МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

О. В. Коломійцев, В. І. Панченко, В. І. Носков, Д. О. Лисиця

ПРОБЛЕМИ І МЕТОДИ МАТЕМАТИЧНОГО І КОМП'ЮТЕРНОГО МОДЕЛЮВАННЯ В НАУКОВИХ ДОСЛІДЖЕННЯХ

Навчальний посібник
для студентів спеціальності F7 «Комп'ютерна інженерія»
денної та заочної форм навчання

Затверджено
редакційно-видавничою
радою НТУ «ХПІ»,
протокол № 1 від 19.02.2026 р.

Харків
НТУ «ХПІ»
2026

УДК 004.94(075) + 519.87(075)

П 78

Рецензенти :

О. О. Можаєв, проф., д-р техн. наук, проф. каф. кібербезпеки та DATA-технологій ННІ № 5, Харківський національний університет внутрішніх справ;

О. В. Сєвєрінов, доц., канд. техн. наук, проф. каф. безпеки інформаційних технологій, Харківський національний університет радіоелектроніки

Проблеми і методи математичного і комп'ютерного моделювання в

П 78 наукових дослідженнях : навчальний посібник для студентів спеціальності F7 «Комп'ютерна інженерія» денної та заочної форм навчання / О. В. Коломійцев, В. І. Панченко, В. І. Носков, Д. О. Лисиця. Харків : НТУ «ХПІ», 2026. 340 с.

ISBN 978-617-05-0615-3

У навчальному посібнику розглянуто теоретичні основи і практичні аспекти моделювання як ефективного методу наукового пізнання складних процесів і систем. Метою посібника є формування в студентів глибоких знань щодо класифікації моделей, етапів моделювання, принципів побудови адекватних математичних і статистичних моделей, а також набуття практичних навичок застосування сучасних програмних засобів імітаційного моделювання. Окремі розділи присвячено питанням планування експериментів, методам дисперсійного, кореляційного та регресійного аналізу результатів імітаційних досліджень. Детально описано використання мови імітаційного моделювання GPSS World.

Призначено для студентів спеціальності F7 «Комп'ютерна інженерія» денної та заочної форм навчання. Також він буде корисним для фахівців з інформаційних технологій, аспірантів та усіх, хто займається теорією та практикою математичного моделювання.

Іл. 60. Табл. 21. Бібліогр. 19 назв.

УДК 004.94(075) + 519.87(075)

© Коломійцев О. В., Панченко В. І.,
Носков В. І., Лисиця Д. О., 2026

ISBN 978-617-05-0615-3

© НТУ «ХПІ», 2026

ЗМІСТ

Вступ 5
1. Сутність моделі та методу моделювання як наукового процесу пізнання 8
1.1. Класифікація моделей і моделювання 8
1.2. Етапи моделювання 26
1.3. Вимоги, що висуваються до моделей 38
Контрольні запитання 49
2. Типові математичні моделі 50
2.1. Дискретні марковські процеси. Схема загибелі та розмноження 50
2.2. Елементи систем масового обслуговування. Коротка характеристика 64
2.3. Сутність і зміст методу динаміки середніх 129
Контрольні запитання 137
3. Статистичне моделювання 139
3.1. Сутність імітаційного моделювання 139
3.2. Загальна характеристика методу імітаційного моделювання 144
3.3. Класифікація випадкових процесів 153
3.4. Способи просування модельного часу 158
Контрольні запитання 168
4. Планування експериментів 169
4.1. Сутність і цілі планування експериментів 169
4.2. Елементи стратегічного планування експериментів 178
4.3. Елементи тактичного планування експериментів 182
4.4. Основні плани, що використовуються в імітаційному експерименті 189
Контрольні запитання 194
5. Обробка результатів імітаційного експерименту 195
5.1. Характеристики випадкових величин і процесів 195

