a_i) \leq r < G(a_{i+1}). \quad (9.9)$$

Отже, для моделювання неперервної випадкової величини з емпіричним розподілом у випадку згрупованих даних генерують випадкове число  $r$ , рівномірно розподілене на півінтервалі  $[0, 1)$ , потім методом перебору знаходять таке число  $i$ , щоб виконувалась формула (9.9), де  $G(a_i)$  задаються формулами (9.6), і, насамкінець, обчислюють число  $x$  за формулою (9.8).

Треба мати на увазі, що на проміжках  $[a_i, a_{i+1})$ , для яких  $n_i = 0$ , за формулами (9.6), (9.9), (9.8) не буде згенероване жодне значення, бо на цих проміжках має місце рівність  $G(a_i) = G(a_{i+1})$  замість нерівності  $G(a_i) < G(a_{i+1})$ . Останнє зауваження вказує ще на один недолік емпіричного розподілу. За межами проміжку  $[a_0, a_k)$  також не буде згенеровано жодного значення.

**Приклад 1.** Обчислити два числових значення неперервної випадкової величини з емпіричним розподілом, що відповідає згрупованій вибірці з наступної таблиці.

$[a_i, a_{i+1})$	[3, 4)	[4, 5)	[5, 7)	[7, 8)	[8, 9)	[9, 10)
$n_i$	1	2	1	0	3	1

Вважати, що генератор випадкових чисел генерує значення 0,869 і 0,236.

**Розв'язання.** Кількість інтервалів  $k = 6$ , а кількість чисел у вибірці  $n = 1 + 2 + 1 + 0 + 3 + 1 = 8$ . Для зручності, користуючись формулами (9.6), заповнимо наступну таблицю (останній рядок містить наближені числа).

$i$	0	1	2	3	4	5	6
$a_i$	3	4	5	7	8	9	10
$G(a_i)$	0	$\frac{1}{8}$	$\frac{3}{8}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{7}{8}$	1
$G(a_i)$	0	0,125	0,375	0,5	0,5	0,875	1

Для  $r = 0,869$  подвійна нерівність (9.9) виконується ( $0,5 \leq 0,869 < 0,875$ ) для значення  $i = 4$ . Обчислюємо  $x$  за формулою (9.8):

$$x = 8 + \frac{0,869 - 0,5}{0,875 - 0,5}(9 - 8) = 8,984.$$

Для  $r = 0,236$  подвійна нерівність (9.9) виконується ( $0,125 \leq 0,236 < 0,375$ ) для значення  $i = 1$ . Знаходимо  $x$  за формулою (9.8):

$$x = 4 + \frac{0,236 - 0,125}{0,375 - 0,125}(5 - 4) = 4,444.$$

**Відповідь:** 8,984, 4,444.

При використанні мови імітаційного моделювання GPSS шукати таке значення  $i$ , щоб виконувалась нерівність (9.9), і виконувати обчислення за формулою (9.8) не потрібно, бо ці дії виконуються автоматично при використанні функції, яка описується командою FUNCTION (§ 6.13). У даному випадку її записують так:

$$N \text{ FUNCTION RN}m, C(k+1) \\ 0, a_0 / \frac{n_1}{n}, a_1 / \frac{n_1+n_2}{n}, a_2 / \dots / \frac{n_1+n_2+\dots+n_{k-1}}{n}, a_{k-1} / 1, a_k$$

Замість  $N$  вказують ім'я функції, а замість  $m$ ,  $n$ ,  $a_0$ ,  $a_i$ ,  $n_i$ ,  $i = 1, 2, \dots, k$ , ставлять відповідні числа. Усі дробу тут також потрібно обчислити.

**Приклад 2.** На мові імітаційного моделювання GPSS створити неперервну функцію для моделювання емпіричного розподілу, що відповідає згрупованій вибірці з наступної таблиці.

$[a_i, a_{i+1})$	[3, 4)	[4, 5)	[5, 7)	[7, 8)	[8, 9)	[9, 10)
$n_i$	1	2	1	0	3	1

**Розв'язання.** Кількість півінтервалів  $k = 6$ , а кількість чисел у вибірці  $n = 8$ . Функцію запишемо, наприклад, так:

```
zgrv FUNCTION RN25,C7
0,3/1/8,4/3/8,5/1/2,7/1/2,8/7/8,9/1,10
```

Після наближеного обчислення дробів маємо:

```
zgrv FUNCTION RN25,C7
0,3/0.125,4/0.375,5/0.5,7/0.5,8/0.875,9/1,10 ■
```

Нехай ми маємо дискретну випадкову величину, про яку відомо те, що значення  $x_i$  у вибірці зустрічається  $n_i$  разів,  $i = 1, 2, \dots, k$ . Припустимо, що числа  $x_i$  відсортовані за зростанням:  $x_1 < x_2 < \dots < x_k$ . Нехай  $n = n_1 + n_2 + \dots + n_k$ . Тоді для функції розподілу використовується відома з курсу математичної статистики формула

$$G(x) = \begin{cases} 0, & x < x_1, \\ \frac{n_1+n_2+\dots+n_i}{n}, & x_i \leq x < x_{i+1}, \quad i = 1, 2, \dots, k-1, \\ 1, & x \geq x_k. \end{cases} \quad (9.10)$$

В основі моделювання цієї випадкової величини знову лежить метод оберненої функції. Для моделювання дискретної випадкової величини, функція розподілу якої подається формулою (9.10), спочатку генерують випадкове число  $r$ , рівномірно розподілене на півінтервалі  $[0, 1)$ , потім методом перебору знаходять таке число  $i$ , щоб виконувалась подвійна нерівність

$$\frac{n_1 + n_2 + \dots + n_{i-1}}{n} \leq r < \frac{n_1 + n_2 + \dots + n_i}{n}, \quad (9.11)$$

і виводять число  $x_i$ .

**Приклад 3.** Обчислити два числових значення дискретної випадкової величини з емпіричним розподілом, що відповідає згрупованій вибірці з наступної таблиці.

$x_i$	6	3	5	2	7	10
$n_i$	1	2	1	4	3	1

Вважати, що генератор випадкових чисел генерує значення 0,869 і 0,236.

**Розв'язання.** Кількість значень дискретної величини  $k = 6$ , а кількість чисел у вибірці  $n = 1 + 2 + 1 + 4 + 3 + 1 = 12$ . Відсортуємо вибірку за зростанням і заповнимо наступну таблицю (останній рядок містить наближені числа).

$i$	0	1	2	3	4	5	6
$x_i$		2	3	5	6	7	10
$n_i$		4	2	1	1	3	1
$\frac{n_1+n_2+\dots+n_i}{n}$	0	$\frac{1}{3}$	$\frac{1}{2}$	$\frac{7}{12}$	$\frac{2}{3}$	$\frac{11}{12}$	1
$\frac{n_1+n_2+\dots+n_i}{n}$	0	0,333	0,5	0,583	0,667	0,917	1

Для  $r = 0,869$  подвійна нерівність (9.11) виконується ( $0,667 \leq 0,869 < 0,917$ ) для значення  $i = 5$ . Отже,  $x = 7$ .

Для  $r = 0,236$  подвійна нерівність (9.11) виконується ( $0 \leq 0,236 < 0,333$ ) для значення  $i = 1$ . Отже,  $x = 2$ .

**Відповідь:** 7, 2.

При використанні мови імітаційного моделювання GPSS функцію для моделювання цієї випадкової величини описують так:

$N$  FUNCTION RN $m$ ,D $k$

$\frac{n_1}{n}, x_1/\frac{n_1+n_2}{n}, x_2/\frac{n_1+n_2+n_3}{n}, x_3/\dots/\frac{n_1+n_2+\dots+n_{k-1}}{n}, x_{k-1}/1, x_k$

Замість  $N$  вказують ім'я функції, а замість  $m, n, n_i, x_i, i = 1, 2, \dots, k$ , ставлять відповідні числа. Усі дробу тут також потрібно обчислити.

**Приклад 4.** На мові імітаційного моделювання GPSS створити дискретну функцію для моделювання емпіричного розподілу, що відповідає згрупованій вибірці з наступної таблиці.

$x_i$	6	3	5	2	7	10
$n_i$	1	2	1	4	3	1

**Розв'язання.** Кількість різних чисел  $k = 6$ , а кількість чисел у вибірці  $n = 12$ . Упорядкуємо вибірку за зростанням:

$x_i$	2	3	5	6	7	10
$n_i$	4	2	1	1	3	1

Функцію записуємо, наприклад, так:

```
zdv FUNCTION RN34,D6
1/3,2/1/2,3/7/12,5/2/3,6/11/12,7/1,10
```

Після наближеного обчислення дробів маємо:

```
zdv FUNCTION RN34,D6
0.333,2/0.5,3/0.583,5/0.667,6/0.917,7/1,10 ■
```

## § 9.5. Методи оцінки незалежності вибірки

Іноді спостереження, отримані протягом певного періоду часу, є залежними. Наприклад, припустимо, що  $x_1, x_2, \dots, x_n$  є температурою повітря, виміряною з інтервалом в одну годину. Ми не можемо розраховувати, що ці дані є незалежними, оскільки значення температури, близькі по часу, будуть пов'язаними між собою. Аналогічно, якщо вибірка  $x_1, x_2, \dots, x_n$  містить час перебування замовлень у черзі в системі масового обслуговування з інтенсивністю надходження вимог, близькою до швидкості обслуговування, то ця вибірка теж є залежною.

Оцінки параметрів розподілу по методу максимальної правдоподібності і перевірки гіпотез за критерієм  $\chi^2$  вимагають незалежності спостережень, тобто спостереження  $x_1, x_2, \dots, x_n$  мають бути незалежною (випадковою) вибіркою з деякого розподілу, що лежить в її основі. Якщо припущення про незалежність вибірки не виконується, то ці статистичні методи не є адекватними (тобто використовувати їх можна, але важко сказати, наскільки отримані по ним результати будуть близькими до справжніх). Але, навіть якщо дані не є незалежними, все одно можна використовувати такі евристичні методи, як гістограми.

Розглянемо два графічні методи неформальної оцінки незалежності даних спостережень  $x_1, x_2, \dots, x_n$ , які є впорядкованими за часом їхнього збору.

*Діаграма розкиду* для вибірки  $x_1, x_2, \dots, x_n$  — це графік точок з координатами  $(x_i, x_{i+1})$ ,  $i = 1, 2, \dots, n - 1$ . Якщо отримані точки є розкиданими у деякій прямокутній області площини, то вибірку вважають незалежною; якщо ці точки розташовуються вздовж деякої лінії на площині, то вибірку вважають залежною.

**Приклад 1.** З допомогою діаграми розкиду перевірити на незалежність вибірку:  $-0,1824; -0,01079; 0,9981; 0,5227; -0,2657; 0,4019; 0,2828; 1,454; 0,5447; 1,124; -0,04849; -0,1606; 0,01237; -0,1720; 0,1703; 0,3347; -0,03626; 0,2109; 0,4173; 0,7286; -0,1929; -0,2704; 0,4542; -0,1968; 0,4555; 0,9337; 0,04769; -0,3455; 0,3980; 0,3663$ .

**Розв'язання.** Діаграму розкиду для цієї вибірки зображено на рис. 9.3. З діаграми робимо висновок про незалежність вибірки. ■

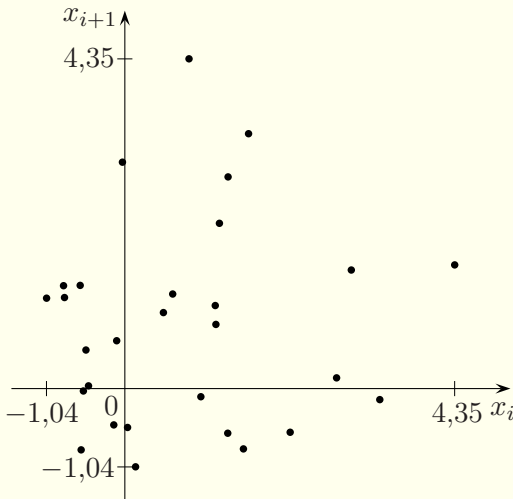


Рис. 9.3.

**Приклад 2.** З допомогою діаграми розкиду перевірити на незалежність вибірку: 0; 3,03; 3,1; 3,55; 2,5; 3,56; 4,39; 5,43; 5,93; 5,44; 0,8; 0,01; 0,77; 0,56; 0; 1,2; 1,39; 0,92; 0,13; 0; 0,21; 0,52; 0,62; 0,25; 0; 0; 0; 0,38; 0,53; 0; 0,16; 0; 0; 1,48; 2,7; 1,3; 2,19; 0,88; 0; 1,09.

**Розв’язання.** Діаграму розкиду для цієї вибірки зображено на рис. 9.4. Оскільки більшість точок зосереджено приблизно вздовж відрізка прямої, дана вибірка є залежною. ■

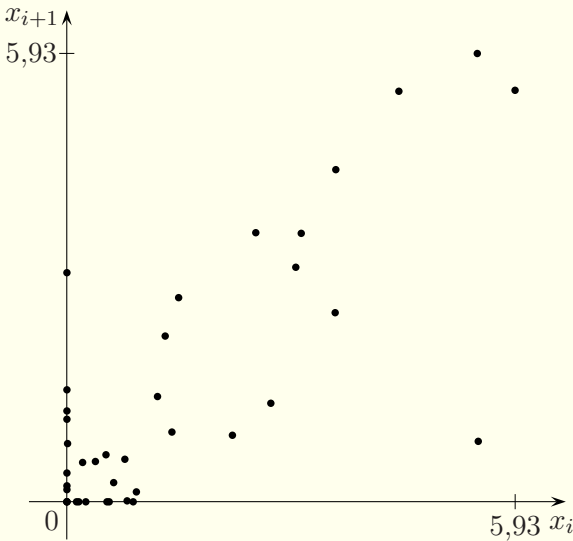


Рис. 9.4.

Другий спосіб оцінки незалежності вибірки полягає в побудові *кореляційного графіка*. Для цього обчислюють вибірко-ву кореляцію  $\rho_j$  за формулами

$$\rho_j = \frac{c_j}{D}, \quad c_j = \frac{1}{n-j} \sum_{i=1}^{n-j} (x_i - \bar{x})(x_{i+j} - \bar{x}), \quad j = 1, 2, \dots, n-1, \tag{9.12}$$

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i, \quad D = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2, \tag{9.13}$$

і будують графік залежності  $\rho_j$  від  $j$ . Оскільки для великих індексів  $j$  у сумі для  $c_j$  буде надто мало доданків, що призведе до неточностей, кореляційний графік будують лише для значно менших від  $n$  значень індексів  $j$ . Для виконання обчислень за формулами (9.12), (9.13) і побудови кореляційного графіка зручно користуватись комп'ютерною технікою.

Для вибіркової кореляції виконується оцінка  $-1 \leq \rho_j \leq 1$ . Якщо спостереження  $x_1, x_2, \dots, x_n$  є незалежними, то значення  $\rho_j$  мають бути близькими до нуля для всіх  $j = 1, 2, \dots, n-1$ . З того, що значення  $\rho_j$  суттєво відрізняються від нуля, явно випливає, що величини  $x_i$  не є незалежними. Якщо на кореляційному графіку всі точки є близькими до осі абсцис, то вважаємо, що вибірка є незалежною, а якщо окремі точки суттєво відхиляються від осі абсцис (особливо, якщо вони утворюють монотонну послідовність), то вважаємо, що вибірка є залежною.

**Приклад 3.** З допомогою кореляційного графіка перевірити на незалежність вибірку:  $-0,1824; -0,01079; 0,9981; 0,5227; -0,2657; 0,4019; 0,2828; 1,454; 0,5447; 1,124; -0,04849; -0,1606; 0,01237; -0,1720; 0,1703; 0,3347; -0,03626; 0,2109; 0,4173; 0,7286; -0,1929; -0,2704; 0,4542; -0,1968; 0,4555; 0,9337; 0,04769; -0,3455; 0,3980; 0,3663$ .

**Розв'язання.** Обчислюємо середнє арифметичне і дисперсію вибірки за формулами (9.13), а вибіркoву кореляцію за формулами (9.12). Отримані числові значення наведені в наступній таблиці.

$\bar{x}$	0,2659	$\rho_3$	-0,1198	$\rho_7$	0,0481	$\rho_{11}$	-0,0028
$D$	0,2058	$\rho_4$	-0,0901	$\rho_8$	-0,2532	$\rho_{12}$	0,0083
$\rho_1$	0,0795	$\rho_5$	-0,0649	$\rho_9$	-0,1863	$\rho_{13}$	-0,072
$\rho_2$	-0,1362	$\rho_6$	0,0612	$\rho_{10}$	0,2701	$\rho_{14}$	-0,3784

На рис. 9.5 побудовано кореляційний графік. Оскільки мінімальне значення вибіркової кореляції дорівнює  $-0,3784$ , максимальне  $-0,2701$ , а точки розкидано на графіку випадко-

вим чином, то цю вибірку можна вважати незалежною. ■

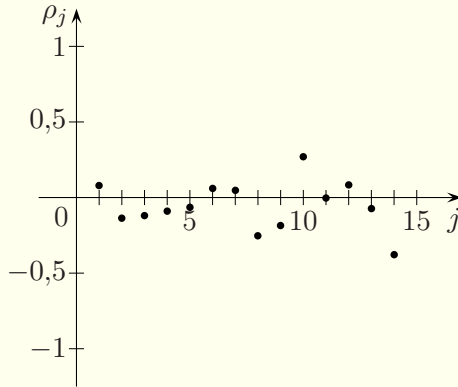


Рис. 9.5.

**Приклад 4.** З допомогою кореляційного графіка перевірити на незалежність вибірку: 0; 3,03; 3,1; 3,55; 2,5; 3,56; 4,39; 5,43; 5,93; 5,44; 0,8; 0,01; 0,77; 0,56; 0; 1,2; 1,39; 0,92; 0,13; 0; 0,21; 0,52; 0,62; 0,25; 0; 0; 0; 0,38; 0,53; 0; 0,16; 0; 0; 1,48; 2,7; 1,3; 2,19; 0,88; 0; 1,09.

**Розв’язання.** На рис. 9.6 наведено кореляційний графік. Оскільки мінімальне значення вибіркової кореляції дорівнює  $-0,64$ , а максимальне  $-0,77$ , то ця вибірка є залежною. ■

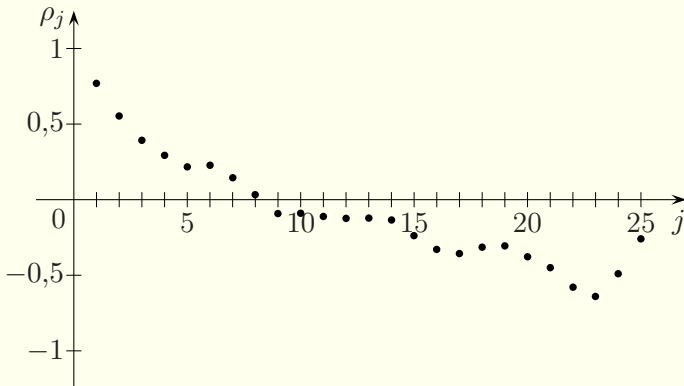


Рис. 9.6.

## § 9.6. Вибір сім'ї теоретичних розподілів з допомогою гістограми

Вибір теоретичного розподілу ймовірностей здійснюють у три етапи: висування гіпотези щодо сім'ї розподілів, яка буде наближати дані спостережень, визначення параметрів розподілу і перевірка правильності гіпотези про характер розподілу. Цей і шість наступних параграфів присвячені послідовному вивченню всіх трьох етапів.

На першому етапі вибору теоретичного розподілу ймовірностей потрібно знайти придатну загальну сім'ю розподілів (експоненціальний розподіл, нормальний розподіл, біноміальний розподіл чи якийсь інший) на основі їхніх форм, не цікавлячись поки що конкретними значеннями параметрів розподілу.

Для вибору сім'ї розподілів використовують три основні методи — побудову і порівняння гістограми з графіками функцій щільності (чи функцій імовірностей) відомих теоретичних розподілів, аналіз ролі випадкової величини в системі та аналіз підсумкової статистики. У цьому і наступному параграфі ми зупинимось на першому методі, двом іншим методам присвячено § 9.8 і § 9.9.

Гістограма дозволяє вибирати розподіли, перш за все, у випадку неперервних випадкових величин. Саме цей випадок ми спочатку розглянемо. Для неперервних даних гістограма є по суті графічною оцінкою графіка функції щільності розподілу ймовірностей, відповідною даним спостережень  $x_1, x_2, \dots, x_n$ .

Для створення *гістограми* розіб'ємо область даних спостережень на  $k$  півінтервалів  $[b_0, b_1), [b_1, b_2), \dots, [b_{k-1}, b_k)$ . Всі півінтервали повинні мати однакову довжину  $\Delta b = b_j - b_{j-1}$ . Нехай  $h_j$  — частка тих величин  $x_i$ , які входять в  $j$ -й інтервал

$[b_{j-1}, b_j)$ ,  $j = 1, 2, \dots, k$ . Тепер визначимо функцію

$$h(x) = \begin{cases} 0, & x < b_0, \\ h_j, & b_{j-1} \leq x < b_j, \quad j = 1, 2, \dots, k, \\ 0, & x \geq b_k. \end{cases}$$

Гістограма складається з  $k$  стовпчиків, висота яких визначається кусково-сталюю функцією  $h(x)$ . Площа, зайнята гістограмою, дорівнює  $\Delta b$ . При побудові гістограми замість частки  $h_j$  можна брати величину  $n_j$  — кількість тих величин  $x_i$ , які входять у півінтервал  $[b_{j-1}, b_j)$ . У такому разі площа, зайнята гістограмою, дорівнюватиме величині  $n \cdot \Delta b$ . При побудові гістограми масштаб по осі абсцис і осі ординат беруть зазвичай різний. Приклад гістограми наведено на рис. 9.16 (с. 317).