5.2. Елементи дисперсійного аналізу 211
5.3. Сутність кореляційного аналізу 233
5.4. Регресійний аналіз 242
Контрольні запитання 248
6. Моделювання в GPSS World 250
6.1. Основи побудови та принципи функціонування мови імітаційного моделювання GPSS 254
6.2. Методи збору статистики в імітаційній моделі 279
6.3. Блок-діаграми GPSS 285
Контрольні запитання 297
7. Організація комп'ютерних експериментів 298
7.1. Врахування перехідного режиму роботи моделі під час комп'ютерного моделювання 306
7.2. Проведення дисперсійного аналізу (відсіювальний експеримент) у GPSS World 313
7.3. Проведення регресійного аналізу (оптимізаційний експеримент) у GPSS World 318
7.4. Проведення експериментів користувача в GPSS World 321
Контрольні запитання 323
8. Розробка та моделювання імітаційних моделей у системі GPSS World ... 325
8.1. Застосування текстових об'єктів і потоків даних 325
8.2. Розробка моделі в GPSS World 330
Контрольні запитання 337
Список літератури 338

ВСТУП

Сучасний розвиток науки і техніки неможливо уявити без широкого використання методів моделювання. Зростання складності технічних, економічних, соціальних і природних систем призвело до того, що безпосереднє дослідження об’єктів реального світу часто виявляється або надто затратним, або взагалі неможливим. У цих умовах саме моделювання стає універсальним науковим інструментом, який дозволяє відтворити суттєві властивості системи, дослідити її поведінку й отримати знання, необхідні для прийняття рішень.

Фахівець повинен володіти не лише базовими знаннями про предметну галузь, але й аналітичним мисленням, умінням формалізувати реальні процеси, будувати адекватні моделі та на їх основі прогнозувати можливі сценарії розвитку подій. У цьому сенсі моделювання виступає як міст між теорією та практикою, а також як основа для проведення наукових досліджень, створення інноваційних рішень і розробки нових технологій.

Навчальний посібник має на меті надати студентам системні знання з теорії та практики моделювання, навчити їх коректному застосуванню математичного і статистичного апарату для аналізу складних процесів, а також опанувати сучасні інструменти імітаційного моделювання.

У першому розділі розкрито сутність моделі та методу моделювання як наукового процесу пізнання. Тут подано класифікацію моделей та методів

моделювання, охарактеризовано етапи цього процесу, а також окреслено вимоги, які висуваються до моделей з точки зору їхньої адекватності, повноти та можливості практичного застосування. Саме розуміння базових понять дозволяє студентам у подальшому усвідомлено будувати власні дослідження.

Другий розділ присвячено типовим математичним моделям, зокрема дискретним марковським процесам, схемі загибелі та розмноження, а також елементам теорії масового обслуговування. Також наведено ознайомлення із методом динаміки середніх, що дозволяє аналітично досліджувати поведінку систем, для яких характерні стохастичні властивості.

Третій розділ зосереджений на статистичному моделюванні, яке є основою імітаційних досліджень. Тут розглядаються ключові поняття випадкових процесів, способи їх класифікації та методи організації модельного часу. Значну увагу приділено характеристиці імітаційного моделювання як особливого підходу, що дозволяє відтворювати складні системи з урахуванням невизначеності та випадковості.

У четвертому розділі висвітлено питання планування експериментів. Студенти отримають уявлення про стратегічне та тактичне планування, а також про основні схеми експериментів, які застосовуються в імітаційному моделюванні. Це дозволяє раціонально організовувати дослідження, мінімізувати затрати часу й ресурсів і водночас отримувати достовірні результати.

П’ятий розділ присвячено обробці результатів імітаційних експериментів. Окремо розглядаються статистичні характеристики випадкових величин і процесів, методи дисперсійного, кореляційного та регресійного аналізу. Таким чином, студенти отримують інструменти для коректної інтерпретації результатів моделювання, перевірки гіпотез і формування обґрунтованих висновків.

Значне місце в посібнику займає розділ, присвячений мові імітаційного моделювання GPSS (General Purpose Simulation System). Володіння цим

середовищем надає студентам практичні навички створення моделей, організації збору статистики, використання блок-діаграм, а також проведення різноманітних експериментів. GPSS World є одним із найпоширеніших інструментів у сфері імітаційного моделювання і широко застосовується як у наукових дослідженнях, так і в індустрії.

У сьомому розділі увагу зосереджено на організації комп'ютерних експериментів, зокрема на врахуванні перехідних режимів, проведенні дисперсійного та регресійного аналізу безпосередньо в середовищі GPSS World. Окремо розглядається можливість реалізації користувацьких експериментів, що сприяє розвитку дослідницької ініціативи студентів.

Останній, восьмий розділ, має суто прикладний характер і містить методичні рекомендації щодо розробки та моделювання імітаційних моделей у GPSS World. Тут крок за кроком показано, як від абстрактної постановки задачі перейти до програмної реалізації, які інструменти використовувати та як організувати аналіз результатів.

Підхід до розгляду матеріалу, виконаний у даному посібнику, дозволяє студентам не лише засвоїти теоретичні основи моделювання, але й отримати практичні навички, необхідні для проведення наукових досліджень та вирішення реальних завдань у професійній діяльності.

Зміст посібника відповідає структурі підготовки магістрів у галузі комп'ютерної інженерії та суміжних спеціальностей. Кожний розділ супроводжується контрольними питаннями, які сприяють самоконтролю та перевірці рівня засвоєння матеріалу, а також допомагають у підготовці до підсумкових оцінювань.

1. СУТНІСТЬ МОДЕЛІ ТА МЕТОДУ МОДЕЛЮВАННЯ ЯК НАУКОВОГО ПРОЦЕСУ ПІЗНАННЯ

1.1. Класифікація моделей і моделювання

Моделювання – складна, трудомістка і відповідальна наукова задача.

Моделювання – це спосіб дослідження будь-яких явищ, процесів бо об’єктів шляхом побудови й аналізу їх моделей. У широкому розумінні моделювання є однією з основних категорій теорії пізнання і мало не єдиним науково обґрунтованим методом наукових досліджень систем і процесів будь-якої природи в багатьох сферах людської діяльності. Людина постійно створює й використовує моделі для відтворення різноманітних життєвих і виробничих ситуацій з метою пошуку оптимальних рішень. На сьогоднішній день моделюванню приділяється значна увага. За допомогою моделей можливо досліджувати ситуації, які в реальності могли б призвести до виходу системи з ладу. Це відкриває можливості для вивчення катастроф, нетипових або рідкісних подій, а також явищ і процесів, які існують лише у віртуальному просторі.

Моделювання – це метод дослідження явищ і процесів, що ґрунтується на заміні конкретного об’єкта досліджень (оригіналу) іншим, подібним до нього (моделлю) [9, 10, 15]. Моделювання – це процес створення та дослідження моделі, а сама модель – засіб, форма наукового пізнання.

У теорії та практиці моделювання ключовими поняттями є система та модель.

Система – це цілісна сукупність взаємопов’язаних елементів, яка має певну структуру та взаємодіє з оточенням.

Структура системи – це впорядкована система зв’язків між її компонентами, де під зв’язком розуміється можливість впливу одного елемента на інший.

Будь-яка система може знаходитись у певному стані.

Стан системи – це внутрішня характеристика, яка в певний момент часу визначає її поточну поведінку або вихідну реакцію. Він відображає набір параметрів, що формують функціонування системи та її відповідь на зовнішні впливи. Стан можна вважати конкретним набором умов або властивостей, які можна зафіксувати чи відтворити.

Вхід системи (або її елемента) – це точка прикладання зовнішнього впливу, який називають стимулом, подразником або вхідним сигналом. Ці сигнали надходять від зовнішніх по відношенню до системи джерел і змінюють пов’язані з ними змінні. Такі вхідні сигнали ще називають факторними параметрами.

Вхідні параметри бувають трьох типів:

- керовані і реєстровані дослідником;
- некеровані, але реєстровані;
- некеровані і нереєстровані (тобто збурення).

Вихід системи – це точка, де фіксується результат її функціонування. Вихідна реакція (вихідний сигнал) відображає відповідь системи на вхідний вплив. Вихідні параметри характеризують дію системи на середовище і залежать від вхідних факторів. Їх також називають результативними показниками.

Окрім вхідних і вихідних параметрів, виділяють ще параметри стану, які описують зміну внутрішнього стану системи з часом, тобто її динаміку.

Для аналізу динаміки системи зазвичай розглядають три ключові стани:

- нульовий стан – початковий, вихідний;
- стан, що встановився – стан після завершення перехідних процесів;
- стан рівноваги – коли всі зміни припиняються і система стає стабільною.

Зміна стану реального об’єкта завжди супроводжується передачею або перетворенням речовини, енергії чи інформації, тому вона не може бути

миттєвою. Процес зміни стану називається перехідним процесом, а сам момент зміни – трансформацією стану.