Гістограму порівнюють з графіками щільності різних розподілів тільки на основі їхніх форм без врахування масштабу і відмінностей розташування. Це дозволяє вибрати той теоретичний розподіл, графік функції щільності якого найбільше нагадує гістограму.

Гістограма є дуже потужним методом вибору сім'ї теоретичних розподілів для неперервних випадкових величин. Однак основною проблемою при побудові гістограми є неоднозначність вибору числа  $k$  — кількості інтервалів у гістограмі.

Число  $k$  рекомендується вибирати за правилом Стерджеса

$$k = [\log_2 n] + 1, \quad (9.14)$$

де квадратні дужки позначають цілу частину числа, проте правило Стерджеса не завжди гарантує оптимальність гістограми. Можна також брати різні значення  $k$  і вибирати з кількох варіантів найкращу гістограму. Якщо взяти надто велике число  $k$ , то гістограма матиме «поламану» форму за рахунок випадкових відхилень. Якщо ж число  $k$  є дуже малим, то гістограма матиме «блочну» будову, зокрема може бути пропущена або розмита вершина графіка функції щільності.

Гістограми для різних значень  $k$  та графіки функцій щільності для різних значень параметрів зручно будувати з допомогою систем комп'ютерної математики, таких як *Mathia*, *Maple* та інші, або з допомогою спеціалізованих статистичних пакетів.

Для дискретних випадкових величин гістограма дозволяє вибрати той дискретний розподіл, графік функції ймовірностей якого нагадує гістограму. У цьому випадку  $h_j$  — частка ( $n_j$  — кількість) величин, рівних  $x_j$ . Графіки функцій імовірностей різних розподілів є подібними (див., наприклад, [18]), тому використання гістограми у дискретному випадку є малоефективним.

### § 9.7. Графіки функцій щільності розподілів імовірностей

Функції щільності експоненціального, нормального, логнормального, трикутного розподілів і розподілу Вейбулла наведені у § 8.6. Експоненціальний розподіл і розподіл Ерланга є окремими випадками більш загального гамма-розподілу, функція щільності якого подається формулою

$$f(x) = \begin{cases} 0, & x \leq 0, \\ \frac{\lambda^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\lambda x}, & x > 0, \end{cases} \quad (9.15)$$

де  $\Gamma(\alpha) = \int_0^\infty t^{\alpha-1} e^{-t} dt$  — гамма-функція,  $\alpha$  — коефіцієнт форми,  $\lambda = \frac{1}{\beta}$ ,  $\beta$  — коефіцієнт стиснення,  $\alpha > 0$ ,  $\lambda > 0$ . При натуральних  $\alpha$  з формули (9.15) отримується формула для функції щільності розподілу Ерланга, а при  $\alpha = 1$  — експоненціального розподілу.

Функція щільності бета-розподілу подається формулою

$$f(x) = \begin{cases} 0, & x \leq 0 \text{ або } x \geq 1, \\ \frac{x^{\alpha_1-1} \cdot (1-x)^{\alpha_2-1}}{B(\alpha_1, \alpha_2)}, & 0 < x < 1, \end{cases}$$

де  $B(\alpha_1, \alpha_2) = \int_0^1 t^{\alpha_1-1} (1-t)^{\alpha_2-1} dt$  — бета-функція,  $\alpha_1, \alpha_2$  — параметри форми,  $\alpha_1 > 0, \alpha_2 > 0$ . Бета-розподіл дає випадкову величину  $\xi$  на проміжку  $[0, 1]$ . Випадкову величину  $\eta$  з розподілом тієї самої форми на проміжку  $[a, b]$  отримують шляхом трансформації  $\eta = a + (b-a)\xi$ . Тому часто вважають, що бета-розподіл задається на довільному проміжку  $[a, b]$ .

Наведемо тепер типові форми графіків функцій щільності найважливіших розподілів неперервних випадкових величин: рис. 9.7 — рівномірний розподіл на проміжку  $[a, b]$ , рис. 9.8 — нормальний розподіл з середнім значенням  $\mu$  і середньоквадратичним відхиленням  $\sigma$ , рис. 9.9 — гамма-розподіли з різними значеннями параметрів  $\alpha$  і  $\lambda$  (графіку 1 відповідають параметри  $\alpha = 1, \lambda = 2$ , тобто експоненціальний розподіл з параметром  $\lambda = 2$ , графіку 2 — параметри  $\alpha = 2, \lambda = 2$ , тобто розподіл Ерланга другого порядку з параметром  $\lambda = 2$ , графіку 3 — параметри  $\alpha = 3,5, \lambda = 2$ ), рис. 9.10 — розподіли Вейбулла з різними значеннями параметрів  $\alpha$  і  $\beta$  (графіку 1 відповідають параметри  $\alpha = 1, \beta = 2/3$ , тобто експоненціальний розподіл з параметром  $\lambda = 3/2$ , графіку 2 — параметри  $\alpha = 2, \beta = 1$ , графіку 3 — параметри  $\alpha = 7, \beta = 2$ ), рис. 9.11 — логнормальні розподіли з різними значеннями параметрів  $\mu$  і  $\sigma$  (графіку 1 відповідають параметри  $\mu = 0,1, \sigma = 1,9$ , графіку 2 — параметри  $\mu = -0,2, \sigma = 1$ , графіку 3 — параметри  $\mu = 2, \sigma = 2$ ), рис. 9.12, 9.13 — бета-розподіли з різними значеннями параметрів  $\alpha_1$  і  $\alpha_2$  (графіку 1 рис. 9.12 відповідають параметри  $\alpha_1 = 1,5, \alpha_2 = 4$ , графіку 2 рис. 9.12 — параметри  $\alpha_1 = 5, \alpha_2 = 5$ , графіку 3 рис. 9.12 — параметри  $\alpha_1 = 3, \alpha_2 = 2$ , графіку 1 рис. 9.13 відповідають параметри  $\alpha_1 = 2, \alpha_2 = 1$ , графіку 2 рис. 9.13 — параметри  $\alpha_1 = 0,95, \alpha_2 = 1,5$ , графіку 3 рис. 9.13 — параметри  $\alpha_1 = 0,88, \alpha_2 = 0,9$ ), рис. 8.5 — трикутний розподіл.

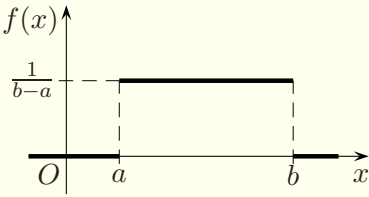


Рис. 9.7.

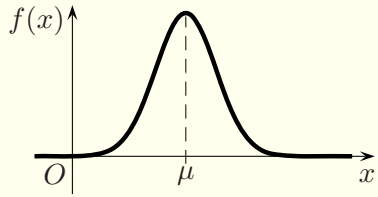


Рис. 9.8.

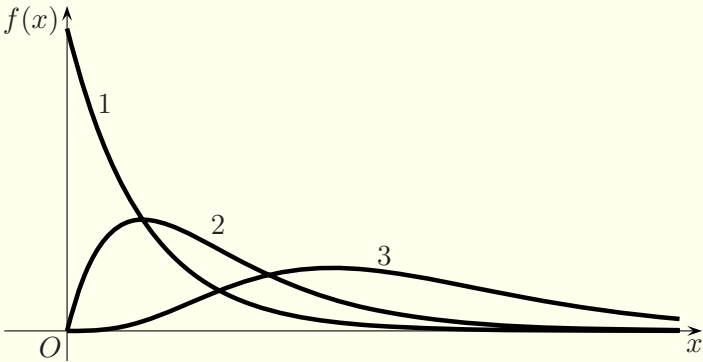


Рис. 9.9.

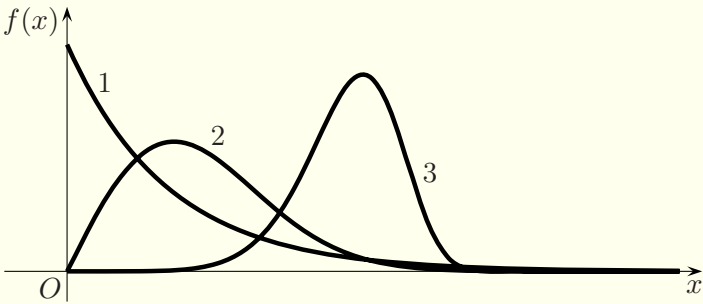


Рис. 9.10.

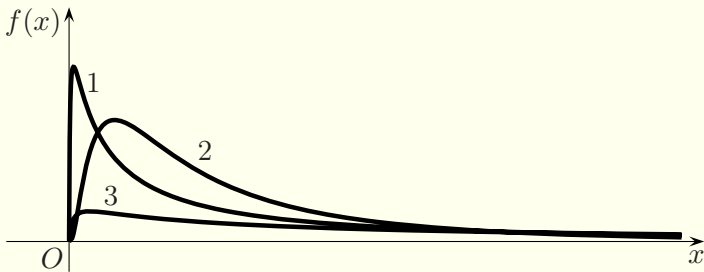


Рис. 9.11.

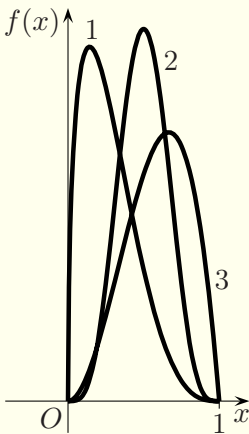


Рис. 9.12.

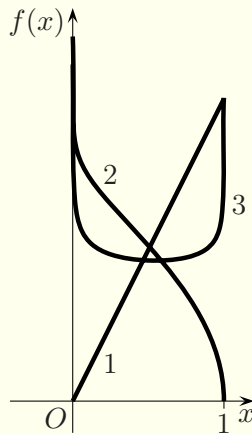


Рис. 9.13.

З графіками функцій щільності багатьох інших розподілів, які використовуються при моделюванні, можна ознайомитись у книзі [18].

### § 9.8. Аналіз ролі випадкової величини в системі

В окремих ситуаціях для того щоб вибрати моделюючий розподіл або принаймні виключити деякі розподіли, можна скористатись інформацією про роль випадкової величини в системі. Наприклад, якщо відомо, що замовлення надходять у систему масового обслуговування по одному з великої кількості незалежних джерел зі сталою інтенсивністю, то є підстави вважати, що вони утворюють стаціонарний пуассонівський

потік і, отже, для моделювання проміжків часу між ними треба використовувати експоненціальний закон розподілу (§ 5.3). Аналіз ролі випадкової величини в системі потрібно підтверджувати даними спостережень з допомогою гістограми (§ 9.6) або підсумкової статистики. Розглянемо тепер використання основних розподілів.

Рівномірний розподіл використовують як наближену модель випадкової величини, що змінюється між  $a$  і  $b$ , але про яку більше майже нічого не відомо.

Експоненціальний розподіл використовують для моделювання проміжків часу між надходженням вимог у систему при сталій інтенсивності, моделювання довільного стаціонарного пуассонівського потоку, а також для моделювання часу безвідмовної роботи пристрою. У деяких випадках вважають, що час виконання задачі (час обслуговування) теж можна моделювати експоненціальним розподілом.

Гамма-розподіл застосовують для моделювання часу виконання деякої задачі, наприклад, обслуговування клієнта чи ремонту машини. Цим розподілом можна також моделювати вхідний потік вимог і час безвідмовної роботи пристрою.

Розподіл Вейбулла використовують для моделювання часу виконання задачі, вхідного потоку вимог, часу безвідмовної роботи пристрою.

Нормальний розподіл використовують для моделювання відхилення від середнього значення, а також для моделювання випадкової величини, що є сумою великої кількості інших випадкових величин (на підставі центральної граничної теореми). Використовувати нормальний закон розподілу для моделювання часу виконання задачі чи проміжків часу між подіями не можна, бо випадкова величина з нормальним законом розподілу може дати від'ємний час.

Логнормальний розподіл використовують для моделювання часу виконання деякої задачі, вхідного потоку вимог, а також для моделювання випадкової величини, що є добутком

великої кількості інших величин (на основі центральної граничної теореми).

Трикутний розподіл використовують як наближену модель при недостатній кількості інформації.

Бета-розподіл використовують як наближену модель при недостатній кількості інформації і для моделювання випадкової частки.

Біноміальний розподіл дозволяє моделювати кількість успішних спроб у серії  $n$  незалежних випробувань, якщо ймовірність успіху кожного випробування дорівнює  $p$ , зокрема біноміальний розподіл можна використовувати для моделювання кількості пошкоджених товарів або кількості певних об'єктів у групі.

Геометричний розподіл використовують для моделювання кількості спроб до першої успішної спроби у послідовності незалежних випробувань, якщо ймовірність успіху кожного випробування дорівнює  $p$ .

Розподіл Пуассона дозволяє моделювати кількість подій у стаціонарному пуассонівському потоці, що виникають за деякий інтервал часу.

### § 9.9. Аналіз підсумкової статистики

У ряді випадків підсумкова статистика, обчислена для даних спостережень, дозволяє зробити припущення про характер теоретичного розподілу, придатного для моделювання випадкового процесу.

Позначимо дані спостережень через  $x_1, x_2, \dots, x_n$ . Відсортуємо їх за зростанням і перепозначимо через  $x_{(i)}$ :

$$x_{(1)} \leq x_{(2)} \leq \dots \leq x_{(n)}.$$

Мінімальне і максимальне значення  $x_{(1)}$  і  $x_{(n)}$  дозволяють наближено оцінити інтервал  $[x_{(1)}, x_{(n)}]$  зміни випадкової величини.

Роль середнього значення може виконувати середнє арифметичне

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

і медіана

$$x_{0,5} = \begin{cases} x_{(\frac{n+1}{2})}, & n - \text{непарне,} \\ \frac{x_{(\frac{n}{2})} + x_{(\frac{n}{2}+1)}}{2}, & n - \text{парне.} \end{cases}$$

Роль показника змінюваності може виконувати дисперсія

$$D = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2,$$

коефіцієнт варіації для неперервних даних

$$cv = \frac{\sqrt{D}}{\bar{x}}$$

і коефіцієнт Лексиса для дискретних даних

$$\tau = \frac{D}{\bar{x}}.$$

Показник асиметрії обчислюється за формулою:

$$\nu = \frac{1}{n \cdot D^{3/2}} \sum_{i=1}^n (x_i - \bar{x})^3.$$

З допомогою вказаних показників у деяких випадках можна висунути припущення щодо сім'ї розподілів.

Для симетричного неперервного розподілу (наприклад, нормального) середнє арифметичне значення і медіана збігаються, а асиметрія  $\nu = 0$ . Тому, якщо середнє арифметичне  $\bar{x}$  і медіана  $x_{0,5}$  даних спостережень є майже рівними, а

асиметрія  $\nu$  є близькою до нуля, то можна припустити, що розподіл, який лежить в основі цих даних спостереження, є нормальним.

Для експоненціального розподілу з довільною інтенсивністю  $\lambda$  коефіцієнт варіації дорівнює одиниці. Отже, якщо коефіцієнт варіації  $cv$  є близьким до одиниці, то можна припускати, що розподіл є експоненціальним. Коефіцієнт асиметрії для експоненціального розподілу має бути близьким до двох.

Якщо  $\nu > 0$ , то графік функції щільності скошений вправо (він може виглядати, наприклад, так, як це показано на рис. 9.14), а якщо  $\nu < 0$ , то графік функції щільності скошений вліво (може виглядати так, як це показано на рис. 9.15). Всі графіки з рис. 9.9, 9.11 і перші два з рис. 9.10 є скошеними вправо. Графік 3 з рис. 9.10 є скошеним вліво. На практиці частіше зустрічаються ситуації, коли графік функції щільності є скошеним вправо.

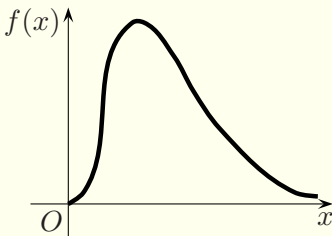


Рис. 9.14.

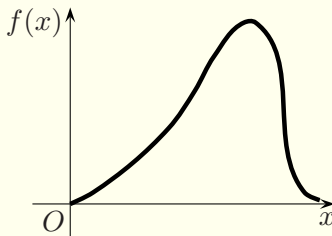


Рис. 9.15.

Якщо  $\nu > 0$  (графік функції щільності скошений вправо), а  $cv < 1$ , то може підходити логнормальний розподіл, гамма-розподіл з параметром форми  $\alpha > 1$  або розподіл Вейбулла з параметром форми  $1 < \alpha < 3,6$ . Якщо  $\nu > 0$ , а  $cv > 1$ , то краще застосовувати логнормальний розподіл, хоча допустимими є також гамма-розподіл і розподіл Вейбулла з параметром  $\alpha < 1$ . Якщо  $\nu < 0$ , то може підходити розподіл Вейбулла з параметром  $\alpha > 3,6$ .

Для дискретного розподілу коефіцієнт Лексиса виконує ту саму роль, що і коефіцієнт варіації для неперервного розподілу. Якщо  $\tau \approx 1$ , то розподіл може бути розподілом Пуассона. Якщо  $\tau < 1$ , то може бути біноміальний розподіл. Якщо  $\tau > 1$ , то більше підходить геометричний розподіл.

Підсумкову статистику зручно обчислювати з допомогою комп'ютерних програм і пакетів, наприклад, таких як Excel, Maxima, Maple, Mathematica, STATISTICA.

По можливості всі три методи вибору розподілу, розглянуті в останніх чотирьох параграфах, потрібно використовувати в комплексі. Перш за все треба побудувати і проаналізувати гістограму, доповнивши її аналіз дослідженням підсумкової статистики і ролі випадкової величини в системі.

**Приклад.** Визначити сім'ю розподілів, якщо дані спостережень мають вигляд: 0,626; 1,21; 7,53; 1,74; 0,695; 5,38; 0,750; 0,584; 3,17; 1,50; 0,969; 1,92; 1,16; 0,150; 0,491; 2,09; 3,44; 0,132; 0,234; 0,969.

**Розв'язання.** Кількість чисел у вибірці  $n = 20$ , причому випадкова величина є неперервною. Кількість стовпчиків у гістограмі знаходимо за формулою (9.14), отже,  $k = 5$ . Оскільки мінімальне значення у вибірці 0,15, а максимальне — 7,53, то гістограму потрібно будувати на проміжку  $[0,1; 7,6]$  (проміжок трохи розширено для зручності). Розіб'ємо проміжок  $[0,1; 7,6]$  на п'ять однакових частин (шириною 1,5), обчислимо кількість чисел, які потрапили в кожну частину, а результати наведемо в наступній таблиці.

$[b_{j-1}; b_j]$	$[0,1; 1,6]$	$[1,6; 3,1]$	$[3,1; 4,6]$	$[4,6; 6,1]$	$[6,1; 7,6]$
$n_j$	13	3	2	1	1

Гістограму (з різним масштабом по осі абсцис і осі ординат) наведено на рис. 9.16. Такій гістограмі можуть відповідати експоненціальний, логнормальний, гамма-розподіл, розподіл Вейбулла. Оскільки у проміжок  $[0; 0,8)$  — першу половину проміжку  $[0; 1,6)$  — потрапляє вісім чисел, тобто більшість чи-

сел з першого проміжку  $[0,1; 1,6)$ , то є підстави вважати, що найбільше підходить експоненціальний закон розподілу.

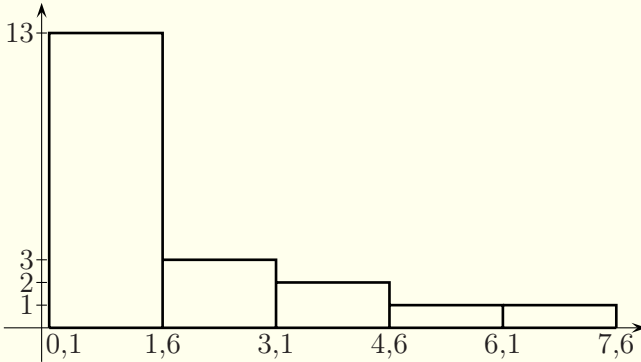


Рис. 9.16.

Обчисливши коефіцієнт варіації  $cv \approx 1,1$  і показник асиметрії  $\nu \approx 1,9$ , бачимо, що вони приблизно дорівнюють значенням, відповідним експоненціальному закону розподілу. Отже, є підстави вважати, що найкращим наближенням для даних спостережень є експоненціальний розподіл. ■

### § 9.10. Визначення параметрів теоретичних розподілів

Вибравши сім'ю теоретичного розподілу, відповідного даним спостережень, потрібно визначити його параметри. Є різні методи оцінки параметрів розподілу. Ми розглянемо один з найефективніших методів визначення параметрів розподілу — *метод максимальної правдоподібності*. Він полягає в тому, що будується функція правдоподібності. Для неперервного розподілу вона має вигляд

$$L(\theta) = f_{\theta}(x_1)f_{\theta}(x_2)\dots f_{\theta}(x_n) \quad (9.16)$$

і залежить від невідомого параметра  $\theta$  (скалярного чи векторного), де  $f_{\theta}(x)$  — функція щільності неперервного розподілу,

а  $x_1, x_2, \dots, x_n$  — дані спостережень. Векторний параметр  $\theta$  містить кілька скалярних параметрів  $\theta_1, \theta_2, \dots, \theta_k$ . Для дискретного розподілу функція правдоподібності має вигляд

$$L(\theta) = p_\theta(x_1)p_\theta(x_2) \dots p_\theta(x_n), \quad (9.17)$$

де  $p_\theta(x)$  — імовірність події.