Залежно від співвідношення між вхідними та вихідними величинами системи, можливі різні варіанти їхньої взаємодії. Найбільш поширені типові схеми взаємозв'язку між входами і виходами наведено на рис. 1.1.

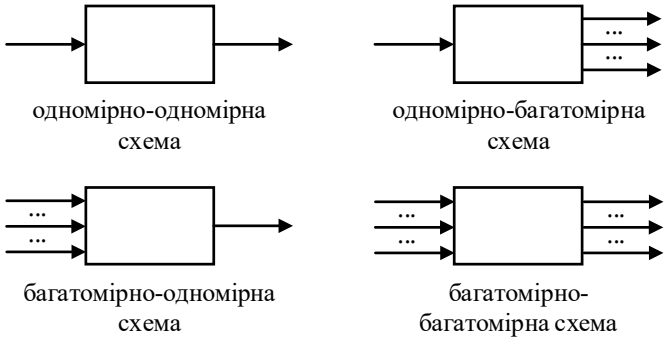


Рисунок 1.1 – Схеми взаємозв'язку між входами і виходами систем

Компоненти та елементи системи мають різні типи взаємозв'язку через свої входи та виходи. Залежно від характеру цих взаємодій, розрізняють незамкнені, замкнені та складні типи зв'язків.

Незамкнені зв'язки – зв'язки, в яких вихід одного елемента впливає лише на вхід іншого, не створюючи зворотного впливу (рис. 1.2).

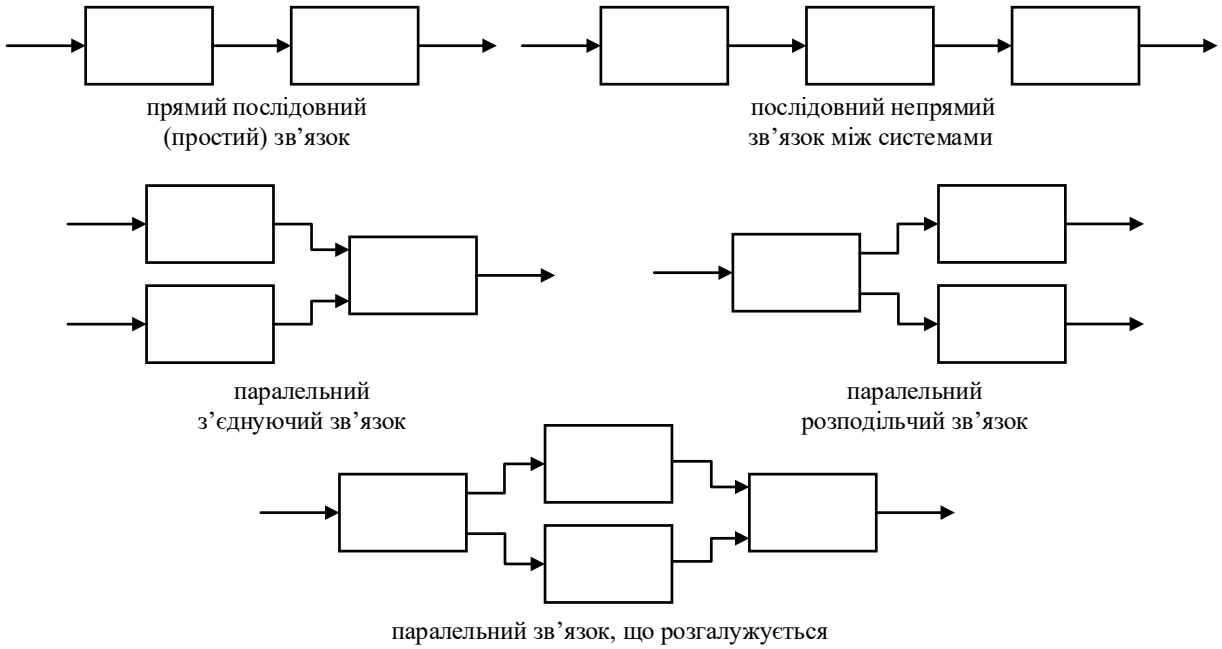


Рисунок 1.2 – Схеми незамкнених зв'язків у системах

Замкнені зв’язки (зворотні) – ситуація, коли вихід елемента (або всієї системи) повертається на її вхід, формуючи контур зворотного зв’язку (рис 1.3).