Нам потрібно знайти такий параметр  $\hat{\theta}$  (або  $\hat{\theta}_1, \hat{\theta}_2, \dots, \hat{\theta}_k$ ), який би надавав максимального значення функції правдоподібності  $L(\theta)$ . Цей параметр  $\hat{\theta}$  (або  $\hat{\theta}_1, \hat{\theta}_2, \dots, \hat{\theta}_k$ ) буде шуканим параметром (параметрами) розподілу. Як видно з формули (9.17), у випадку дискретного розподілу максимуму функції правдоподібності надає те значення параметрів розподілу, за якого ймовірність отримання даних спостережень є максимальною.

Для зручності замість функції  $L(\theta)$  досліджують на екстремум функцію  $M(\theta) = \ln L(\theta)$ . Необхідними умовами екстремуму функції  $M(\theta)$  є рівність нулю частинних похідних цієї функції за всіма параметрами  $\theta_1, \theta_2, \dots, \theta_k$ . З необхідних умов екстремуму знаходять значення параметрів і доводять, що для цих значень параметрів функція  $M(\theta)$  справді має максимум.

**Приклад 1.** Визначити методом максимальної правдоподібності параметр експоненціального розподілу для даних спостережень  $x_1, x_2, \dots, x_n$ .

**Розв'язання.** Для експоненціального розподілу функція щільності  $f_\lambda(x) = \lambda e^{-\lambda x}$  для  $x \geq 0$ . Тому функція правдоподібності за формулою (9.16) для даних спостережень  $x_1, x_2, \dots, x_n$  набуває вигляду:

$$L(\lambda) = \lambda e^{-\lambda x_1} \cdot \lambda e^{-\lambda x_2} \cdot \dots \cdot \lambda e^{-\lambda x_n} = \lambda^n e^{-\lambda \sum_{i=1}^n x_i}.$$

Тоді

$$M(\lambda) = \ln L(\lambda) = n \ln \lambda - \lambda \sum_{i=1}^n x_i.$$

З необхідної умови екстремуму маємо:

$$\frac{dM}{d\lambda} = \frac{n}{\lambda} - \sum_{i=1}^n x_i = 0 \quad \Rightarrow \quad \lambda = \frac{n}{\sum_{i=1}^n x_i} = \frac{1}{\bar{x}},$$

де  $\bar{x}$  — середнє арифметичне даних спостережень.

Перевіримо тепер достатні умови екстремуму. Оскільки

$$\frac{d^2M}{d\lambda^2} = -\frac{n}{\lambda^2} < 0$$

для всіх  $\lambda \neq 0$ , то значення

$$\lambda = \frac{1}{\bar{x}} = \frac{n}{\sum_{i=1}^n x_i} \quad (9.18)$$

справді максимізує функцію максимальної правдоподібності і є шуканим параметром експоненціального розподілу. ■

**Приклад 2.** Визначити методом максимальної правдоподібності параметри нормального розподілу для даних спостережень  $x_1, x_2, \dots, x_n$ .

**Розв'язання.** Для нормального розподілу функція щільності

$$f_{(\mu, \sigma)}(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}.$$

Тому функція правдоподібності для даних спостережень  $x_1, x_2, \dots, x_n$  має вигляд:

$$L(\mu, \sigma) = \left( \frac{1}{\sqrt{2\pi}\sigma} \right)^n e^{-\frac{1}{2\sigma^2} \sum_{i=1}^n (x_i - \mu)^2}.$$

Тоді

$$M(\mu, \sigma) = \ln L(\mu, \sigma) = -n \ln \sqrt{2\pi} - n \ln \sigma - \frac{1}{2\sigma^2} \sum_{i=1}^n (x_i - \mu)^2.$$

З необхідних умов екстремуму для функції двох змінних маємо:

$$\begin{cases} \frac{\partial M}{\partial \mu} = \frac{1}{\sigma^2} \sum_{i=1}^n (x_i - \mu) = 0, \\ \frac{\partial M}{\partial \sigma} = -\frac{n}{\sigma} + \frac{1}{\sigma^3} \sum_{i=1}^n (x_i - \mu)^2 = 0 \end{cases} \Rightarrow \begin{cases} \mu = \frac{1}{n} \sum_{i=1}^n x_i = \bar{x}, \\ \sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2} = s, \end{cases}$$

де  $\bar{x}$  — середнє арифметичне даних спостережень, а  $s$  — середньоквадратичне відхилення даних спостережень.

Перевіримо тепер достатні умови екстремуму:

$$\begin{aligned} a_{11} &= \left. \frac{\partial^2 M}{\partial \mu^2} \right|_{\substack{\mu=\bar{x} \\ \sigma=s}} = -\frac{n}{s^2} < 0, \\ a_{12} &= \left. \frac{\partial^2 M}{\partial \sigma \partial \mu} \right|_{\substack{\mu=\bar{x} \\ \sigma=s}} = -\frac{2}{s^3} \sum_{i=1}^n (x_i - \bar{x}) = 0, \\ a_{22} &= \left. \frac{\partial^2 M}{\partial \sigma^2} \right|_{\substack{\mu=\bar{x} \\ \sigma=s}} = \frac{n}{s^2} - \frac{3}{s^4} \sum_{i=1}^n (x_i - \bar{x})^2 = -\frac{2n}{s^2}, \\ a_{11}a_{22} - a_{12}^2 &= \frac{2n^2}{s^4} > 0. \end{aligned}$$

Отже,

$$\mu = \frac{1}{n} \sum_{i=1}^n x_i = \bar{x}, \quad \sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2} = s$$

надають максимуму функції правдоподібності і є шуканими параметрами нормального розподілу. ■

**Приклад 3.** Визначити методом максимальної правдоподібності параметр геометричного розподілу для даних спостережень  $x_1, x_2, \dots, x_n$ .

**Розв'язання.** Для геометричного розподілу

$$P_p(k) = p(1-p)^{k-1}, \quad 0 < p < 1.$$

Тому функція правдоподібності за формулою (9.17) для даних спостережень  $x_1, x_2, \dots, x_n$  набуває вигляду:

$$L(p) = p(1-p)^{x_1-1} \cdot p(1-p)^{x_2-1} \cdot \dots \cdot p(1-p)^{x_n-1} = p^n (1-p)^{\sum_{i=1}^n x_i - n}.$$

Тоді

$$M(p) = \ln L(p) = n \ln p + \left( \sum_{i=1}^n x_i - n \right) \ln(1-p).$$

З необхідної умови екстремуму маємо:

$$\begin{aligned} \frac{dM}{dp} &= \frac{n}{p} - \frac{1}{1-p} \left( \sum_{i=1}^n x_i - n \right) = 0 \quad \Rightarrow \\ n - np &= p \left( \sum_{i=1}^n x_i - n \right) \quad \Rightarrow \quad p = \frac{n}{\sum_{i=1}^n x_i} = \frac{1}{\bar{x}}, \end{aligned}$$

де  $\bar{x}$  — середнє арифметичне даних спостережень.

Перевіримо тепер достатні умови екстремуму. Маємо:

$$\frac{d^2M}{dp^2} = -\frac{n}{p^2} - \frac{1}{(1-p)^2} \left( \sum_{i=1}^n x_i - n \right) < 0$$

для всіх  $0 < p < 1$ , бо  $x_i \geq 1$ ,  $i = 1, 2, \dots, n$ . Отже, значення

$$p = \frac{1}{\bar{x}} = \frac{n}{\sum_{i=1}^n x_i}$$

справді максимізує функцію максимальної правдоподібності і є шуканим параметром геометричного розподілу. ■

**Приклад 4.** Визначити розподіл, якщо дані спостережень мають вигляд: 0,626; 1,21; 7,53; 1,74; 0,695; 5,38; 0,750; 0,584; 3,17; 1,50; 0,969; 1,92; 1,16; 0,150; 0,491; 2,09; 3,44; 0,132; 0,234; 0,969.

**Розв'язання.** У прикладі з § 9.9 (с. 316) встановлено, що для моделювання цих даних спостережень найбільше підходить експоненціальний розподіл. Параметр  $\lambda$  експоненціального розподілу для цих даних спостережень згідно з формулою (9.18), виведеною методом максимальної правдоподібності у прикладі 1 цього параграфа, наближено дорівнює 0,5757. ■

### § 9.11. Графічні процедури перевірки придатності розподілу

Для перевірки гіпотези про придатність підбраного теоретичного розподілу ймовірностей для опису даних спостережень можна використовувати графічні (евристичні) процедури і статистичні критерії. Цей параграф присвячено таким графічним процедурам перевірки придатності розподілу, як порівняння графіка функції щільності розподілу ймовірностей і гістограми, частотне порівняння, аналіз графіка відмінностей між функціями розподілу. Наступний параграф містить метод перевірки гіпотез з допомогою критерію  $\chi^2$ .

Всі графічні процедури перевірки придатності розподілу зазвичай реалізують з допомогою комп'ютерної техніки. При цьому рекомендується використовувати універсальні системи комп'ютерної математики, такі як Maxima, Maple, Mathematica, або спеціалізовані статистичні пакети.

**Графік функції щільності розподілу ймовірностей поверх гістограми.** Для неперервних даних можна будувати графік функції  $S \cdot f(x)$  поверх гістограми (про побудову гістограми див. у § 9.6), де  $S$  — площа гістограми, а  $f(x)$  — знайдена функція щільності розподілу ймовірностей. Множення на

масштабний коефіцієнт  $S$  необхідне, оскільки площа, обмежена графіком функції  $f(x)$ , дорівнює 1, а площа, зайнята гистограмою, дорівнює ширині стовпця гистограми, помноженій на сумарну висоту всіх стовпців гистограми. При побудові гистограми з площею, рівною 1, множення функції щільності розподілу ймовірностей на коефіцієнт  $S$  не потрібне. Якщо форма графіка функції щільності нагадує гистограму, то теоретичний розподіл, відповідний даним спостережень, підібрано вдало. У протилежному випадку потрібно шукати інший теоретичний розподіл.

**Приклад 1.** З допомогою графіка функції щільності поверх гистограми перевірити гіпотезу про придатність експоненціального розподілу для моделювання таких даних спостережень: 0,626; 1,21; 7,53; 1,74; 0,695; 5,38; 0,750; 0,584; 3,17; 1,50; 0,969; 1,92; 1,16; 0,150; 0,491; 2,09; 3,44; 0,132; 0,234; 0,969.

**Розв'язання.** Параметр  $\lambda$  експоненціального розподілу для цих даних спостережень згідно з формулою (9.18), виведеною методом максимальної правдоподібності, наближено дорівнює 0,5757. Гистограма з п'ятьма стовпцями для заданих даних спостережень була побудована у прикладі з § 9.9 (рис. 9.16). Ширина кожного зі стовпців цієї гистограми дорівнює 1,5. Для побудови графіка функції щільності поверх гистограми скористаємось системою комп'ютерної математики Maxima (про роботу з системою Maxima див., наприклад, у [15, 19]):

```
(%i1) load(distrib)$  
(%i2) hist: histogram_description([0.626,1.21,7.53,  
1.74,0.695,5.38,0.750,0.584,3.17,1.50,0.969,  
1.92,1.16,0.150,0.491,2.09,3.44,0.132,0.234,  
0.969], nclasses=5, frequency=density)$  
pdfexp: explicit(pdf_exp(x,0.5757),x,0.01,9)$  
wxdraw2d(hist,line_width=3,pdfexp);
```

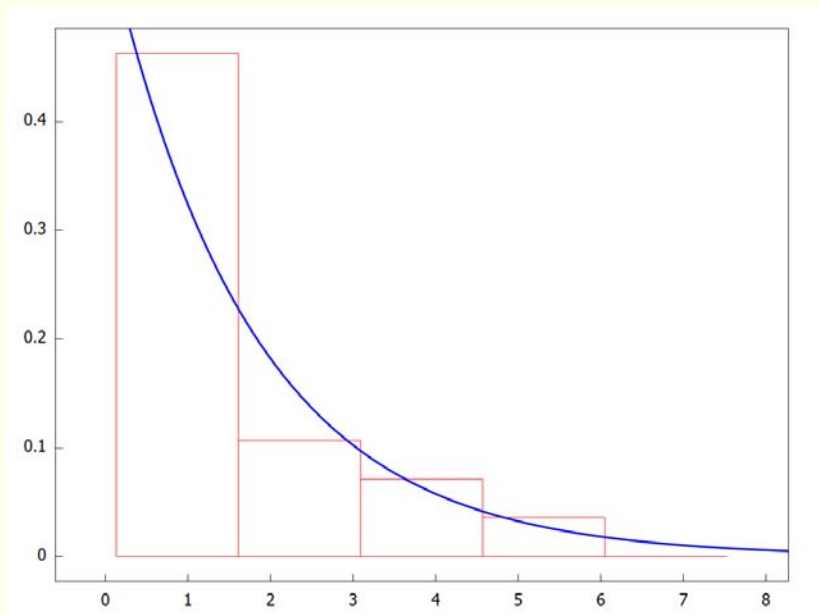


Рис. 9.17.

На рис. 9.17 поверх гістограми площею 1 побудовано графік функції щільності. Оскільки графік функції щільності приблизно відповідає гістограмі, то можна вважати, що експоненціальний розподіл достатньо добре наближає дані спостережень. ■

Аналогічний результат можна отримати, скориставшись пакетом STATISTICA. Цей пакет автоматично знаходить параметри розподілу і будує графік функції щільності поверх гістограми (рис. 9.18).

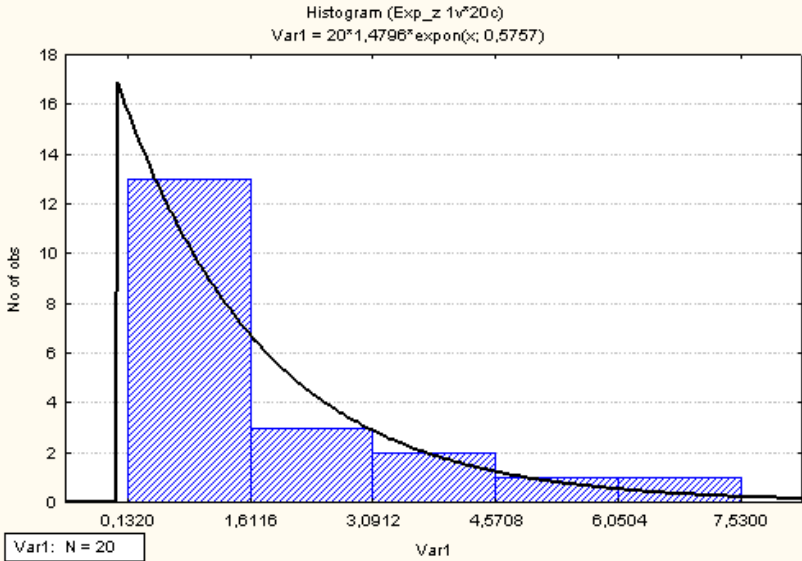


Рис. 9.18.

**Частотне порівняння.** Для неперервних випадкових величин частотне порівняння — це альтернативне графічне порівняння гістограми з функцією щільності. Нехай  $[b_0, b_1)$ ,  $[b_1, b_2)$ ,  $\dots$ ,  $[b_{k-1}, b_k)$  — набір  $k$  півінтервалів гістограми, ширина кожного з яких становить  $\Delta b = b_j - b_{j-1}$ , величина  $h_j$  — зареєстрована частка з  $n$  спостережень, які потрапляють в  $j$ -й півінтервал  $[b_{j-1}, b_j)$ , а  $r_j$  — очікувана частка з  $n$  спостережень, які мають потрапити в  $j$ -й півінтервал для вибраного розподілу, тобто

$$r_j = \int_{b_{j-1}}^{b_j} f(x) dx = F(b_j) - F(b_{j-1}), \quad j = 1, 2, \dots, k, \quad (9.19)$$

де  $f(x)$  — функція щільності, а  $F(x)$  — функція розподілу вибраного розподілу ймовірностей. Після обчислення  $h_j$  і  $r_j$

виконується їхнє порівняння. Для цього поруч різним кольором відображаються пари стовпчиків з висотами  $h_j$  і  $r_j$  для  $j = 1, 2, \dots, k$ . Якщо обидва стовпчики в кожній парі є приблизно рівними, то теоретичний розподіл відповідає даним спостережень. У протилежному випадку потрібно шукати інший теоретичний розподіл.

Для дискретних випадкових величин частотне порівняння полягає в порівнянні гістограми з імовірністю  $p(x)$  вибраного розподілу ймовірностей. Нехай  $h_j$  — частка величин, рівних значенню  $x_j$ , яка спостерігається, а  $r_j$  — очікувана частка спостережень, рівних  $x_j$ , тобто  $r_j = p(x_j)$ . Як і у випадку неперервних випадкових величин, будуються стовпчики для  $h_j, r_j$  і здійснюється їхнє візуальне порівняння.

**Приклад 2.** З допомогою частотного порівняння перевірити гіпотезу про придатність експоненціального розподілу для моделювання даних спостережень з попереднього прикладу: 0,626; 1,21; 7,53; 1,74; 0,695; 5,38; 0,750; 0,584; 3,17; 1,50; 0,969; 1,92; 1,16; 0,150; 0,491; 2,09; 3,44; 0,132; 0,234; 0,969.

**Розв'язання.** Параметр  $\lambda$  експоненціального розподілу для цих даних спостережень був обчислений у попередньому прикладі, він наближено дорівнює 0,5757. Функція розподілу для експоненціального закону розподілу подається формулою (8.8). Як і в прикладі з § 9.9, розіб'ємо проміжок  $[0, 1; 7, 6]$ , який містить дані спостережень, на п'ять однакових частин (довжиною 1,5), обчислимо величини  $h_j$  і  $r_j$  (останні за формулами (9.19)) для  $j = 1, 2, 3, 4, 5$ , а результати після заокруглень наведемо в наступній таблиці.

$j$	1	2	3	4	5
$[b_{j-1}; b_j)$	[0,1; 1,6)	[1,6; 3,1)	[3,1; 4,6)	[4,6; 6,1)	[6,1; 7,6]
$h_j$	0,65	0,15	0,1	0,05	0,05
$r_j$	0,546	0,2302	0,0971	0,0409	0,0173

Пари стовпців для частотного порівняння, побудовані з допомогою електронних таблиць Excel, наведені на рис. 9.19.

Оскільки стовпці у парах є майже однаковими, то можна вважати, що експоненціальний розподіл достатньо добре наближає дані спостережень. ■

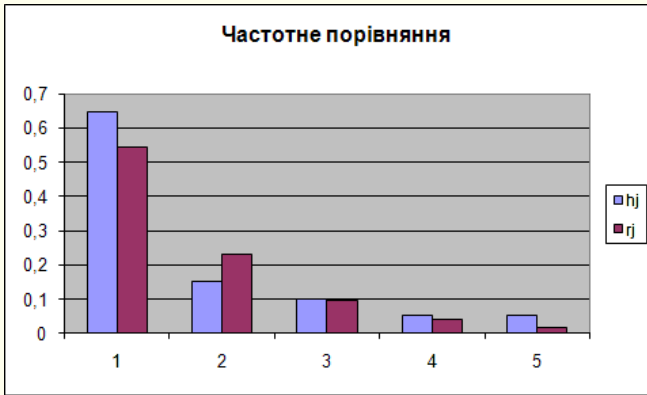


Рис. 9.19.

**Графік відмінностей між функціями розподілу.** За даними спостережень можна побудувати емпіричну функцію розподілу (формули (9.1) і (9.4) з §§ 9.2 і 9.3). Безпосередньо порівнювати графіки теоретичної і емпіричної функцій розподілу незручно, бо вони є подібними. Зручніше побудувати графік різниці емпіричної і теоретичної функцій розподілу. В ідеалі при відповідності розподілів він мав би бути прямою горизонтальною лінією  $y = 0$ . Але за рахунок неминучих відхилень графік буде кривою. Чим суттєвішими є відхилення, тим менше підходить вибраний розподіл для пояснення даних.

**Приклад 3.** З допомогою графіка відмінностей між функціями розподілу перевірити гіпотезу про придатність експоненціального розподілу для моделювання даних спостережень з попереднього прикладу: 0,626; 1,21; 7,53; 1,74; 0,695; 5,38; 0,750; 0,584; 3,17; 1,50; 0,969; 1,92; 1,16; 0,150; 0,491; 2,09; 3,44; 0,132; 0,234; 0,969.

**Розв'язання.** Параметр  $\lambda$  експоненціального розподілу для цих даних спостережень був обчислений у прикладі 1, він наближено дорівнює 0,5757. Побудуємо графік різниці емпіричної і теоретичної функцій розподілу з допомогою системи комп'ютерної математики Махіма (про роботу з системою Махіма див., наприклад, у [15, 19]):

```
(%i3) load(distrib)$
(%i4) ds: sort([0.626,1.21,7.53,1.74,0.695,5.38,0.750,
0.584,3.17,1.50,0.969,1.92,1.16,0.150,0.491,
2.09,3.44,0.132,0.234,0.969]))$
(%i5) FT(x):=cdf_exp(x,0.5757)$
(%i6) FE(x):=block([n:length(ds)],
if x<ds[1] then return(0),
if x>=ds[n] then return(1),
for i:1 thru n-1 do
if x>=ds[i] and x<ds[i+1] then
return((i-1)/(n-1)+(x-ds[i])/((n-1)*(ds[i+1]-
ds[i]))))$
(%i7) for j:0 step 0.01 thru 10 do
if j=0 then sp:[[j,FE(j)-FT(j)]]
else sp:append(sp,[[j,FE(j)-FT(j)]])$
(%i8) wxdraw2d(color=black, points_joined=true,
point_type=none, line_width=2, points(sp),
xlabel=x, ylabel="FE(x)-FT(x)");
```



Рис. 9.20.

На рис. 9.20 зображено шуканий графік різниці емпіричної і теоретичної функцій розподілу. Для його побудови дані спостережень було відсортовано за зростанням (див. § 9.2), задано функцію  $FT(x)$  для обчислення значення теоретичної функції розподілу, створено процедуру  $FE(x)$  для обчислення значення емпіричної функції розподілу і протабульовано їхню різницю на проміжку  $[0, 10]$  з кроком 0,01. З графіка видно, що відхилення між емпіричною і теоретичною функціями розподілу для всіх значень  $x$  є невеликим (меншим від 0,1), тому ми можемо вважати, що експоненціальний розподіл достатньо добре наближає дані спостережень. Найбільша невідповідність спостерігається поблизу лівого кінця області даних спостережень. ■

### § 9.12. Перевірка гіпотези придатності розподілу з допомогою критерію $\chi^2$

Найбільш вживаним і найдавнішим критерієм перевірки гіпотези про придатність розподілу є критерій узгодженості Пірсона, тобто критерій  $\chi^2$  («хі-квадрат»).

Для застосування цього критерію треба спочатку розбити всю область розподілу на  $k$  суміжних півінтервалів  $[a_0, a_1)$ ,  $[a_1, a_2)$ ,  $\dots$ ,  $[a_{k-1}, a_k)$ , де  $a_0$  може дорівнювати  $-\infty$  (тобто перший інтервалом буде  $(-\infty, a_1)$ ) або  $a_k$  може дорівнювати  $+\infty$ , або може бути одночасно одне та інше. Позначимо через  $N_j$  кількість величин  $x_i$  в  $j$ -му півінтервалі  $[a_{j-1}, a_j)$  для  $j = 1, 2, \dots, k$ , причому

$$\sum_{j=1}^k N_j = n.$$

Обчислимо очікувану частку  $p_j$  тих величин  $x_i$ , які потрапляють в  $j$ -й півінтервал. Для неперервних даних

$$p_j = \int_{a_{j-1}}^{a_j} f(x) dx = F(a_j) - F(a_{j-1}),$$

де  $f(x)$  — щільність вибраного розподілу, а  $F(x)$  — відповідна функція розподілу. Для дискретних даних

$$p_j = \sum_{a_{j-1} \leq x_i < a_j} p(x_i),$$

де  $p(x)$  — імовірність вибраного розподілу. Далі обчислимо

$$\chi^2 = \sum_{j=1}^k \frac{(N_j - np_j)^2}{np_j} \quad (9.20)$$

і порівнюємо його з  $\chi_{k-1, \alpha}^2$  або  $\chi_{k-m-1, \alpha}^2$  — теоретичним (табличним) значенням критерію, де  $\alpha$  — рівень значущості (0,01,

0,05 і т. п.),  $m$  — кількість параметрів розподілу, які визначались. Рівень значущості — це ймовірність відхилити правильну початкову гіпотезу про придатність знайденого розподілу.

Якщо  $\chi^2 > \chi_{k-1,\alpha}^2$ , то вибраний розподіл не підходить для моделювання даних спостережень при заданому рівні значущості  $\alpha$ . Якщо  $\chi^2 < \chi_{k-m-1,\alpha}^2$ , то підібраний розподіл можна використовувати для моделювання даних спостережень. Якщо  $\chi_{k-m-1,\alpha}^2 < \chi^2 < \chi_{k-1,\alpha}^2$ , то маємо сумнівний випадок, коли можна як прийняти, так і відкинути початкову гіпотезу про придатність знайденого розподілу.

Для невеликих  $n$  висновком критерію зазвичай буде придатність вибраного розподілу. Для великих  $n$  майже завжди висновком буде непридатність вибраного розподілу, бо накопичуються відхилення. Крім того, для середніх  $n$  висновок може залежати від розбиття області на проміжки та їхньої кількості. Рекомендується вибирати проміжки так, щоб імовірності потрапляння в них були однаковими  $p_1 = p_2 = \dots = p_k$  (для дискретного розподілу — приблизно однаковими). Крім того, кількість проміжків має бути не меншою від трьох, але по можливості такою, щоб виконувались нерівності  $np_j \geq 5$  для всіх  $j = 1, 2, \dots, k$ .

**Приклад.** З допомогою критерію  $\chi^2$  перевірити гіпотезу про придатність експоненціального розподілу для моделювання даних спостережень з попереднього прикладу: 0,626; 1,21; 7,53; 1,74; 0,695; 5,38; 0,750; 0,584; 3,17; 1,50; 0,969; 1,92; 1,16; 0,150; 0,491; 2,09; 3,44; 0,132; 0,234; 0,969. Рівень значущості  $\alpha = 0,01$ .

**Розв'язання.** Параметр  $\lambda$  експоненціального розподілу для цих даних спостережень був обчислений вище, він наближено дорівнює 0,5757. З умов  $np_j \geq 5$ ,  $j = 1, 2, \dots, k$ ,  $p_1 = p_2 = \dots = p_k$  для  $n = 20$  впливає, що число  $k$  має бути не більшим від 4. Тому візьмемо  $k = 4$ . З допомогою математичного пакета підберемо числа  $a_j$  так, щоб імовірності  $p_j$  були приблизно рівними, і заповнимо таблицю:

$j$	1	2	3	4
$[a_{j-1}; a_j)$	$[0; 0,5)$	$[0,5; 1,2)$	$[1,2; 2,4)$	$[2,4; \infty)$
$p_j$	0,2501	0,2487	0,25	0,2512
$N_j$	4	7	5	4

За формулою (9.20) обчислюємо  $\chi^2 \approx 1,2$ . Оскільки  $\chi_{2,0,01}^2 = 9,2$ , то  $\chi^2 < \chi_{2,0,01}^2$  і гіпотезу про придатність експоненціального розподілу для моделювання даних спостережень можна прийняти. ■

Є також інші критерії для перевірки гіпотез про придатність розподілу, але їх можна застосовувати лише для деяких конкретних розподілів (див., наприклад, [18]).

### § 9.13. Композиція функцій розподілу

Зустрічаються такі ситуації, коли гістограми мають кілька локальних вершин (мод), у таких випадках жоден зі стандартних розподілів не може забезпечити адекватне подання даних.

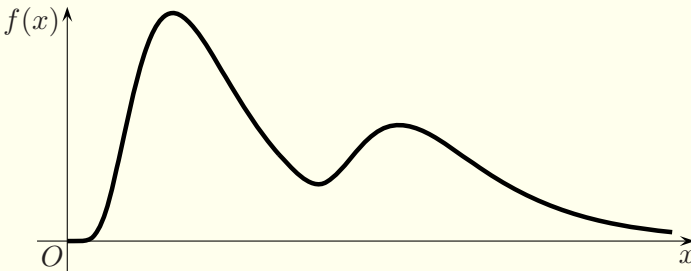


Рис. 9.21.

На рис. 9.21 наведено приклад графіка функції щільності неперервної випадкової величини, яка має два максимуми. Цей приклад може подавати дані про інтервали часу, протягом яких відбувається ремонт машин, зібрані впродовж року. Несправності машин бувають двох типів. Більшість із них носить не дуже серйозний характер і, отже, час ремонту для

них є невеликим. Цей випадок відповідає лівій вершині на рис. 9.21. Невелика кількість несправностей має більш серйозний характер, і час, потрібний для їхнього усунення, є більш тривалим, оскільки для їхнього ремонту потрібно замовляти деталі. На це вказує права вершина на рис. 9.21.

У випадку такої ситуації, як на рис. 9.21, можна побудувати емпіричний розподіл (§ 9.2) або скористатись методом композиції функцій розподілу, який полягає у наступному. Нехай  $f_1(x)$  — функція щільності для дрібного ремонту,  $f_2(x)$  — функція щільності для складного ремонту,  $p_1$  — імовірність дрібного ремонту, а  $p_2$  — імовірність складного ремонту. Тоді загальну функцію щільності  $f(x)$  можна подати формулою

$$f(x) = p_1 f_1(x) + p_2 f_2(x),$$

де  $p_1 > 0$ ,  $p_2 > 0$ ,  $p_1 + p_2 = 1$ . Відповідна функція розподілу записується аналогічно:

$$F(x) = p_1 F_1(x) + p_2 F_2(x),$$

де  $F_1(x)$  і  $F_2(x)$  — функції розподілу для дрібного і складного ремонтів відповідно.

Розглянемо тепер загальний випадок. Нехай функцію розподілу  $F(x)$  можна подати як комбінацію інших функцій розподілу  $F_1(x)$ ,  $F_2(x)$ ,  $\dots$ ,  $F_k(x)$ , тобто

$$F(x) = \sum_{j=1}^k p_j F_j(x),$$

де  $p_j > 0$ ,  $p_1 + p_2 + \dots + p_k = 1$ . Якщо випадкова величина має щільність розподілу  $f(x)$ , відповідну функції розподілу  $F(x)$ , то

$$f(x) = \sum_{j=1}^k p_j f_j(x),$$

де  $f_j(x)$  — частинні функції щільності. Аналогічно, для дискретних випадкових величин

$$P(x) = \sum_{j=1}^k p_j P_j(x),$$

де  $P_j(x)$  — частинні функції ймовірності.

Для генерування випадкової величини з таким законом розподілу використовують наступний алгоритм.

**Крок 1.** Знаходять рівномірно розподілене на півінтервалі  $[0, 1)$  випадкове число  $r$  і за формулою (8.2) та ймовірностями  $p_i$  визначають число  $j$ .

**Крок 2.** Генерують випадкове число  $x$  з функцією розподілу  $F_j(x)$ .

Метод композиції розподілів має більшою мірою теоретичне, ніж практичне значення. На практиці зазвичай легше побудувати емпіричний розподіл так, як це робиться у §§ 9.2–9.4, ніж шукати частинні функції розподілів.

### § 9.14. Зміщені розподіли

Для моделювання часу обслуговування часто використовуються експоненціальний розподіл, гамма-розподіл, логнормальний розподіл і розподіл Вейбулла. Всі ці розподіли генерують випадкові числа з проміжку  $[0, \infty)$ , і, отже, час обслуговування може виявитись як завгодно малим числом. Однак часто час обслуговування не може бути як завгодно малим. Наприклад, у банку, скоріш за все, неможливо обслужити клієнта швидше, ніж за 20 секунд.

Кожен розподіл можна змінити, змістивши його на деяку відстань вправо чи вліво. Параметр зміщення насправді включається у функцію щільності розподілу ймовірностей. Наприклад, зміщений експоненціальний розподіл має щільність

$$f(x) = \begin{cases} 0, & x < \gamma, \\ \lambda e^{-\lambda(x-\gamma)}, & x \geq \gamma, \end{cases}$$

де  $\gamma$  — величина зміщення вправо. Аналогічно, використання різниці  $x - \gamma$  замість  $x$  у функції щільності логнормального розподілу, гамма-розподілу і розподілу Вейбулла дозволяє отримати відповідний зміщений розподіл. Якщо звичайний розподіл діяв в області  $[0, \infty)$ , то після зміщення на  $\gamma$  його область дії становитиме  $[\gamma, \infty)$ .

Згенерувати випадкову величину, розподілену за зміщеним законом розподілу, якщо є випадкова величина, розподілена за звичайним законом, дуже просто. Для цього потрібно до випадкової величини зі звичайним законом розподілу додати зміщення  $\gamma$ .

У мові GPSS World для генерування зміщеного розподілу передбачений спеціальний параметр, замість якого у прикладах раніше ставився нуль (§ 6.14). Наприклад, випадкова величина зі зміщеним експоненціальним законом розподілу генерується функцією `Exponential(R,  $\gamma$ , 1/ $\lambda$ )`.

За наявності зміщених розподілів, крім інших параметрів, потрібно також визначити за даними спостережень зміщення  $\gamma$ , якщо воно не є відомим одразу з деяких міркувань. Для експоненціального закону розподілу знайти параметр  $\gamma$  можна разом з параметром  $\lambda$  методом максимальної правдоподібності, проте у випадках гамма-розподілу, розподілу Вейбулла і логнормального розподілу при застосуванні цього методу виникають проблеми. Вибрати замість зміщення  $\gamma$  найменше значення серед даних спостережень часто також не можна, бо зміщення зазвичай має бути меншим від нього.

Розглянемо спеціальний метод відшукування зміщення  $\gamma$ . Припустимо, що дані спостережень  $x_1, x_2, \dots, x_n$  впорядковано за зростанням:

$$x_{(1)} \leq x_{(2)} \leq \dots \leq x_{(n)}.$$

Пропонується (див. [18]) параметр  $\gamma$  обчислювати за форму-

лою

$$\gamma = \frac{x_{(1)}x_{(n)} - x_{(k)}^2}{x_{(1)} + x_{(n)} - 2x_{(k)}}, \quad (9.21)$$

де  $k$  — найменше ціле число з чисел  $\{2, 3, \dots, n-1\}$ , для якого  $x_{(k)} > x_{(1)}$ . Проте справжнє значення  $\gamma$  може бути значно меншим, ніж обчислене за формулою (9.21).

Після визначення зміщення  $\gamma$  знаходять числа  $\tilde{x}_i$  за формулами  $\tilde{x}_i = x_i - \gamma$ ,  $i = 1, 2, \dots, n$ . Решту параметрів розподілу визначають методом максимальної правдоподібності для отриманих чисел  $\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_n$ .

Нехай, наприклад, гістограма для даних спостережень має вигляд, поданий на рис. 9.22. Ці дані спостережень добре наближає зміщений логнормальний закон розподілу, графік функції щільності якого зображений на рис. 9.22 поверх гістограми.

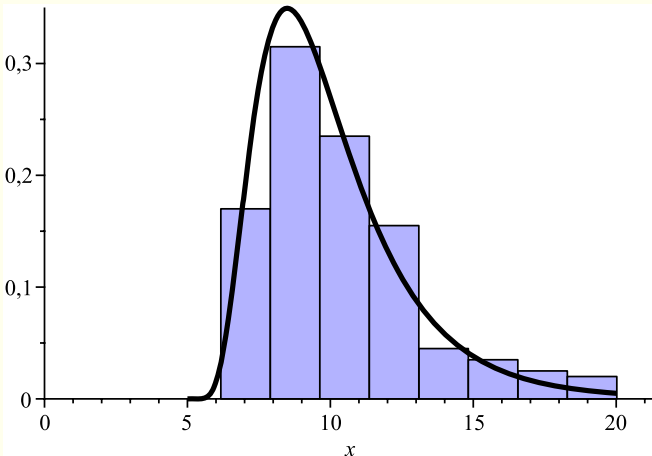


Рис. 9.22.

### § 9.15. Зрізані розподіли

У деяких випадках підібраний розподіл загалом є непоганою моделлю даних спостережень, але відомо, наприклад, що жодне значення не може бути більшим за деяку скінченну сталу  $b > 0$ . Якщо функція щільності підбраного розподілу має ненульові значення на півосі  $[0, \infty)$ , то вона несумісна з верхньою межею  $b$ , тому потрібно використовувати зрізаний розподіл. Його щільність визначається формулою

$$f^*(x) = \begin{cases} \frac{f(x)}{F(b)}, & 0 \leq x \leq b, \\ 0, & x < 0 \text{ або } x > b, \end{cases}$$

де

$$F(b) = \int_0^b f(x) dx < 1,$$

а  $f(x)$  — функція щільності незрізаного закону розподілу. Аналогічно можна зрізати розподіл і зліва.

На рис. 9.23 наведено як приклад графік функції щільності зрізаного справа у точці  $x = 7$  розподілу Вейбулла з параметрами  $\alpha = 2$ ,  $\beta = 4$ .

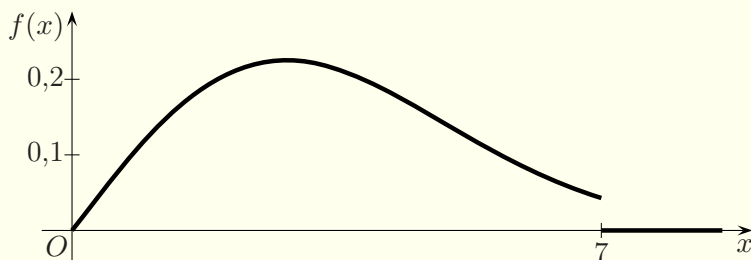


Рис. 9.23.

Розглянемо тепер загальний випадок. Припустимо, що  $f(x)$  — щільність деякого розподілу, а  $F(x)$  — відповідна функція розподілу. Для  $a < b$  (можливо, що  $a = -\infty$  або

$b = +\infty$ ) визначимо щільність зрізаного розподілу за формулою

$$f^*(x) = \begin{cases} \frac{f(x)}{F(b)-F(a)}, & a \leq x \leq b, \\ 0, & x < a \text{ або } x > b, \end{cases}$$

і відповідну функцію зрізаного розподілу як

$$F^*(x) = \begin{cases} 0, & x < a, \\ \frac{F(x)-F(a)}{F(b)-F(a)}, & a \leq x \leq b, \\ 1, & x > b. \end{cases}$$

Розглянемо тепер моделювання випадкових величин зі зрізаним законом розподілу. Якщо для функції розподілу  $F(x)$  можна знайти обернену функцію  $F^{-1}(r)$ , то, користуючись методом оберненої функції, матимемо

$$\frac{F(x) - F(a)}{F(b) - F(a)} = r,$$

де  $r$  — випадкове число, рівномірно розподілене на проміжку  $[0, 1)$ . Тоді

$$\begin{aligned} F(x) &= F(a) + (F(b) - F(a))r \quad \Rightarrow \\ x &= F^{-1}(F(a) + (F(b) - F(a))r). \end{aligned} \quad (9.22)$$

Отже, у випадку, коли для функції розподілу  $F(x)$  вдається знайти обернену функцію  $F^{-1}(r)$ , для моделювання випадкової величини зі зрізаним законом розподілу генерують випадкове число  $r \in [0, 1)$  і обчислюють значення  $x$  за формулою (9.22). Цей метод можна використовувати, наприклад, для моделювання зрізаного експоненціального розподілу або зрізаного розподілу Вейбулла.

У випадку, коли обернену функцію  $F^{-1}(r)$  до функції розподілу  $F(x)$  неможливо знайти, розглянутим методом для моделювання випадкової величини скористатись не вдається. У

цьому випадку для моделювання випадкової величини зі зрізаним законом розподілу користуються наступним алгоритмом.

**Крок 1.** Генерують випадкове число  $x$  з функцією розподілу  $F(x)$ .

**Крок 2.** Якщо число  $x$  не належить проміжку  $[a, b]$ , то переходять до кроку 1, інакше вважають, що випадкове число  $x$  має зрізаний закон розподілу з функцією розподілу  $F^*(x)$ .

Мова імітаційного моделювання GPSS World не має вбудованих засобів генерації зрізаних законів розподілу. Тому такі розподіли при використанні цієї мови доводиться моделювати так, як описано вище.

### § 9.16. Вибір розподілу за відсутності даних спостережень

При вивченні багатьох систем немає можливості зібрати дані спостережень. Це може бути пов'язано як з відсутністю таких систем, так і з неможливістю збору даних. У таких випадках застосовувати розглянуті вище методи не можна, проте існують спеціальні методи, які дозволяють підібрати розподіл і за відсутності даних спостережень. Зрозуміло, що такий розподіл буде лише наближено моделювати реальні процеси. У цьому параграфі ми розглянемо кілька методів вибору розподілу за відсутності даних спостережень.

Перш за все, у будь-якому випадку потрібно оцінити проміжок  $[a, b]$ , в який з імовірністю, близькою до одиниці, буде потрапляти випадкова величина. Якщо випадкова величина  $\epsilon$ , наприклад, часом виконання деякої задачі, то фахівцям пропонується визначити крайні оцінки часу виконання задачі.

Якщо, крім проміжку  $[a, b]$ , більше нічого не відомо, то для моделювання випадкової величини використовують рівномірний розподіл на відрізку  $[a, b]$ . Часто на початкових етапах моделювання (при створенні пробної моделі) розподіл вважають рівномірним, а пізніше його уточнюють.

Якщо, крім проміжку  $[a, b]$ , вдається знайти моду  $c$  — найбільш імовірне значення випадкової величини, то для моделювання цієї випадкової величини використовують трикутний закон розподілу з параметрами  $a$ ,  $b$  і  $c$  (графік функції щільності трикутного розподілу зображений на рис. 8.5, с. 276). Оцінку моди здійснюють експерти.