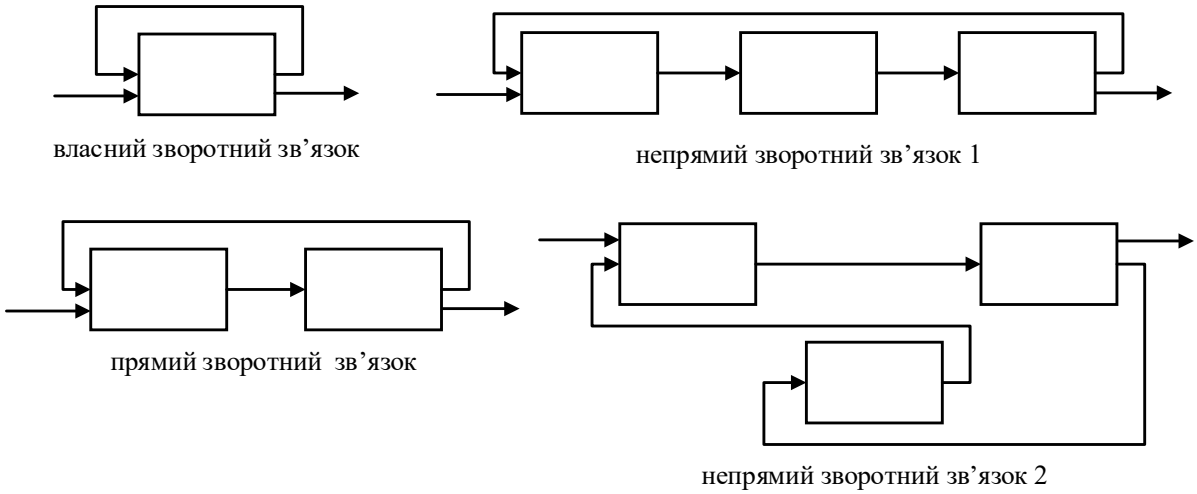


Рисунок 1.3 – Схеми замкнених зв’язків у системах

Складні зв’язки – комбінація прямих і зворотних зв’язків, часто з багатьма взаємодіючими елементами, де кожен компонент може впливати на кілька інших і водночас отримувати зворотні сигнали (рис. 1.4).

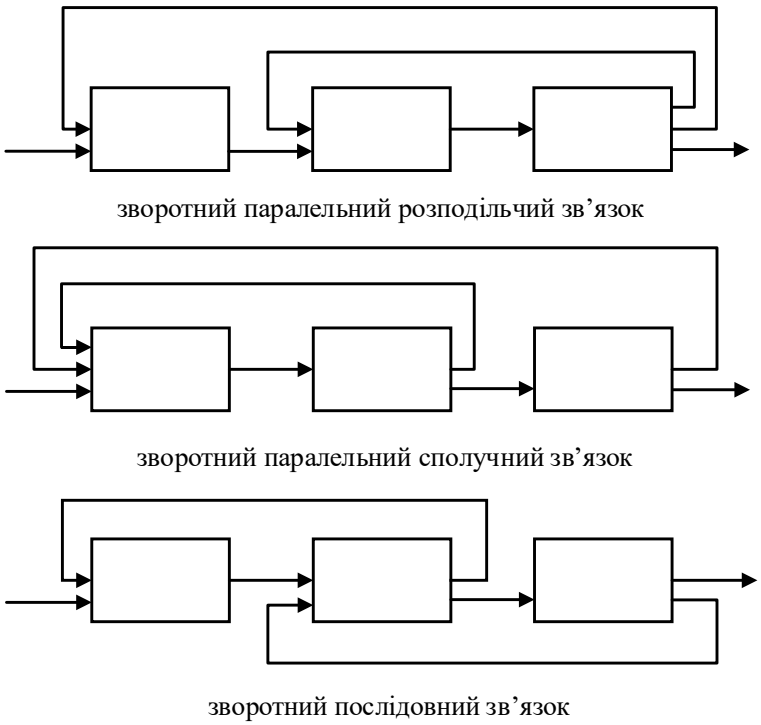


Рисунок 1.4 – Схеми складних зв’язків у системах

Середовище – це набір зовнішніх чинників, які самі не належать до системи, але можуть впливати на її стан або поведінку.

Системи поділяють на відкриті, які взаємодіють із зовнішнім середовищем, і закриті, які не враховують такого впливу, що обумовлено метою дослідження.