Найбільш точний метод моделювання випадкової величини полягає в наближенні її бета-розподілом на проміжку  $[a, b]$ . Проте у цьому випадку потрібно мати найбільше інформації: проміжок зміни випадкової величини  $[a, b]$ , моду  $c$  і середнє значення випадкової величини  $\mu$ . Всі ці параметри оцінюють експерти. У § 9.7 вже йшла мова про те, що якщо  $\xi$  є випадковою величиною з класичним бета-розподілом на проміжку  $[0, 1]$ , то  $\eta = a + (b - a)\xi$  є випадковою величиною з бета-розподілом на проміжку  $[a, b]$ . Бета-розподіл має два параметри форми  $\alpha_1$  і  $\alpha_2$ . З теорії ймовірностей відомо [18], що середнє значення  $\mu$  випадкової величини з бета-розподілом для проміжку  $[a, b]$  завжди подається формулою

$$\mu = a + \frac{\alpha_1(b - a)}{\alpha_1 + \alpha_2}. \quad (9.23)$$

Якщо  $\alpha_1 > 1$  і  $\alpha_2 > 1$ , то графік функції щільності бета-розподілу для проміжку  $[a, b]$  має приблизно такий вигляд, як показано на рис. 9.14, 9.15, тобто має один максимум всередині проміжку  $[a, b]$ . У цьому випадку моду  $c$  бета-розподілу обчислюють за формулою

$$c = a + \frac{(\alpha_1 - 1)(b - a)}{\alpha_1 + \alpha_2 - 2}. \quad (9.24)$$

У всіх інших випадках мода спостерігатиметься на кінцях проміжку  $[a, b]$  або буде відсутньою взагалі.

Визначимо параметри  $\alpha_1$  і  $\alpha_2$  через  $\mu$ ,  $c$ ,  $a$  і  $b$ . З формул (9.23), (9.24) маємо:

$$\begin{cases} (\mu - a)\alpha_3 = \alpha_1(b - a), \\ (c - a)(\alpha_3 - 2) = (\alpha_1 - 1)(b - a), \end{cases}$$

де  $\alpha_3 = \alpha_1 + \alpha_2$ . Віднімаючи від другого рівняння перше, отримуємо:

$$(c - \mu)\alpha_3 - 2c + 2a = a - b,$$

звідки в припущенні, що  $\mu \neq c$ , маємо:

$$\alpha_3 = \frac{2c - a - b}{c - \mu}.$$

Отже,

$$\alpha_1 = \frac{\mu - a}{b - a} \cdot \frac{2c - a - b}{c - \mu}, \quad \alpha_2 = \frac{b - \mu}{b - a} \cdot \frac{2c - a - b}{c - \mu}. \quad (9.25)$$

Формули (9.25) можна використовувати тільки тоді, коли  $\mu \neq c$ . Якщо  $\mu = c = \frac{a+b}{2}$ , то варто використовувати трикутний закон розподілу. Якщо  $\mu = c \neq \frac{a+b}{2}$ , то, скоріш за все, допущено помилку при оцінюванні величин  $\mu$  і  $c$ . Якщо при обчисленні за формулами (9.25) хоча б одне з чисел  $\alpha_1$  чи  $\alpha_2$  виявляється меншим або рівним одиниці, то числа  $\mu$  і  $c$  неправильно оцінені або бета-розподіл неможливо використовувати для моделювання даної випадкової величини (у всякому разі, в такому випадку не можна користуватись знайденими за формулами (9.25) параметрами  $\alpha_1$  і  $\alpha_2$ ). Якщо  $\mu > c$ , то графік функції щільності розподілу ймовірностей скошений вправо, якщо  $\mu < c$ , то — вліво.

**Приклад.** Про випадкову величину відомо лише те, що вона може набувати значень з проміжку  $[2, 5]$  з середнім значенням 4 і модою 4,4. Знайти параметри розподілу для моделювання цієї випадкової величини.

**Розв'язання.** Знайдемо параметри бета-розподілу для проміжку  $[2, 5]$  за формулами (9.25):  $\alpha_1 = 3$ ,  $\alpha_2 = 1,5$ . Отримані числа є більшими від одиниці, тому вони є справжніми параметрами бета-розподілу для проміжку  $[2, 5]$ . ■

### § 9.17. Групове надходження вимог

У багатьох реальних системах вимоги надходять групами. Наприклад, люди, які з'являються на спортивні змагання, в кафе чи в готель, часто приходять групами. Цей параграф присвячено моделюванню такого процесу.

Спочатку моделюється час між надходженням груп. Для цього може використовуватись будь-який неперервний розподіл. Потім з допомогою дискретного розподілу здійснюється моделювання розмірів послідовних груп. Розміри груп мають бути натуральними числами.

Розглянемо моделювання такої ситуації з допомогою мови імітаційного моделювання GPSS World. Для створення груп використовується блок **GENERATE** з деяким неперервним розподілом. Якщо у групі є  $k$  об'єктів, то додаткові  $k - 1$  транзактів створюються блоком **SPLIT**  $k - 1$ , тобто створюється  $k - 1$  копій транзакта. Транзакти і їхні копії є рівноправними, і, якщо групу не треба об'єднувати після якої-небудь черги, то всі транзакти можуть окремо залишати модель через блок **TERMINATE**. Для того щоб групи зберігались під час руху через модель, потрібно використовувати блоки **MATCH** або **GATHER** для синхронізації руху транзактів (§ 7.1).

Якщо групи вимог скрізь у системі мають оброблятися разом, то їх можна моделювати окремими транзактами і взагалі обійтись без блока **SPLIT**. Транзакт, що моделюватиме групу вимог, в одному зі своїх параметрів має містити кількість вимог у групі, а при входженні в багатоканальний пристрій він має займати кількість каналів, відповідну обсягу групи.

### § 9.18. Використання умовних розподілів для моделювання випадкових векторів

Розглянемо наступну ситуацію. У ремонтній майстерні здійснюється ремонт пристроїв у два етапи. На першому етапі діагностуються наявні несправності і замовляються потрібні

для ремонту деталі зі складу. На другому етапі здійснюється усунення несправностей. Можна припускати, що більш серйозні несправності вимагають більше часу як на їхнє виявлення, так і на їхнє усунення. Отже, дві випадкові величини, які характеризують час ремонту, мають бути пов'язаними між собою.

Пов'язані між собою випадкові величини можна моделювати з допомогою спільних або багатовимірних розподілів.

Розглянемо моделювання випадкового вектора з допомогою умовних розподілів. Припустимо, що в нас є функція спільного розподілу  $F(x_1, x_2, \dots, x_d)$ , для якої ми хочемо згенерувати випадковий вектор  $\vec{\xi} = (\xi_1, \xi_2, \dots, \xi_d)^T$ . Припустимо також, що для  $i = 2, 3, \dots, d$  можна отримати умовний розподіл величин  $\xi_i$  за умови, що  $\xi_j = x_j$  для  $j = 1, 2, \dots, i - 1$ . Позначимо функцію умовного розподілу через  $F_i(x_i | \xi_1 = x_1, \xi_2 = x_2, \dots, \xi_{i-1} = x_{i-1})$ . Крім того, нехай  $F_1(x_1)$  — функція безумовного розподілу. Тоді загальний алгоритм генерування випадкового вектора  $\vec{\xi}$  з функцією розподілу  $F(x_1, x_2, \dots, x_d)$  має наступний вигляд.

**Крок 1.** Генеруємо випадкову величину  $\xi_1$  з функцією розподілу  $F_1(x_1)$ .

**Крок 2.** Генеруємо випадкову величину  $\xi_2$  з функцією розподілу  $F_2(x_2 | \xi_1 = x_1)$ .

**Крок 3.** Генеруємо випадкову величину  $\xi_3$  з функцією розподілу  $F_3(x_3 | \xi_1 = x_1, \xi_2 = x_2)$ .

...

**Крок  $d$ .** Генеруємо випадкову величину  $\xi_d$  з функцією розподілу  $F_d(x_d | \xi_1 = x_1, \xi_2 = x_2, \dots, \xi_{d-1} = x_{d-1})$ .

**Крок  $d+1$ .** Виводимо  $\vec{\xi} = (\xi_1, \xi_2, \dots, \xi_d)^T$ .

Моделювання спільного розподілу з допомогою умовних розподілів має більшою мірою теоретичне, ніж практичне значення. На практиці для застосування цього підходу не лише має бути відомим спільний розподіл, але й виведені всі умовні і безумовні розподіли, для яких мають бути розроблені алгоритми генерації випадкових величин. Такого рівня деталізації

рідко коли вдається досягти при моделюванні систем.

### § 9.19. Двовимірний нормальний розподіл і його моделювання

Двовимірний нормальний розподіл є, певно, найвідомішим окремим випадком багатовимірного розподілу. Безпосереднє застосування цього розподілу є обмеженим його симетричністю і безмежністю за обома змінними. Проте двовимірний нормальний розподіл служить основою для більш придатного для моделювання пов'язаних випадкових величин двовимірного логнормального розподілу.

Функція щільності двовимірного нормального розподілу має вигляд

$$f(x, y) = \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-r_{xy}^2}} \exp\left(-\frac{1}{2(1-r_{xy}^2)}\left(\frac{(x-\mu_x)^2}{\sigma_x^2} - 2r_{xy}\frac{(x-\mu_x)(y-\mu_y)}{\sigma_x\sigma_y} + \frac{(y-\mu_y)^2}{\sigma_y^2}\right)\right),$$

де  $\mu_x, \mu_y$  — середні значення по  $x$  і  $y$ ,  $\sigma_x, \sigma_y$  — середньоквадратичні відхилення, а  $r_{xy}$  — кореляція.

Нехай є дані спостережень  $x_1, x_2, \dots, x_n$  і  $y_1, y_2, \dots, y_n$ , пов'язані між собою двовимірним нормальним розподілом. Методом максимальної правдоподібності можна отримати наступні оцінки параметрів цього розподілу:

$$\mu_x = \frac{1}{n} \sum_{i=1}^n x_i, \quad \mu_y = \frac{1}{n} \sum_{i=1}^n y_i, \quad (9.26)$$

$$\sigma_x = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \mu_x)^2}, \quad \sigma_y = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \mu_y)^2}, \quad (9.27)$$

$$r_{xy} = \frac{K_{xy}}{\sigma_x\sigma_y}, \quad K_{xy} = \frac{1}{n} \sum_{i=1}^n (x_i - \mu_x)(y_i - \mu_y). \quad (9.28)$$

Для генерування випадкових чисел з двовимірним нормальним законом розподілу потрібно використовувати наступний алгоритм.

**Крок 1.** Генеруємо два числа  $z_1$  і  $z_2$  з нормальним одновимірним розподілом з середнім значенням 0 і середньоквадратичним відхиленням 1.

**Крок 2.** Обчислюємо

$$x = \mu_x + \sigma_x z_1, \quad y = \mu_y + \sigma_y r_{xy} z_1 + \sigma_y \sqrt{1 - r_{xy}^2} z_2.$$

Отримані числа  $x$  і  $y$  мають двовимірний нормальний закон розподілу з середніми значеннями  $\mu_x$ ,  $\mu_y$ , середньоквадратичними відхиленнями  $\sigma_x$ ,  $\sigma_y$  і кореляцією  $r_{xy}$ .

У мові імітаційного моделювання GPSS World немає готової функції чи процедури для генерування пари випадкових величин, пов'язаних двовимірним нормальним законом розподілу. Тому таку пару випадкових величин генерують, користуючись наведеним вище алгоритмом. Для їхнього генерування можна створити PLUS-процедуру (§ 7.9).

**Приклад.** На верстаті виготовляються валики. Довжина  $x$  і діаметр  $y$  валиків є випадковими величинами з двовимірним нормальним законом розподілу з середнім значенням довжини 60 мм, середньоквадратичним відхиленням довжини 0,04 мм, середнім значенням діаметра 20 мм, середньоквадратичним відхиленням діаметра 0,025 мм і кореляцією 0,3. Валик вважається стандартним, якщо його розміри задовольняють умови:

$$59,9 \text{ мм} < x < 60,15 \text{ мм},$$

$$19,95 \text{ мм} < y < 20,05 \text{ мм}.$$

Нестандартний валик є бракованим.

Склавши модель на мові GPSS World, змоделювати виготовлення 20000 валиків, визначити відсоток бракованих виробів, створити таблиці розподілів довжин і діаметрів валиків та побудувати гістограми, відповідні цим таблицям. Для

генерації пов'язаних між собою випадкових величин з двовимірним нормальним розподілом створити універсальну PLUS-процедуру.

**Розв'язання.** Будуємо наступну модель:

```
x_ EQU 0
y_ EQU 0
PROCEDURE normal2(mux,muy,sigmax,sigmay,rxy)
BEGIN
TEMPORARY z1_,z2_;
z1_=Normal(5,0,1);
z2_=Normal(5,0,1);
x_=mux+z1_#sigmax;
y_=muy+z1_#rxy#sigmay+z2_#sqr(1-rxy#rxy)#sigmay;
RETURN x_;
END;

tabx TABLE P$dov,59.7,0.05,15
taby TABLE P$dia,19.7,0.05,15

GENERATE ,,20000
ASSIGN dov,(normal2(60,20,0.04,0.025,0.3))
ASSIGN dia,y_
TEST E ((P$dov<60.15)&(P$dov>59.9)&(P$dia<20.05)&
(P$dia>19.95)),0,tabul ;продовження попереднього рядка
SAVEVALUE brak+,1
tabul TABULATE tabx
TABULATE taby
TERMINATE 1
START 20000
```

Оскільки всередині PLUS-процедури не можна присвоювати значення параметрам транзактів, а процедура вертає лише одне число, для передачі з неї результатів використано змінні користувача  $x_$ ,  $y_$ . Насправді достатньо було взяти тільки одну змінну користувача, друга змінна використана лише для кращого сприйняття програми. Для перевірки відповідності

валика стандартам використано логічний вираз (§ 6.5). Цей вираз не помістився в рядку на сторінці і тому був перенесений на наступний рядок, однак у тексті програми всі блоки мають займати по одному рядку. У комірці пам'яті **brak** (§ 6.9) накопичується кількість бракованих виробів.

Зі звіту бачимо:

SAVEVALUE	RETRY	VALUE
BRAK	0	1021.000

Тому відсоток бракованих деталей знаходимо за формулою

$$\frac{\text{brak}}{20000} \cdot 100\% = \frac{1021}{20000} \cdot 100\% \approx 5\%.$$

У звіті містяться також таблиці розподілів довжин і діаметрів валиків:

TABLE	MEAN	STD.DEV.	RANGE	RETRY	FREQUENCY	CUM.%
TABX	60.000	0.040		0		
		59.800 -	59.850		1	0.01
		59.850 -	59.900		131	0.66
		59.900 -	59.950		1971	10.52
		59.950 -	60.000		7870	49.87
		60.000 -	60.050		7872	89.22
		60.050 -	60.100		2028	99.36
		60.100 -	60.150		125	99.99
		60.150 -	60.200		2	100.00
TABY	20.000	0.025		0		
		19.850 -	19.900		2	0.01
		19.900 -	19.950		457	2.29
		19.950 -	20.000		9492	49.76
		20.000 -	20.050		9601	97.76
		20.050 -	20.100		447	100.00
		20.100 -	20.150		1	100.00

Гістограма для довжин валиків наведена на рис. 9.24. Гістограма для діаметрів валиків має аналогічний вигляд, її пропонується побудувати читачам самостійно. З таблиць і гістограм видно, що двовимірний нормальний закон розподілу згенеровано правильно. ■

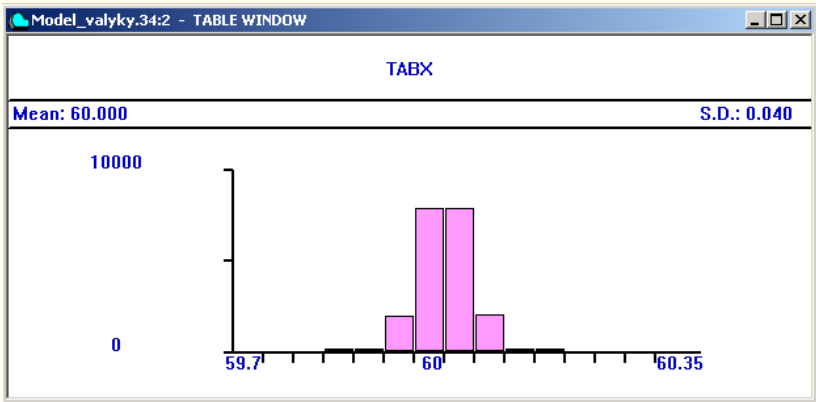


Рис. 9.24.

Для моделювання цієї задачі необов'язково було використовувати мову GPSS World, бо тут не потрібно враховувати час. У даному випадку досить легко можна було скласти програму на будь-якій універсальній мові програмування.

### § 9.20. Двовимірний логнормальний розподіл і його моделювання

Двовимірний логнормальний розподіл дає можливість використовувати безумовні розподіли з додатною асиметрією в області  $[0, \infty)$  і можливістю кореляції між ними.

Опишемо двовимірний логнормальний розподіл не через функцію щільності розподілу, а з точки зору його зв'язку з двовимірним нормальним розподілом. Випадкові величини  $\xi$ ,  $\eta$  тоді і тільки тоді мають двовимірний логнормальний розподіл, коли випадкові величини  $\varphi = \ln \xi$ ,  $\psi = \ln \eta$  мають двовимірний нормальний розподіл. Тоді випадкові величини двовимірного логнормального розподілу можуть бути подані формулами

$$\xi = e^{\varphi}, \quad \eta = e^{\psi},$$

де випадкові величини  $\varphi$  і  $\psi$  мають двовимірний нормальний розподіл. Параметрами двовимірного логнормального розподілу вважають параметри відповідного двовимірного нормального розподілу.

Для визначення параметрів двовимірного логнормального закону розподілу для даних спостережень  $x_1, x_2, \dots, x_n$  і  $y_1, y_2, \dots, y_n$  спочатку за формулами  $u_i = \ln x_i$ ,  $v_i = \ln y_i$ ,  $i = 1, 2, \dots, n$ , утворюють набори даних  $u_1, u_2, \dots, u_n$  і  $v_1, v_2, \dots, v_n$ , а потім для них за формулами, отриманими з формул (9.26)–(9.28) заміною  $x_i$  на  $u_i$  і  $y_i$  на  $v_i$ ,  $i = 1, 2, \dots, n$ , визначають параметри відповідного двовимірного нормального розподілу.

Для генерування чисел з двовимірним логнормальним розподілом генерують числа  $u$  і  $v$  з відповідним двовимірним нормальним розподілом і знаходять їхні експоненти  $e^u$  і  $e^v$ .

### § 9.21. Нестационарний пуассонівський потік і його моделювання

Якщо стаціонарний пуассонівський потік є стаціонарним, ординарним і не має післядії, то нестационарний пуассонівський потік має лише дві останні властивості. Його інтенсивність  $\lambda(t)$  залежить від часу. У багатьох реальних системах частіше від стаціонарного зустрічається нестационарний пуассонівський потік. Наприклад, інтенсивність прибуття клієнтів або їхніх груп у ресторан є більшою в час обіду, ніж посеред другої половини дня; рух автомобілів по дорозі часто є найбільш інтенсивним вранці і ввечері в час пік і найменш інтенсивним вночі.

Виникає питання, як визначити інтенсивність  $\lambda(t)$  нестационарного пуассонівського потоку. Розглянемо відповідь на це питання на прикладі роботи магазину.

У магазині фіксується час прибуття покупців протягом робочого дня з 8.00 до 19.00 протягом 10 днів. Щоб визначити оцінку  $\lambda(t)$ , кожен день розбиваємо на проміжки часу три-

валістю 10 хв. Для кожного дня визначаємо кількість появ покупців у кожному з проміжків. Потім визначаємо середню кількість появ покупців у кожному проміжку за 10 днів. Нарешті, середню кількість появ покупців у кожному проміжку ділимо на тривалість цього проміжку, тобто на 10 хв. Отримується кусково-стала функція інтенсивності  $\lambda(t)$  (приклад її графіка наведено на рис. 9.25).

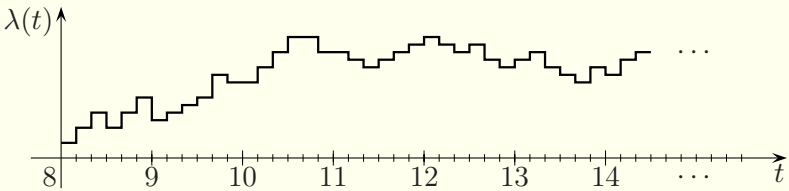


Рис. 9.25.

Довжину проміжків доводиться вибирати експериментально. Якщо, наприклад, проміжки довжиною 5 хв. дають надто «поламаний» графік функції  $\lambda(t)$ , то треба використовувати довші проміжки. Проміжки довжиною 15 хв. можуть дати надто «рівний» графік функції  $\lambda(t)$ , що свідчить про втрату інформації, в цьому випадку треба використовувати коротші проміжки.

При необхідності отриману кусково-сталу функцію  $\lambda(t)$  можна інтерполювати деякою гладкою функцією.

Для моделювання нестационарного пуассонівського потоку зручно використовувати *метод проріджування*. Припустимо, що  $\lambda^* = \max_t \lambda(t) < \infty$ . Метод проріджування полягає у генеруванні стаціонарного пуассонівського потоку зі сталою інтенсивністю  $\lambda^*$  і часом настання події  $t_i^*$  з наступним «проріджуванням»  $t_i^*$  відкиданням кожного значення  $t_i^*$  з імовірністю  $1 - \frac{\lambda(t_i^*)}{\lambda^*}$ . Так, скоріш за все, буде прийнято значення  $t_i^*$ , якщо  $\lambda(t_i^*)$  має велике значення, що дозволить забезпечити збільшення зростання частоти надходжень для великих значень

$\lambda(t)$ . Стационарний пуассонівський потік моделюється експоненціальним розподілом з параметром  $\lambda^*$ .

У припущенні, що значення  $t_{i-1}$  часу попередньої події у нестационарному пуассонівському потоці згенеровано правильно, для генерації значення  $t_i$  часу наступної події використовують наступний алгоритм, який ґрунтується на методі проріджування.

**Крок 1.** Нехай  $t = t_{i-1}$ .

**Крок 2.** Генеруємо  $r_1$  і  $r_2$  — випадкові числа, рівномірно розподілені на проміжку  $[0, 1)$ .

**Крок 3.** Замінюємо  $t$  на  $t - \frac{1}{\lambda^*} \ln(1 - r_1)$  (експоненціальний розподіл з інтенсивністю  $\lambda^*$ ).

**Крок 4.** Якщо  $r_2 \leq \frac{\lambda(t)}{\lambda^*}$ , то отримуємо  $t_i = t$ . У протилежному випадку повертаємось до кроку 2.

Якщо обчислення значення функції  $\lambda(t)$  виконується повільно, а  $\lambda_* = \min_t \lambda(t)$  не є дуже малою величиною, то можна оптимізувати алгоритм додатковим кроком 3а, який має виконуватись перед кроком 4: якщо  $r_2 \leq \frac{\lambda_*}{\lambda^*}$ , то отримуємо  $t_i = t$  і алгоритм завершується. Це буде ефективно працювати лише у випадку, коли функція  $\lambda(t)$  є достатньо рівною.

Мова імітаційного моделювання GPSS World не має вбудованих засобів генерації нестационарного пуассонівського потоку. Такий потік доводиться моделювати з допомогою наведеного вище алгоритму.

**Приклад.** Нехай нестационарний пуассонівський потік має інтенсивність  $\lambda(t)$ , яка змінюється за наведеним нижче законом.

$$\lambda(t) = \begin{cases} 5, & 0 \leq t \leq 10, \\ 1, & 10 < t \leq 20, \\ 6, & 20 < t \leq 30, \\ 5, & 30 < t \leq 40, \\ 2, & 40 < t \leq 50, \end{cases} \quad \lambda(t + 50) = \lambda(t).$$

Написати PLUS-процедуру на мові GPSS World для моделювання цього потоку. Скласти модель для перевірки правильності роботи PLUS-процедури.

**Розв'язання.** Модель для цього прикладу може бути такою:

```

ttt EQU 0
lamb FUNCTION ttt,D5
10,5/20,1/30,6/40,5/50,2
PROCEDURE nestac_poisson()
BEGIN
TEMPORARY t,rr2,lambda,t50;
t=AC1;
rr2=1;
lambda=0;
WHILE (rr2>lambda/6) DO
BEGIN
t=t+Exponential(5,0,1/6);
t50=int(t/50);
ttt=t-t50#50;
lambda=FN$lamb;
rr2=Uniform(7,0,1);
END;
RETURN (t-AC1);
END;

tab TABLE AC1,10,10,3000
GENERATE (nestac_poisson())
TABULATE tab
TERMINATE 1
START 1000

```

У цій моделі процедура `nestac_poisson` використовується для моделювання методом проріджування нестационарного пуассонівського потоку з заданою функцією інтенсивності  $\lambda(t)$ . Для спрощення моделі використана допоміжна дис-

кретна функція `lamb`. Змінна користувача `ttt` дозволяє передавати дані у функцію `lamb`. Побудована PLUS-процедура використовується для створення нестационарного пуассонівського вхідного потоку транзактів. Після виконання моделювання статистичний звіт міститиме таблицю часу появи транзактів у моделі. Читачам пропонується самостійно порівняти її з функцією  $\lambda(t)$ . ■

### Питання до розділу 9

1. Як можна використовувати дані спостережень для моделювання випадкових процесів?

2. Якою є функція розподілу емпірично розподіленої неперервної (дискретної) випадкової величини у випадку, коли відомими є дані окремих спостережень? Як її моделювати без використання мови GPSS (з використанням GPSS)?

3. Якою є функція розподілу емпірично розподіленої неперервної (дискретної) випадкової величини у випадку згрупованих даних? Як її моделювати без використання мови GPSS (з використанням GPSS)?

4. Назвіть і опишіть методи оцінки незалежності вибірки.

5. Назвіть етапи вибору теоретичного розподілу.

6. Які методи використовують для вибору сім'ї розподілів?

7. Як побудувати гістограму?

8. Який вигляд мають графіки функцій щільності рівномірного, нормального, експоненціального, логнормального розподілу, розподілу Вейбулла, гамма-розподілу, бета-розподілу?

9. Які розподіли випадкових величин найчастіше використовують для моделювання вхідного потоку (часу обслуговування, відхилення від середнього значення)?

10. Як можна використовувати підсумкову статистику при виборі сім'ї розподілів?

11. У чому полягає метод максимальної правдоподібності?

12. Назвіть основні графічні й аналітичні способи перевірки придатності теоретичного розподілу для моделювання даних спостереження.

13. Коли використовують і як моделюють зміщені і зрізані розподіли?

14. Як моделюють випадкову величину за відсутності даних спостережень?
15. Як моделюють групове надходження вимог?
16. Як визначити параметри двовимірного нормального (лог-нормального) закону розподілу? Як його моделювати?
17. Як моделюють нестационарний пуассонівський потік?

### Вправи до розділу 9

1. Обчисліть три числових значення неперервної випадкової величини з емпіричним розподілом, що відповідає вибірці: 12,34; 13,01; 8,75; 14,2; 10,21; 16,1; 9,2; 11,22; 10,82; 9,2. Вважати, що генератор випадкових чисел генерує значення 0,239, 0,792 і 0,465.

2. На мові імітаційного моделювання GPSS створіть неперервну функцію для моделювання емпіричного розподілу, що відповідає вибірці з вправи 1.

3. Обчисліть три числових значення неперервної випадкової величини з емпіричним розподілом, що відповідає вибірці: 5,73; 6,21; 4,95; 8,07; 7,61; 6,34; 8,12; 7,23; 9,05; 7,29; 4,51; 7,6. Вважати, що генератор випадкових чисел генерує значення 0,123, 0,477 і 0,912.

4. На мові імітаційного моделювання GPSS створіть неперервну функцію для моделювання емпіричного розподілу, що відповідає вибірці з вправи 3.

5. Обчисліть три числових значення дискретної випадкової величини з емпіричним розподілом, що відповідає вибірці 5, 6, 8, 6, 9, 7, 4, 3, 5, 6, якщо генератор випадкових чисел генерує значення 0,791, 0,669 і 0,322.

6. На мові імітаційного моделювання GPSS створіть дискретну функцію для моделювання емпіричного розподілу, що відповідає вибірці з вправи 5.

7. Обчисліть три числових значення дискретної випадкової величини з емпіричним розподілом, що відповідає вибірці 13, 14, 12, 14, 15, 11, 9, 10, 12, 16, якщо генератор випадкових чисел генерує значення 0,527, 0,903 і 0,059.

8. На мові імітаційного моделювання GPSS створіть дискретну функцію для моделювання емпіричного розподілу, що відповідає вибірці з вправи 7.

9. Обчисліть три числових значення неперервної випадкової величини з емпіричним розподілом, що відповідає згрупованій вибірці

з наступної таблиці.

[2, 3]	[3, 4]	[4, 5]	[5, 6]	[6, 7]	[7, 8]	[8, 9]	[9, 10]	[10, 11]	[11, 12]
3	2	1	0	5	6	4	2	3	2

Другий рядок таблиці містить кількість чисел, які потрапляють у відповідний проміжок з першого рядка таблиці. Вважати, що генератор випадкових чисел генерує значення 0,157, 0,812 і 0,44.

10. На мові імітаційного моделювання GPSS створіть неперервну функцію для моделювання емпіричного розподілу, що відповідає згрупованій вибірці з вправи 9.

11. Обчисліть три числових значення неперервної випадкової величини з емпіричним розподілом, що відповідає згрупованій вибірці з наступної таблиці.

[1, 2]	[2, 3]	[3, 4]	[4, 5]	[5, 6]	[6, 7]	[7, 8]	[8, 9]	[9, 10]
1	0	3	4	5	4	3	2	3

Другий рядок таблиці містить кількість чисел, які потрапляють у відповідний проміжок з першого рядка таблиці. Вважати, що генератор випадкових чисел генерує значення 0,232, 0,588 і 0,779.

12. На мові імітаційного моделювання GPSS створіть неперервну функцію для моделювання емпіричного розподілу, що відповідає згрупованій вибірці з вправи 11.

13. Обчисліть три числових значення дискретної випадкової величини з емпіричним розподілом, що відповідає згрупованій вибірці з наступної таблиці.

$x_i$	0	1	2	3	4	5	7	8
$n_i$	4	3	2	3	2	1	1	2

Вважати, що генератор випадкових чисел генерує значення 0,855, 0,091 і 0,311.

14. На мові імітаційного моделювання GPSS створіть дискретну функцію для моделювання емпіричного розподілу, що відповідає згрупованій вибірці з вправи 13.

15. Обчисліть три числових значення дискретної випадкової величини з емпіричним розподілом, що відповідає згрупованій вибірці з наступної таблиці.

$x_i$	1	2	4	5	7	10	11	12
$n_i$	2	2	5	4	3	1	2	2

Вважати, що генератор випадкових чисел генерує значення 0,148, 0,351 і 0,689.

16. На мові імітаційного моделювання GPSS створіть дискретну функцію для моделювання емпіричного розподілу, що відповідає згрупованій вибірці з вправи 15.

17. З допомогою діаграми розкиду перевірте на незалежність вибірку: 1,87; 2,13; 2,33; 2,37; 0,15; 0,64; 0,13; 0,86; 1,79; 0,82; 0,13; 2,05,  $-0,77$ ; 0,95; 0,24.

18. З допомогою діаграми розкиду перевірте на незалежність вибірку:  $-1,82$ ; 1,52; 0,56;  $-1,97$ ; 1,1; 1,08;  $-1,97$ ; 0,59; 1,51;  $-1,82$ ; 0,03; 1,82;  $-1,51$ ;  $-0,53$ ; 1,99.

19. З допомогою кореляційного графіка перевірте на незалежність вибірку з вправи 17.

20. З допомогою кореляційного графіка перевірте на незалежність вибірку з вправи 18.

21. З допомогою діаграми розкиду перевірте на незалежність вибірку:  $-0,69$ ; 1,22; 0,24; 0,94; 0,1; 2,05; 1,23; 0,94;  $-0,93$ ;  $-0,76$ ;  $-0,23$ ;  $-0,58$ ; 2,26; 2,55; 1,32.

22. З допомогою діаграми розкиду перевірте на незалежність вибірку: 1,98;  $-1,92$ ; 1,83;  $-1,68$ ; 1,53;  $-1,31$ ; 1,11;  $-0,83$ ; 0,61;  $-0,28$ ; 0,05; 0,29;  $-0,5$ ; 0,83;  $-1,01$ .

23. З допомогою кореляційного графіка перевірте на незалежність вибірку з вправи 21.

24. З допомогою кореляційного графіка перевірте на незалежність вибірку з вправи 22.

25. Методом максимальної правдоподібності знайдіть параметри логнормального розподілу.

26. Методом максимальної правдоподібності знайдіть параметри розподілу Вейбулла.

27. Методом максимальної правдоподібності знайдіть параметри біноміального розподілу.

28. Визначте сім'ю розподілів, якщо дані спостережень мають вигляд: 0,16; 0,08; 0,07; 0,61; 0,26; 1,47; 0,014; 0,2; 0,004; 0,31; 0,09; 1,06; 0,015; 0,26; 0,06; 0,32; 0,16; 0,03; 0,45; 0,086.

29. Визначте параметри розподілу для даних спостережень з вправи 28.

30. З допомогою комп'ютера побудуйте графік функції щільності розподілу, знайденого у вправі 29, поверх гістограми для даних спостережень з вправи 28. Зробіть висновки.

31. З допомогою комп'ютера виконайте частотне порівняння

для даних спостережень з вправи 28 і розподілу, знайденого у вправі 29. Зробіть висновки.

32. З допомогою комп'ютера побудуйте графік відмінностей між функціями розподілу для даних спостережень з вправи 28 і розподілу, знайденого у вправі 29. Зробіть висновки.

33. З допомогою критерію  $\chi^2$  перевірте гіпотезу про придатність розподілу, знайденого у вправі 29, для моделювання даних спостережень з вправи 28. Рівень значущості  $\alpha = 0,01$ .

34. Визначте сім'ю розподілів, якщо дані спостережень мають вигляд: 394, 206, 309, 149, 215, 96, 1557, 275, 207, 190, 510, 148, 134, 105, 37, 297, 81, 929, 147, 572.

35. Визначте параметри розподілу для даних спостережень з вправи 34.

36. З допомогою комп'ютера побудуйте графік функції щільності розподілу, знайденого у вправі 35, поверх гістограми для даних спостережень з вправи 34. Зробіть висновки.

37. З допомогою комп'ютера виконайте частотне порівняння для даних спостережень з вправи 34 і розподілу, знайденого у вправі 35. Зробіть висновки.

38. З допомогою комп'ютера побудуйте графік відмінностей між функціями розподілу для даних спостережень з вправи 34 і розподілу, знайденого у вправі 35. Зробіть висновки.

39. З допомогою критерію  $\chi^2$  перевірте гіпотезу про придатність розподілу, знайденого у вправі 35, для моделювання даних спостережень з вправи 34. Рівень значущості  $\alpha = 0,01$ .

40. Визначте сім'ю розподілів, якщо дані спостережень мають вигляд: 10,4; 10,6; 10,6; 10,1; 8,8; 8,7; 10,2; 10,5; 10,1; 12,3; 10,9; 9,0; 9,8; 8,4; 10,4; 9,6; 10,3; 11,2; 11,6; 9,7.

41. Визначте параметри розподілу для даних спостережень з вправи 40.

42. З допомогою комп'ютера побудуйте графік функції щільності розподілу, знайденого у вправі 41, поверх гістограми для даних спостережень з вправи 40. Зробіть висновки.

43. З допомогою комп'ютера виконайте частотне порівняння для даних спостережень з вправи 40 і розподілу, знайденого у вправі 41. Зробіть висновки.

44. З допомогою комп'ютера побудуйте графік відмінностей між функціями розподілу для даних спостережень з вправи 40 і розпо-

ділу, знайденого у вправі 41. Зробіть висновки.

45. З допомогою критерію  $\chi^2$  перевірте гіпотезу про придатність розподілу, знайденого у вправі 41, для моделювання даних спостережень з вправи 40. Рівень значущості  $\alpha = 0,01$ .

46. Про випадкову величину відомо лише те, що вона може набувати значень з проміжку  $[1, 5]$  з середнім значенням 2,5 і модою 2. Знайдіть параметри бета-розподілу для моделювання цієї випадкової величини.

47. Про випадкову величину відомо лише те, що вона може набувати значень з проміжку  $[3, 10]$  з середнім значенням 7 і модою 8. Знайдіть параметри бета-розподілу для моделювання цієї випадкової величини.

48. Визначте сім'ю і параметри розподілу, якщо дані спостережень мають вигляд: 14,0; 13,8; 13,2; 14,2; 13,5; 14,3; 14,3; 13,3; 13,7; 14,5; 14,3; 13,5; 14,7; 14,0; 13,8; 13,3; 14,0; 13,9; 13,9; 13,7; 14,0; 14,1; 12,9; 12,5; 13,6. Крім того, відомо, що випадкова величина не може набувати значень, менших від 10.

49. Вхідний потік деталей на виробничу дільницю є стаціонарним пуассонівським з середнім значенням 3. Обробка деталей на виробничій дільниці здійснюється у два етапи. Під час кожного етапу може оброблятися не більше 5 деталей одночасно. Час обробки деталей є випадковою величиною зі зміщеним двовимірним логнормальним законом розподілу з середнім значенням 1,9 і середньоквадратичним відхиленням 1,1 для першого етапу і з середнім значенням 2,4 і середньоквадратичним відхиленням 0,6 для другого етапу, коефіцієнт кореляції дорівнює 0,7, зміщення для першого етапу становить 1, а для другого етапу — 1,5.

Складіть модель на мові GPSS World для моделювання роботи виробничої дільниці. Для моделювання зміщеного двовимірного логнормального закону розподілу потрібно створити універсальну процедуру `lognormal2`.

Визначте середні тривалості обробки 1000 деталей на першому і на другому етапах роботи. Побудуйте і проаналізуйте графіки тривалостей обробки 20 деталей на першому і на другому етапах. Щоб можна було побачити взаємозв'язок, другий графік потрібно будувати з коефіцієнтом 0,5.

50. Час появи покупців у магазині з дев'ятої до дванадцятої години дня протягом десяти днів наведений нижче з точністю до

однієї хвилини. Цей потік є ординарним і не має післядії. Знайдіть інтенсивність цього потоку як функцію часу.

1 день. 9:04, 9:11, 9:11, 9:13, 9:15, 9:16, 9:19, 9:19, 9:21, 9:22, 9:26, 9:26, 9:28, 9:30, 9:31, 9:33, 9:34, 9:41, 9:41, 9:43, 9:44, 9:51, 9:53, 9:54, 9:54, 9:59, 10:04, 10:04, 10:05, 10:13, 10:13, 10:15, 10:18, 10:19, 10:21, 10:21, 10:21, 10:22, 10:25, 10:28, 10:40, 10:41, 10:41, 10:43, 10:45, 10:49, 11:04, 11:04, 11:06, 11:06, 11:07, 11:11, 11:12, 11:13, 11:15, 11:17, 11:20, 11:22, 11:22, 11:23, 11:24, 11:26, 11:28, 11:29, 11:30, 11:34, 11:36, 11:39, 11:40, 11:40, 11:43, 11:45, 11:47, 11:47, 11:48, 11:50, 11:50, 11:52, 11:59, 12:00.

2 день. 9:03, 9:04, 9:06, 9:08, 9:09, 9:11, 9:18, 9:20, 9:25, 9:26, 9:26, 9:27, 9:28, 9:30, 9:30, 9:30, 9:32, 9:33, 9:33, 9:34, 9:38, 9:42, 9:44, 9:45, 9:49, 9:52, 9:55, 9:56, 9:57, 9:57, 10:10, 10:10, 10:10, 10:13, 10:13, 10:13, 10:18, 10:20, 10:21, 10:23, 10:28, 10:30, 10:33, 10:34, 10:38, 10:39, 10:39, 10:39, 10:41, 10:42, 10:42, 10:43, 10:43, 10:45, 10:51, 10:54, 10:57, 10:57, 11:03, 11:03, 11:04, 11:04, 11:04, 11:05, 11:07, 11:08, 11:10, 11:13, 11:14, 11:16, 11:17, 11:17, 11:18, 11:18, 11:18, 11:18, 11:18, 11:18, 11:19, 11:20, 11:22, 11:22, 11:23, 11:24, 11:24, 11:25, 11:27, 11:37, 11:38, 11:38, 11:39, 11:43, 11:44, 11:44, 11:46, 11:47, 11:48, 11:49, 11:58, 11:58.

3 день. 9:02, 9:05, 9:13, 9:14, 9:14, 9:15, 9:21, 9:28, 9:29, 9:34, 9:36, 9:39, 9:40, 9:42, 9:43, 9:45, 9:46, 9:47, 9:47, 9:52, 9:53, 9:55, 9:59, 9:59, 10:00, 10:01, 10:04, 10:09, 10:11, 10:20, 10:20, 10:21, 10:21, 10:22, 10:24, 10:26, 10:30, 10:32, 10:34, 10:36, 10:40, 10:41, 10:42, 10:42, 10:44, 10:45, 10:45, 10:45, 10:52, 10:57, 10:58, 10:59, 11:01, 11:02, 11:02, 11:08, 11:09, 11:09, 11:10, 11:12, 11:13, 11:13, 11:14, 11:15, 11:15, 11:16, 11:17, 11:20, 11:21, 11:21, 11:22, 11:22, 11:23, 11:23, 11:24, 11:25, 11:29, 11:32, 11:33, 11:34, 11:36, 11:37, 11:38, 11:39, 11:40, 11:41, 11:41, 11:43, 11:47, 11:49, 11:51, 11:54, 11:55, 11:56, 11:59, 12:00.

4 день. 9:01, 9:02, 9:05, 9:06, 9:12, 9:13, 9:13, 9:13, 9:14, 9:16, 9:17, 9:22, 9:26, 9:28, 9:31, 9:31, 9:31, 9:33, 9:34, 9:36, 9:37, 9:37, 9:41, 9:44, 9:44, 9:46, 9:46, 9:48, 9:52, 9:56, 9:58, 10:00, 10:02, 10:02, 10:05, 10:08, 10:08, 10:09, 10:14, 10:18, 10:18, 10:20, 10:22, 10:23, 10:26, 10:28, 10:29, 10:32, 10:32, 10:32, 10:33, 10:33, 10:36, 10:40, 10:47, 10:48, 10:49, 10:50, 10:52, 10:56, 10:57, 10:57, 11:02, 11:02, 11:02, 11:04, 11:04, 11:05, 11:07, 11:07, 11:07, 11:08, 11:11, 11:13, 11:14, 11:16, 11:18, 11:19, 11:21, 11:21, 11:21, 11:22, 11:22, 11:24, 11:28, 11:31, 11:33, 11:34, 11:34, 11:36, 11:40, 11:41, 11:44, 11:46, 11:47, 11:48, 11:51, 11:53, 11:59.