Система – це сукупність взаємодіючих елементів, об’єднаних для досягнення певної мети, що особливо актуально для технічних та соціальних систем.

У межах абстрактної теорії систем виділяють кілька рівнів опису систем:

- лінгвістичний (символічний);
- теоретико-множинний;
- алгебричний (абстрактний);
- топологічний;
- логіко-математичний;
- інформаційний (теоретико-інформаційний);
- динамічний;
- евристичний.

Найзагальнішим з них є лінгвістичний рівень, що дозволяє побудувати інші рівні абстракції. Тут вводиться поняття предметної області, яку описують за допомогою алгебричних моделей, прив’язаних до певної мови. Формалізація опису здійснюється через формальні мови, які формують логіко-алгебричну модель, що дозволяє робити висновки в межах логічної системи тверджень.

Вибір відповідного рівня абстракції є складним і відповідальним етапом дослідження системи, оскільки він значною мірою залежить від досвіду, знань дослідника, а також цілей дослідження.

Процес моделювання складається з кількох послідовних етапів:

- формування гіпотези або теорії;

- побудова моделі для її перевірки;
- визначення параметрів моделі;
- перевірка моделі та аналіз статистичних результатів;
- прогнозування на основі моделі;
- практичне використання моделі (наприклад, для управління).

Будь-яке моделювання передбачає наявність трьох ключових компонентів:

- оригіналу (об’єкта дослідження);
- моделі (як відображення об’єкта);
- дослідника (того, хто здійснює моделювання).

Під моделлю розуміється об’єкт будь-якої природи (уявлена або матеріально реалізована система), котрий, відображаючи чи відтворюючи в певному сенсі об’єкт дослідження, здатний заміщати його так, що вивчення моделі дає нову інформацію про об’єкт.

Методом моделювання називається такий загальнонауковий метод дослідження, при якому вивчається не сам об’єкт пізнання, а його зображення у вигляді моделі, однак результат дослідження переноситься з моделі на об’єкт.

Основні поняття, що використовуються при моделюванні систем: аналогія, абстракція і спрощення. Науковим підґрунтям моделювання як способу пізнання та дослідження об’єктів і процесів є теорія подібності, в основі якої лежить поняття аналогії – тобто наявність спільних ознак між різними об’єктами. Об’єкти, що мають подібність, називаються аналогами. Аналогія може встановлюватися як на якісному, так і на кількісному рівнях.

Найважливішим прикладом кількісної аналогії є математична подібність, коли об’єкти описуються рівняннями або функціями. Функції та змінні вважаються подібними, якщо вони збігаються з точністю до певних коефіцієнтів. Окремими випадками такої подібності є геометрична подібність (подібність форм), та часова подібність (подібність функцій часу), де

масштаб часу вказує на співвідношення параметрів, таких як період чи затримка. Ще одним різновидом кількісної аналогії є фізична подібність, яку можна встановити без точного математичного опису, лише на основі зіставлення фізичних характеристик процесів у реальному об’єкті та моделі. Залежно від характеру процесів виділяють типи подібності: гідравлічну, електричну, аеродинамічну тощо.

Завдяки теорії подібності можна визначити еквівалентність або відповідність між двома системами – як реальними, так і абстрактними – за певними характеристиками.

Модель – це прообраз, опис або зображення якогось об’єкта [9]. Моделі зазвичай бувають у вигляді малюнків, креслень, схем, таблиць, матриць, символів або описуються у вигляді тексту. До моделі включають важливі з точки зору пізнання риси і виключають несуттєві.

Будь-яка модель нетотожна об’єкту-оригіналу, оскільки за її побудови дослідник враховував лише найважливіші з погляду чинники. У цьому плані будь-яка модель є неповною. Повна модель, вочевидь, буде повністю тотожною оригіналу.

До побудови моделі вдаються тоді, коли використання об’єкта-оригіналу з будь-яких причин утруднене або неможливе, наприклад:

- об’єкт має або надто великі (сонячна система) або надто малі розміри (молекули);
- процес, який моделюється, має або надто велику швидкість (згоряння палива в двигуні внутрішнього згоряння) або занадто повільний (процес еволюції живих істот);
- дослідження об’єкта може бути небезпечним для оточуючих (процес атомного вибуху);
- об’єкт-оригінал може бути зруйнований під час дослідження (дослідження характеристик міцності конструкції будівлі);

- об’єкт, який досліджується, не існує (проєктування нового пристрою, який ще не має фізичної реалізації) тощо.