5 день. 9:05, 9:06, 9:11, 9:17, 9:18, 9:20, 9:22, 9:22, 9:24, 9:28, 9:29,

9:34, 9:34, 9:35, 9:36, 9:39, 9:45, 9:45, 9:45, 9:49, 9:49, 9:50, 9:53, 9:56, 9:56, 9:58, 10:03, 10:14, 10:15, 10:17, 10:19, 10:22, 10:22, 10:23, 10:29, 10:31, 10:32, 10:38, 10:38, 10:39, 10:45, 10:46, 10:48, 10:50, 10:52, 10:52, 10:53, 10:53, 10:53, 10:54, 10:55, 10:55, 10:56, 10:58, 10:58, 11:00, 11:03, 11:05, 11:12, 11:13, 11:13, 11:14, 11:15, 11:18, 11:21, 11:25, 11:26, 11:27, 11:30, 11:31, 11:31, 11:35, 11:36, 11:39, 11:39, 11:43, 11:48, 11:49, 11:49, 11:50, 11:51, 11:52, 11:53, 11:55, 11:58.

6 день. 9:02, 9:05, 9:05, 9:07, 9:07, 9:09, 9:16, 9:16, 9:16, 9:24, 9:25, 9:29, 9:30, 9:31, 9:32, 9:32, 9:33, 9:34, 9:34, 9:39, 9:43, 9:44, 9:45, 9:48, 9:50, 9:51, 9:54, 9:56, 9:56, 10:00, 10:01, 10:07, 10:10, 10:16, 10:19, 10:23, 10:24, 10:26, 10:26, 10:27, 10:31, 10:31, 10:33, 10:34, 10:36, 10:40, 10:41, 10:42, 10:46, 10:47, 10:48, 10:50, 10:52, 10:56, 10:56, 10:57, 10:58, 10:59, 11:01, 11:01, 11:01, 11:05, 11:05, 11:07, 11:13, 11:17, 11:18, 11:20, 11:26, 11:26, 11:27, 11:27, 11:28, 11:29, 11:32, 11:33, 11:37, 11:39, 11:45, 11:46, 11:47, 11:50, 11:54, 11:55, 11:57, 12:00.

7 день. 9:00, 9:01, 9:09, 9:14, 9:18, 9:18, 9:20, 9:20, 9:26, 9:28, 9:28, 9:29, 9:36, 9:37, 9:39, 9:42, 9:54, 9:54, 9:57, 9:57, 9:58, 9:58, 10:03, 10:09, 10:10, 10:12, 10:13, 10:15, 10:18, 10:18, 10:19, 10:21, 10:26, 10:26, 10:27, 10:31, 10:33, 10:35, 10:36, 10:36, 10:37, 10:42, 10:44, 10:46, 10:48, 10:50, 10:51, 10:52, 10:54, 10:55, 10:56, 10:57, 10:57, 10:58, 10:59, 11:00, 11:00, 11:01, 11:01, 11:02, 11:03, 11:04, 11:05, 11:07, 11:07, 11:08, 11:10, 11:10, 11:10, 11:11, 11:15, 11:15, 11:19, 11:19, 11:21, 11:21, 11:26, 11:27, 11:27, 11:36, 11:36, 11:36, 11:38, 11:39, 11:39, 11:40, 11:41, 11:45, 11:46, 11:51, 11:52, 11:52, 11:54, 11:58, 11:59.

8 день. 9:06, 9:06, 9:08, 9:09, 9:14, 9:15, 9:33, 9:34, 9:34, 9:35, 9:36, 9:37, 9:40, 9:43, 9:44, 9:46, 9:47, 9:48, 9:49, 9:50, 9:51, 9:52, 9:54, 9:55, 9:56, 9:58, 10:00, 10:01, 10:02, 10:02, 10:06, 10:08, 10:09, 10:09, 10:12, 10:17, 10:18, 10:25, 10:25, 10:27, 10:27, 10:30, 10:31, 10:31, 10:34, 10:34, 10:37, 10:39, 10:41, 10:42, 10:44, 10:45, 10:47, 10:55, 10:56, 10:56, 10:57, 10:57, 11:06, 11:14, 11:15, 11:15, 11:16, 11:17, 11:17, 11:18, 11:18, 11:19, 11:22, 11:24, 11:25, 11:26, 11:26, 11:27, 11:28, 11:29, 11:34, 11:36, 11:38, 11:43, 11:45, 11:46, 11:47, 11:47, 11:50, 11:52, 11:55, 11:56, 11:58, 11:58.

9 день. 9:05, 9:05, 9:05, 9:06, 9:10, 9:15, 9:17, 9:17, 9:18, 9:19, 9:21, 9:26, 9:26, 9:35, 9:36, 9:41, 9:42, 9:43, 9:45, 9:46, 9:48, 9:48, 9:49, 9:49, 9:50, 9:50, 9:51, 9:52, 9:53, 9:54, 9:54, 9:56, 9:58, 10:04, 10:09, 10:19, 10:21, 10:22, 10:23, 10:34, 10:34, 10:35, 10:36, 10:37, 10:38, 10:42, 10:47, 10:47, 10:47, 10:48, 10:48, 10:49, 10:50, 10:51, 10:51, 10:51, 10:52, 10:53, 10:55, 10:57, 10:58, 10:59, 10:59, 11:01, 11:02, 11:02, 11:04, 11:06, 11:09,

11:10, 11:17, 11:18, 11:19, 11:20, 11:22, 11:22, 11:22, 11:30, 11:33, 11:35, 11:36, 11:36, 11:48, 11:53, 11:54, 11:56, 11:56, 11:56, 11:57.

10 день. 9:01, 9:02, 9:04, 9:07, 9:07, 9:11, 9:13, 9:18, 9:18, 9:19, 9:20, 9:22, 9:28, 9:30, 9:36, 9:38, 9:39, 9:42, 9:43, 9:48, 9:50, 9:54, 9:55, 10:02, 10:04, 10:10, 10:10, 10:11, 10:13, 10:19, 10:28, 10:28, 10:28, 10:29, 10:36, 10:36, 10:37, 10:39, 10:40, 10:41, 10:41, 10:44, 10:44, 10:46, 10:49, 10:55, 10:57, 10:58, 11:00, 11:03, 11:07, 11:08, 11:09, 11:10, 11:10, 11:11, 11:11, 11:13, 11:13, 11:19, 11:20, 11:21, 11:24, 11:25, 11:25, 11:26, 11:27, 11:27, 11:32, 11:32, 11:34, 11:34, 11:38, 11:39, 11:45, 11:45, 11:46, 11:47, 11:52, 11:57.

### Критичні точки розподілу $\chi^2$

Для рівня значущості  $\alpha = 0,01$  має місце наступна таблиця.

Кількість ступенів свободи, $k$	$\chi_{k,\alpha}^2$	Кількість ступенів свободи, $k$	$\chi_{k,\alpha}^2$
1	6,6	11	24,7
2	9,2	12	26,2
3	11,3	13	27,7
4	13,3	14	29,1
5	15,1	15	30,6
6	16,8	16	32,0
7	18,5	17	33,4
8	20,1	18	34,8
9	21,7	19	36,2
10	23,2	20	37,6

## Розділ 10. Планування експерименту

### § 10.1. Перехідний і стаціонарний режими

Планування експерименту — це розробка такого плану експерименту, який дає можливість за мінімальної кількості прогонів моделі і за мінімальних затрат ресурсів зробити статистично значимі висновки щодо функціонування системи.

Якщо розглядати досить тривалий період моделювання систем масового обслуговування, то, як правило, можна виділити так званий період «розігріву» чи «розгону» моделі, або перехідний режим роботи, після якого модель переходить у стаціонарний (сталий) режим роботи. Врахування даних перехідного режиму для оцінювання результатів моделювання спричинює зміщення статистичних оцінок параметрів. Отже, якщо тривалість перехідного режиму є відомою, то всі дані спостереження за моделлю протягом нього треба відкинути. У середовищі GPSS World це можна зробити командою **RESET**, яка знищує всю накопичену статистику, але залишає транзакти в чергах і пристроях. Звичайно, для відкидання накопиченої статистики треба знати тривалість перехідного режиму. Досі не існує цілком надійних методів визначення цієї тривалості.

Для визначення тривалості перехідного режиму можна побудувати графіки залежностей від часу деяких параметрів, таких як коефіцієнти завантаження пристроїв, довжини черг та інше. Перехідний режим триває, доки формуються черги, встановлюються певні значення коефіцієнтів завантаження тощо. На рис. 10.1, 10.2 наведено приклади графіків залежностей коефіцієнтів завантаження пристроїв або інших параметрів від часу  $t$ . Добре видно, що ці величини спочатку зростають до деякого значення, а потім протягом решти часу є майже незмінними. На рис. 10.1, 10.2 величини  $t_{\text{пер}}$  позначають кінець перехідного режиму. Для надійності потрібно здійснити  $n$  запусків моделі (з різними випадковими числами) і з

усіх знайдених із графіків тривалостей перехідного режиму вибрати найдовшу.

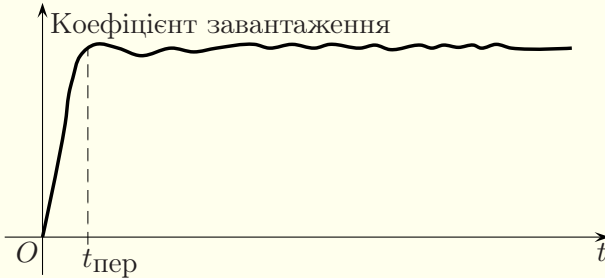


Рис. 10.1.

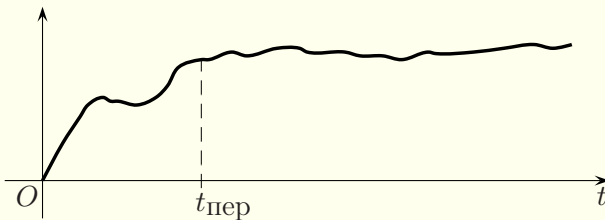


Рис. 10.2.

Для збору статистичної інформації потрібно здійснити моделювання тривалістю не меншою за  $100 t_{\text{пер}}$ , якщо, звичайно, мова не йде про моделювання протягом якогось певного часу (моделювання за таймером) чи до настання якоїсь певної події (наприклад, відсутності вимог, чи обробки фіксованої кількості вимог). Статистичну інформацію протягом перехідного режиму роботи моделі ігнорують.

Якщо є підстави вважати, що перехідний режим роботи моделі є відсутнім чи суттєво не впливає на статистичні оцінки, то для перевірки цієї гіпотези здійснюють два запуски моделі. Якщо при першому запуску моделі черги є порожніми, а пристрої — вільними (звичайна ситуація на початку моделювання), то при другому запуску на початку моделювання у

чергах є вимоги, а пристрої є зайнятими. У мові GPSS останнє можна реалізувати з допомогою двох додаткових блоків GENERATE і TRANSFER, наприклад:

```
GENERATE ,,N  
TRANSFER ,mitka
```

У цьому прикладі  $N$  — кількість транзактів, які генеруються у початковий момент моделювання і передаються на мітку *mitka*. Число  $N$  не повинно бути надто великим, бо інакше можемо отримати перехідний режим роботи моделі при її другому запуску.

Якщо під час моделювання в обох випадках буде отримано близькі результати, то це, як правило, свідчатиме про несуттєвий вплив перехідного режиму роботи моделі на статистичні оцінки.

Аналітичне моделювання систем масового обслуговування в розділі 5 вивчалось переважно для стаціонарного режиму роботи.

## § 10.2. Визначення кількості реалізацій процесу моделювання для оцінювання ймовірності

Часто моделювання здійснюють для визначення ймовірності настання певної події. Для оцінювання цієї ймовірності з наперед заданою точністю потрібно здійснити достатню кількість реалізацій процесу моделювання. Для систем масового обслуговування кількість реалізацій процесу моделювання визначається кількістю оброблених вимог, пов'язаною з тривалістю модельованого процесу.

Припустимо, що у кожній з  $n$  реалізацій процесу настання події  $A$  випадкова величина  $\xi$  набуває значення  $x_1^* = 1$  з невідомою ймовірністю  $p$  (подія  $A$  настає) і  $x_2^* = 0$  з ймовірністю  $1 - p$  (подія  $A$  не настає). Математичне сподівання  $M(\xi)$  і дисперсію  $D(\xi)$  випадкової величини  $\xi$  можна визначити за

формулами

$$\begin{aligned} M(\xi) &= x_1^* p + x_2^* (1 - p) = p, \\ D(\xi) &= (x_1^* - M(\xi))^2 p + (x_2^* - M(\xi))^2 (1 - p) = \\ &= (1 - p)^2 p + p^2 (1 - p) = p(1 - p). \end{aligned}$$

Як оцінку ймовірності  $p$  використовують частоту  $\bar{x}$  настання події  $A$ . Її обчислюють за формулою

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i, \quad (10.1)$$

де  $x_i$  визначає настання (якщо  $x_i = 1$ ) або ненастання (якщо  $x_i = 0$ ) події  $A$  в  $i$ -й реалізації.

Величина  $\bar{x}$  також є випадковою величиною і відрізняється від імовірності  $p$  менше, ніж на  $\varepsilon$ , тобто

$$|p - \bar{x}| < \varepsilon,$$

де  $\varepsilon$  — точність або похибка оцінки. Нехай імовірність виконання цієї нерівності

$$P(|p - \bar{x}| < \varepsilon) \geq \gamma.$$

У математичній статистиці  $\varepsilon$  — це половина довірчого інтервалу, а  $\gamma$  — рівень надійності оцінки.

З курсу теорії ймовірностей відомо, що

$$M(C\xi) = CM(\xi), \quad M(\xi_1 + \xi_2) = M(\xi_1) + M(\xi_2), \quad (10.2)$$

$$D(C\xi) = C^2 D(\xi), \quad D(\xi_1 + \xi_2) = D(\xi_1) + D(\xi_2), \quad (10.3)$$

де  $C$  — довільна стала,  $\xi_1$  і  $\xi_2$  — незалежні випадкові величини. Тоді

$$M(\bar{x}) = \frac{1}{n} np = p, \quad D(\bar{x}) = \frac{1}{n^2} np(1 - p) = \frac{p(1 - p)}{n}. \quad (10.4)$$

Згідно з центральною граничною теоремою випадкова величина (10.1) при достатньо великих значеннях  $n$  має розподіл, близький до нормального.

З курсу математичної статистики відомо, що

$$\Phi\left(\frac{\varepsilon}{\sqrt{D(\bar{x})}}\right) = \frac{1 + \gamma}{2}, \quad (10.5)$$

де  $\Phi(x)$  — функція стандартного нормального розподілу (з нульовим середнім значенням і одиничним середньоквадратичним відхиленням),

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{x^2}{2}} dx.$$

Тоді з формул (10.4) і (10.5) випливає формула

$$\sqrt{\frac{n}{p(1-p)}} \cdot \varepsilon = \Phi^{-1}\left(\frac{1 + \gamma}{2}\right),$$

де  $\Phi^{-1}(x)$  — обернена функція до функції розподілу  $\Phi(x)$ . Звідси

$$n = \left[ \left( \Phi^{-1}\left(\frac{1 + \gamma}{2}\right) \right)^2 \frac{p(1-p)}{\varepsilon^2} \right] + 1, \quad (10.6)$$

де квадратні дужки позначають цілу частину числа.

Оскільки ймовірність  $p$  заздалегідь невідома, то спочатку здійснюють близько ста випробувань, по ним наближено оцінюють імовірність  $p$  за формулою (10.1) і підставляють у формулу (10.6) для визначення числа  $n$  — кількості реалізацій процесу моделювання. Значення  $\Phi^{-1}\left(\frac{1+\gamma}{2}\right)$  знаходять за таблицями стандартного нормального розподілу або з допомогою системи комп'ютерної математики, наприклад, такої як Maple.

**Приклад.** Визначити потрібну кількість реалізацій процесу моделювання для обчислення ймовірності настання події з точністю  $\varepsilon = 0,005$  при рівні надійності  $\gamma = 0,99$ , якщо ймовірність  $p \approx 0,7$ .

**Розв'язання.** Оскільки

$$\Phi^{-1}\left(\frac{1+\gamma}{2}\right) = \Phi^{-1}(0,995) \approx 2,58,$$

то за формулою (10.6) кількість реалізацій

$$n = \left[ 2,58^2 \cdot \frac{0,7 \cdot (1 - 0,7)}{0,005^2} \right] + 1 = 55914. \blacksquare$$

### § 10.3. Визначення кількості реалізацій процесу моделювання для оцінювання середнього значення

Для оцінювання середнього значення деякої величини  $z$  наперед заданою точністю, як і для оцінювання ймовірності, потрібно достатню кількість разів реалізувати процес моделювання.

Припустимо, що в  $i$ -й реалізації процесу моделювання випадкова величина  $\xi$  набуває значення  $x_i$ . Нехай ця випадкова величина має математичне сподівання  $a$  і дисперсію  $\sigma^2$ . Як оцінку математичного сподівання  $a$  використаємо середнє арифметичне:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i,$$

де  $n$  — кількість реалізацій процесу моделювання.

Величина  $\bar{x}$  також є випадковою величиною і відрізняється від математичного сподівання  $a$  менше, ніж на  $\varepsilon$ , тобто  $|a - \bar{x}| < \varepsilon$ , де  $\varepsilon$  — точність або похибка оцінки. Нехай ймовірність виконання цієї нерівності

$$P(|a - \bar{x}| < \varepsilon) \geq \gamma,$$

де  $\gamma$  — рівень надійності оцінки.

Користуючись формулами (10.2) і (10.3), обчислюємо математичне сподівання і дисперсію випадкової величини  $\bar{x}$ :

$$M(\bar{x}) = \frac{1}{n}na = a, \quad D(\bar{x}) = \frac{1}{n^2}n\sigma^2 = \frac{\sigma^2}{n}. \quad (10.7)$$

Згідно з центральною граничною теоремою випадкова величина  $\bar{x}$  при достатньо великих значеннях  $n$  має розподіл, близький до нормального.

З формул (10.5) і (10.7) випливає формула

$$\frac{\sqrt{n}}{\sigma}\varepsilon = \Phi^{-1}\left(\frac{1+\gamma}{2}\right),$$

де  $\Phi^{-1}(x)$  — обернена функція до функції розподілу  $\Phi(x)$ . Тоді

$$n = \left[ \left( \Phi^{-1}\left(\frac{1+\gamma}{2}\right) \right)^2 \frac{\sigma^2}{\varepsilon^2} \right] + 1, \quad (10.8)$$

де квадратні дужки позначають цілу частину числа.

Оскільки дисперсія  $\sigma^2$  випадкової величини заздалегідь невідома, то спочатку здійснюють близько ста випробувань, по яким наближено оцінюють дисперсію  $\sigma^2$  і підставляють її у формулу (10.8) для визначення числа  $n$  — кількості реалізацій процесу моделювання. Значення  $\Phi^{-1}\left(\frac{1+\gamma}{2}\right)$  знаходять за таблицями стандартного нормального розподілу або з допомогою системи комп'ютерної математики.

**Приклад.** Визначити потрібну кількість реалізацій процесу моделювання для обчислення середнього значення з точністю  $\varepsilon = 0,001$  при рівні надійності  $\gamma = 0,98$ , якщо дисперсія  $\sigma^2 \approx 1,2$ .

**Розв'язання.** Оскільки

$$\Phi^{-1}\left(\frac{1+\gamma}{2}\right) = \Phi^{-1}(0,99) \approx 2,33,$$

то за формулою (10.8) кількість реалізацій

$$n = \left\lceil \frac{2,33^2 \cdot 1,2}{0,001^2} \right\rceil + 1 = 6514681. \blacksquare$$

### Питання до розділу 10

1. Яка різниця між перехідним і стаціонарним режимами моделювання?
2. Як можна визначити тривалість перехідного режиму?
3. Як визначити кількість реалізацій процесу моделювання для відшукування ймовірності з заданою точністю?
4. Як визначити кількість реалізацій процесу моделювання для відшукування середнього значення з заданою точністю?

### Вправи до розділу 10

Для відшукування значення оберненої функції до функції розподілу в наступних задачах пропонується скористатись системою комп'ютерної математики.

1. Визначте потрібну кількість реалізацій процесу моделювання для обчислення ймовірності настання події з точністю  $\varepsilon = 0,002$  при рівні надійності  $\gamma = 0,95$ , якщо ймовірність  $p \approx 0,4$ .
2. Визначте потрібну кількість реалізацій процесу моделювання для обчислення середнього значення з точністю  $\varepsilon = 0,001$  при рівні надійності  $\gamma = 0,99$ , якщо дисперсія  $\sigma^2 \approx 2,5$ .

### Значення функції $\Phi^{-1}(x)$

У таблиці подані значення оберненої функції до функції стандартного нормального розподілу для деяких значень аргументу.

$x$	$\Phi^{-1}(x)$	$x$	$\Phi^{-1}(x)$	$x$	$\Phi^{-1}(x)$
0,9	1,28	0,96	1,75	0,98	2,05
0,925	1,44	0,965	1,81	0,985	2,17
0,95	1,64	0,97	1,88	0,99	2,33
0,955	1,70	0,975	1,96	0,995	2,58

## Розділ 11. Прогнозування систем

### § 11.1. Основні методи прогнозування систем

Однією з основних цілей моделювання систем є прогнозування їхнього розвитку. Розглянемо коротко основні аспекти прогнозування розвитку систем.

Системи можна ділити на ритмічні і неритмічні. Особливістю ритмічних систем є те, що вони в процесі свого розвитку повторюють раніше пройдені траєкторії, що робить їх практично передбачуваними. Прикладами ритмічних процесів є цикли сонячної активності, хвилі економічної кон'юнктури, сезонні коливання клімату та багато іншого. Зрозуміло, що прогнозувати розвиток ритмічних систем набагато простіше, ніж неритмічних.

Прогнози бувають довгостроковими і короткостроковими. Довгострокові прогнози спрямовані на передбачення якісних змін системи і середовища. Короткострокові прогнози орієнтовані на передбачення у майбутньому розвитку існуючих закономірностей змін.

Для довгострокових прогнозів використовують експертні і логічні методи. У першому випадку прогнози роблять експерти на основі власного досвіду й інтуїції. Логічні методи базуються на проведенні аналогій функціонування розглядуваної системи з історією функціонування інших систем.

Для короткострокових прогнозів використовують математичне моделювання, методи екстраполяції і статистичні методи.

Математичні моделі за умови відсутності змін основних факторів дозволяють прогнозувати процеси, об'єкти і явища.

Під екстраполяцією розуміють відшукання значень функцій поза межами проміжку, який містить відомі значення функції. Як і у випадку інтерполяції тут використовують інтерполяційні многочлени чи інші функції.

Статистичні методи прогнозування спираються на теорію ймовірностей, математичну статистику, теорію випадкових процесів. Часто будують прогнози на основі функцій наближення даних спостереження методом найменших квадратів.

Прогнозування ймовірнісних систем може бути лише наближеним: загальним, а не конкретним, тобто можна визначити середню чи максимальну довжину черги, а не довжину черги у конкретний час.

### § 11.2. Точки біфуркації

Серйозною проблемою є прогнозування детерміновано-ймовірнісних систем. В одних випадках вони ведуть себе детерміновано, а в інших — ймовірно. Якщо точніше, головна особливість систем цього класу полягає в їх здатності самостійно «визирати» поведінку в процесі функціонування. Вибір поведінки відбувається не постійно, а в особливих областях, які називають біфуркаціями (від лат. bifurcus — роздвоєння).

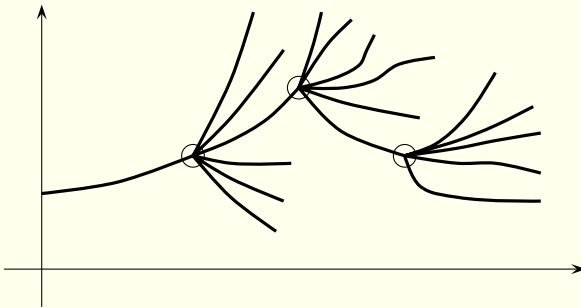


Рис. 11.1.

В математичному сенсі *біфуркація* — це виникнення при деякому критичному значенні параметра нового розв'язку системи диференціальних рівнянь, які описують розвиток системи (неєдиність розв'язку). На рис. 11.1 наведені три послідовні точки біфуркації. У цих точках система ніби сама вибирає

новий напрям свого розвитку. Після вибору розвиток системи детермінується до наступної точки біфуркації. У цій точці знову відбувається вибір і процес повторюється.

У загальному випадку точно передбачити моменти виникнення біфуркацій і особливо результати вибору неможливо ні при як завгодно глибокому аналізі системи, ні при як завгодно довгому спостереженні за нею. Більше того, точки біфуркації і вибір варіантів траєкторії розвитку не відомі самій системі.

Біфуркацію можна назвати кризою так, як говорять про кризу хвороби чи кризу у суспільстві (революції, повстання, перевороти теж є кризами). В економіці точки біфуркації теж відповідають кризам.

Прогнозування таких систем між точками біфуркації можливе різними методами. Для прогнозування вибору шляху розвитку системи в точці біфуркації апарат диференціальних рівнянь не годиться, бо є порушення єдиності розв'язку. Статистичний підхід теж не підходить, бо в момент біфуркації невідомою є повна множина можливих станів системи.

Дуже багато складних систем мають точки біфуркації. Тому довгострокові прогнози для них — річ дуже ненадійна.

### § 11.3. Детермінований хаос

Поняття детермінованого хаосу є порівняно новим у теорії систем — воно з'явилося у 70-х роках ХХ століття. Хаотичні системи є окремим підкласом детермінованих систем. По можливостям прогнозування розвитку хаотичні системи суттєво відрізняються від звичайних детермінованих і ймовірнісних систем, нагадуючи собою детерміновано-ймовірнісні системи. Якщо в звичайній детермінованій системі, знаючи закони її розвитку і початковий стан, майбутню траєкторію системи можна передбачити на як завгодно великий час вперед, а в імовірнісній моделі точний прогноз взагалі неможливий на будь-який час, то в хаотичній системі помилка прогнозу зростає швидко, але можливий прогноз на обмежений час вперед

з допустимою помилкою. У детерміновано-ймовірнісних системах прогноз на невеликий час вперед до точки біфуркації може бути точним.

В теорії диференціальних рівнянь атрактор — це обмежена множина точок фазового простору, до якої при  $t \rightarrow \infty$  прямують всі фазові траєкторії динамічної системи. Динамічна система у вигляді системи диференціальних чи різнице-вих рівнянь називається *хаотичною*, якщо при її русі у фазовому просторі існує атрактор і цей атрактор є дивним, тобто всі траєкторії, що належать атрактору, є нестійкими за Ляпуновим, незважаючи на те, що жодна з цих траєкторій область атрактора не залишає. Дивний атрактор має фрактальну структуру.

Виявляється, що неможливо передбачити поведінку хаотичної системи, тобто абсолютно детермінованої системи, в якій немає випадковостей і якій відповідає адекватна абсолютно детермінована модель у вигляді диференціальних рівнянь, розв'язки яких існують і єдині. Справа в тому, що для передбачення поведінки такої системи треба знати початковий стан з безмежно великою точністю, тобто не просто з малою похибкою  $\varepsilon$  між початковими координатами реального об'єкта і його моделі, а саме з точністю  $\varepsilon = 0$ , у протилежному разі за рахунок цієї малої похибки модель системи і сама система, розвиваючись спочатку однаково, будуть вести себе далі по-різному.

У 1963 році метеоролог Едвард Лоренц показав, що якісний характер явищ атмосферної турбулентності, які описуються складними рівняннями з частинними похідними Нав'є–Стокса, може бути переданий простою нелінійною моделлю трьох рівнянь:

$$\begin{cases} \frac{dx}{dt} = \sigma(y - x), \\ \frac{dy}{dt} = rx - y - xz, \\ \frac{dz}{dt} = xy - bz. \end{cases}$$

На рис. 11.2 наведено траєкторії останньої системи для

$\sigma = 8$ ,  $b = 2$  і  $r = 40$ . Результати показують, що в системі не спостерігається тенденції до встановлення якогось простого стаціонарного чи періодичного режиму. Траєкторії розміщуються всередині деякої обмеженої області, своїми коливаннями її заповнюють, математичні сподівання і дисперсії фазових координат  $x$ ,  $y$ ,  $z$  практично стаціонарні. Це нагадує поведінку ймовірнісної системи, хоча модель є детермінованою і точок біфуркації немає. Ця модель є прикладом детермінованого хаосу.

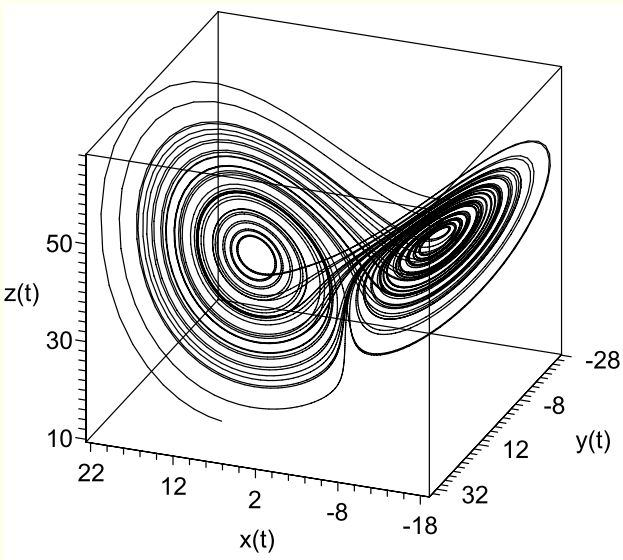


Рис. 11.2.

Ще одним прикладом системи з детермінованим хаосом є круглий більярд. Звичайний прямокутний більярд з абсолютно пружними ударами і дзеркальним законом відбиття (тобто кінетична енергія кулі не змінюється від зіткнення з бортом більярда, а кути падіння і відбиття рівні) веде себе як детермінована система і не має суттєвої залежності від початкових умов (рис. 11.3). Круглий стіл з тими самими законами

суттєво залежить від початкових умов (рис. 11.4).

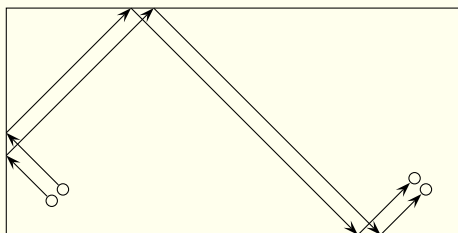


Рис. 11.3.

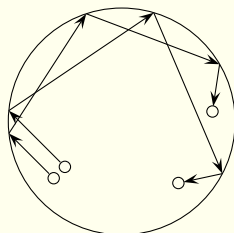


Рис. 11.4

Складні неритмічні системи зазвичай мають точки біфуркації чи детермінований хаос, тому їх прогнозування на тривалий період є складним або навіть неможливим.

### Питання до розділу 11

1. Які методи використовують для короткострокового прогнозування?
2. Які особливості довгострокових прогнозів?
3. Що таке точки біфуркації? Якими є особливості прогнозування систем з біфуркаціями?
4. Що таке детермінований хаос? Якими є особливості прогнозування систем з детермінованим хаосом?

### Список рекомендованої літератури

1. Бартіш М. Я., Дудзяний І. М. Дослідження операцій. Ч. 5: Моделі з чинником часу. Львів : ЛНУ імені Івана Франка, 2012. 256 с.
2. Жерновий Ю. В. Імітаційне моделювання систем масового обслуговування : Практикум. Львів : Вид. центр ЛНУ ім. І. Франка, 2007. 307 с.
3. Жерновий Ю. В. Марковські моделі масового обслуговування : Тексти лекцій. Львів : Видавничий центр ЛНУ ім. Івана Франка, 2004. 154 с.
4. Жученко А. І., Ладієва Л. Р., Піргач М. С., Жураковський Я. Ю. Математичне моделювання процесів і систем. Київ : КПІ ім. Ігоря Сікорського, 2021. 351 с.
5. Махней О. В. Лабораторний практикум з імітаційного моделювання у GPSS: методичні рекомендації до проведення лабораторних занять. Ч. 1. 2-ге вид. Івано-Франківськ : Голіней, 2020. 40 с.
6. Махней О. В. Лабораторний практикум з імітаційного моделювання у GPSS: методичні рекомендації до проведення лабораторних занять. Ч. 2. Івано-Франківськ : Голіней, 2012. 32 с.
7. Махней О. В. Практикум з математичного моделювання. Івано-Франківськ : Голіней, 2022. 172 с.
8. Маценко В. Г. Математичне моделювання. Чернівці : Чернівецький нац. ун-т, 2014. 519 с.
9. Неруш В. Б., Курдеча В. В. Імітаційне моделювання систем та процесів. К. : НН ІТС НТУУ «КПІ», 2012. 115 с.
10. Ніколюк П. К. Моделювання систем. Вінниця : ДонНУ, 2023. 228 с.

11. Соколовський Я. І., Шабатура Ю. В., Виклюк Я. І. та ін. Моделювання систем в середовищі GPSS World. Львів : Новий світ – 2000, 2025. 288 с.
12. Стеценко І. В. Моделювання систем. Черкаси : ЧДТУ, 2009. 399 с.
13. Томашевський В. М. Моделювання систем. К. : Видавнича група ВНУ, 2005. 352 с.
14. Хусаїнов Д. Я., Харченко І. І., Шатирко А. В. Введення в моделювання динамічних систем. К. : КНУ, 2010. 132 с.
15. Шваліковський Дмитро. CAS Maxima: основи роботи. Луцьк : Вежа-Друк, 2022. 106 с.
16. Amel'kin V. V. Differential Equations in Applications. Mir, 1987. 286 p.
17. Knuth Donald E. The Art of Computer Programming. Vol. 3. 2nd ed. Addison-Wesley, 1998. 803 p.
18. Law Averill M. Simulation Modeling and Analysis. McGraw-Hill, 2014. 800 p.
19. Maxima Manual.

## Предметний покажчик

- Автомат клітинний 55
- Мілі 51
  - Мура 52
- Атрибут системний числовий 149
- Безпечність 76
- Біфуркація 371
- Блок 148
- Вузол 69
- Гістограма 306
- Графік кореляційний 303
- Діаграма розкиду 302
- Досяжність розмітки 75
- Дуга 69
- заперечення 90
  - інформаційна 90
- Емерджентність 11
- Живість 75
- потенційна 75
- Змінна користувача 238
- Значення розмітки 73
- Імовірність відмови 118, 123
- Імовірність фінальна 116, 121, 128
- Інваріант вузлів 79
- переходів 79
- Інтенсивність зведена 117, 122, 129
- потоку 101, 349
- Кількість середня вимог у черзі 125, 130, 184
- – зайнятих каналів 118, 123, 129, 183
- Коефіцієнт завантаження 119, 124, 130, 182, 183
- Команда керування 147, 156, 220
- опису 147, 162–171, 187, 215, 238
- Комірка пам'яті 170
- Консервативність 76
- строга 77
- Лінія пружна 44
- Маркер 69
- Матриця виходів 74
- входів 74
  - змінювань 74
- Мережа Петрі 69
- – некласична 88
  - –  $p$ -інваріантна 80
  - –  $t$ -інваріантна 80
- Метод максимальної правдоподібності 317
- оберненої функції 263
- Мітка 151, 245
- Модель 13
- Моделювання 13
- аналітичне 18, 24, 69, 100
  - випадкових величин 267–277, 290–301, 332–353
  - – подій 262
  - імітаційне 18, 57, 94, 144, 256
  - комбіноване 20, 238
  - комп'ютерне 20
  - математичне 17
  - пуассонівського потоку 104, 349
- Обмеженість 76
- Оборотність 76
- Оператор переходу на мітку 245
- повернення значення 245
  - присвоювання 244
  - розгалуження 244

- циклу 245
- Параметр транзакта 158
- Перемикач логічний 148, 170
- Перехід 69
  - дозволений 70
- Підсистема 11
- Повторюваність 76
  - стаціонарна 76
- Порівняння частотне 325
- Потік подій 101
  - без післядії 102
  - випадковий 102
  - детермінований 101
  - найпростіший 102
  - ординарний 102
  - регулярний 101
  - стаціонарний 102
  - стаціонарний пуассонівський 102
- Пристрій багатоканальний 104, 148, 162
  - одноканальний 104, 148, 159, 161
- Пріоритет транзакта 159
- Режим переривань 161
  - пріоритетів 162
- Рівняння фундаментальне 78
- Розмітка 72, 73
  - внутрішня 83
  - гранична 83
  - дублююча 83
  - термінальна 83
- Розподіл бета 193, 308, 340
  - біноміальний 193, 268
  - Вейбулла 193, 275, 309
  - гамма 193, 308
  - геометричний 193, 267
  - експоненціальний 193, 271, 309
  - емпіричний 290
- Ерланга 272, 309
- зміщений 334
- зрізаний 337
- логнормальний 193, 274, 309, 348
- нормальний 193, 273, 309, 344
- Пуассона 193, 270
- рівномірний 193, 267, 271, 309
- трикутний 193, 275
- Система 11
  - балансу потоків вимог 135
  - масового обслуговування 100
  - складна 13
  - станів мережі Петрі 78
- Сім'я транзактів 214
- Стійкість 76
- Таблиця 164
- Термінальність 76
- Транзакт 148
  - активний 158
- Формула Літгла 112
- Формули Ерланга 118
- Хаос детермінований 372
- Час середній очікування в черзі 125, 130, 184
- Черга 105, 166, 233
- Символи, ключові слова та інші терміни мови і середовища GPSS World:
  - \$ 150
  - ~ 153
  - # 153
  - / 153, 187
  - \ 153
  - @ 154
  - +, – 154
  - = 154, 244

- <, >, <=, >= 154  
\* 152, 219  
; 152, 243  
& 154  
| 154  
A1 217  
ABS 192  
AC1 216  
ADOPT 214  
ADVANCE 157  
ALL 173, 236  
AND 154  
ASSEM 186  
ASSEMBLE 214  
ASSIGN 158  
ATN 192  
AVAIL. 182  
AVE. (-0) 184  
AVE.C. 183  
AVE.CONT. 184, 238  
AVE.TIME 182, 184, 238  
AVL. 183  
BACK 236  
BDT 186  
BEGIN 243  
Beta 193  
Binomial 193  
BLOCK TYPE 182  
BLOCKS 180  
BOTH 173  
BV 169, 219  
BVARIABLE 169  
C 188  
C1 216  
CA 235  
CAP. 183  
CC 235  
CEC XN 186  
CH 235  
CLEAR 220  
CM 235  
CONT. 184  
CONTINUE 220  
COS 192  
Create Simulation 145  
CT 235  
CUM.% 185  
CURRENT 186  
CURRENT COUNT 182  
D 188  
DELAY 183, 183  
DEPART 167  
DISPLACE 231  
DO 245  
DUniform 193  
E 154, 176, 189  
ELSE 244  
END 243  
END TIME 180  
ENTER 163  
ENTRIES 182, 183, 238  
ENTRY 184  
ENTRY COUNT 182  
ENTRY(0) 184  
EQU 238  
EXECUTE 250  
EXP 192  
Exponential 193  
F 217  
FACILITIES 180  
FACILITY 182  
FAVAIL 231  
FC 217  
FEC XN 186  
FI 217  
FIFO 234

- FN 188, 219  
FNV 177  
FR 217  
FREQUENCY 185  
FT 218  
FUNAVAIL 231  
FUNCTION 187  
FV 177, 217  
FVARIABLE 169  
G 154, 176  
Gamma 193  
GATE 177  
GATHER 214  
GE 154, 176  
GENERATE 155  
Geometric 193  
GOTO 245  
HALT 146, 220  
I 170, 177  
IF 244  
INCLUDE 222  
INITIAL 170, 171, 215  
Insert GPSS Blocks 152  
INT 192  
INTEGRATE 239  
INTEGRATION 242  
INTER 183  
L 154, 176, 189  
LABEL 181  
LE 154, 176  
LEAVE 163  
LIFO 234  
LINK 234  
LOC 181  
LOG 192  
LOGIC 170  
LOGICSWITCH 185  
LogNormal 193  
LOOP 179  
LR 178  
LS 170, 178, 219  
M 178, 189  
M1 164, 186, 217  
MARK 164  
MATCH 178, 214  
MATRIX 215, 216  
MAX 183, 184, 238  
MEAN 185  
MIN 183  
MP 164, 217  
MSAVEVALUE 216  
MX 215, 219  
N 217  
NAME 180  
NE 154, 176  
NEXT 186  
Next Error 146  
NI 177  
NM 178  
Normal 193  
NU 177  
OR 154  
OWNER 182  
P 158, 175, 217  
PARAMETER 186  
PEND 182  
PICK 174  
PLUS 246  
Poisson 193  
PR 159, 162, 217  
PREEMPT 161  
PRI 186  
PRIORITY 159  
PROCEDURE 243  
Q 167, 218  
QA 218

- QC 218  
QM 218  
QT 218  
QTABLE 167  
QUEUE 166, 184  
QueryXNAssemblySet 250  
QueryXNExist 250  
QueryXNM1 250  
QueryXNParameter 250  
QueryXNPriority 250  
QX 219  
QZ 218  
R 170, 218  
RANGE 185  
RE 162  
RELEASE 160, 162  
REM. 183  
RESET 220  
RETRY 182–185, 238  
RETURN 160, 162, 245  
RMULT 222  
RN 186, 216  
S 170, 218  
SA 218  
SAVAIL 231  
SAVEVALUE 171, 185  
SBR 175  
SC 218  
SE 178, 218  
SEIZE 159  
Settings 187  
SF 177, 218  
SHOW 222  
Simulation Snapshot 228  
Simulation Window 222  
SIN 192  
SIZE 238  
SM 218  
SNE 178  
SNF 178  
SNV 178  
SPLIT 213  
SQR 192  
SR 218  
ST 218  
START 156, 220  
START TIME 180  
STD.DEV. 185  
STEP 220  
STOP 220  
STORAGE 162, 183  
STORAGES 180  
SUNAVAIL 231  
SV 178, 218  
TABLE 164, 185  
TABULATE 164  
TAN 192  
TB 219  
TC 219  
TD 219  
TEMPORARY 243  
TEMPORARY MATRIX 243  
TERMINATE 156  
TEST 176  
THEN 244  
TG1 217  
TRANSFER 172  
Triangular 193  
U 177  
UNLINK 236  
Uniform 193  
UNSPECIFIED 181  
USER CHAIN 238  
UTIL. 182, 183  
V 169, 219  
VALUE 181, 185, 186

---

VARIABLE	169	X	171, 219
W	217	XN1	158, 217
Weibull	193	Z1	217
WHILE	245		