Кожна модель створюється для конкретної мети і, отже, унікальна. Математичні моделі є важливим засобом, за допомогою якого людина пізнає явища навколишнього світу. Вони відображають основні закономірності та взаємозв’язки, характерні для об’єкта дослідження, і можуть бути представлені у вигляді формул, рівнянь, систем правил або умов, виражених математично.

Модель подібна до оригіналу тільки в обмеженій кількості відносин.

На рис. 1.5 наведено спрощену схему моделювання.

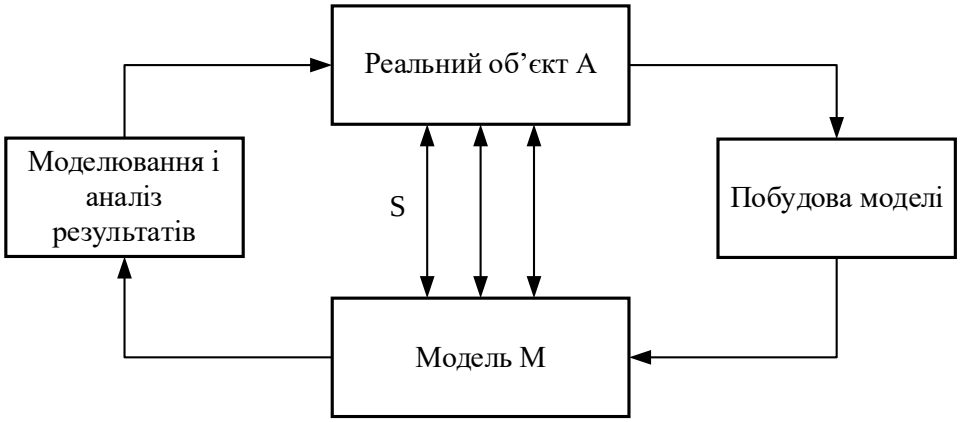


Рисунок 1.5 – Спрощена схема моделювання

Важливо відмітити, що модель M є моделлю реального об’єкта A лише відносно деякої системи S її характеристик (властивостей). При цьому модель M обирається або будується саме для імітації цих характеристик. Оскільки модель M будується лише імітації частини властивостей реального об’єкта A, зазвичай вона виявляється простіше його.

Методи комп’ютерного моделювання знаходять широке застосування в різноманітних сферах – від створення моделей технічних, технологічних та організаційних систем до аналізу глобальних проблем людства та довкілля. Серед класичних об’єктів моделювання – інформаційні, виробничі, транспортні й інші логістичні системи, моделі яких здебільшого використовуються для проєктування, модернізації та стратегічного

планування. Комп'ютерне моделювання є незамінним інструментом у процесі прийняття рішень, прогнозування наслідків певних дій.

Наявність спільних рис моделей дає змогу згрупувати все їхнє різноманіття в окремі класи, що полегшує їх розроблення та вивчення. У теорії розглядають багато ознак класифікації та їхня кількість не встановилася [9, 11, 10]. Найбільш актуальні ознаки класифікації моделей:

- за характером модельованої сторони об'єкта, що моделюється:
  - функціональні (кібернетичні);
  - геометричні;
  - структурні;
  - інформаційні;
- за характером процесів, що протікають в об'єкті:
  - детерміновані або стохастичні;
  - статичні або динамічні;
  - дискретні, безперервні або дискретно-безперервні;
- за способом реалізації моделі:
  - абстрактні (уявні) моделі:
    - символьні;
    - математичні:
      - аналітичні;
      - імітаційні;
      - змішані (аналітико-імітаційні);
  - матеріальні моделі:
    - макетні (масштабні);
    - аналогові;
    - натурні.

Функціональна модель (functional model) відображає сукупність виконуваних системою функцій. У цьому випадку модельований об'єкт розглядається як «чорний ящик», який має входи і виходи. Фізична сутність

об’єкта, природа процесів, що протікають, структура об’єкта залишаються поза увагою дослідника.

Функціональне моделювання полягає в спостереженні за виходом модельованого об’єкта при зміні вхідних впливів. За цими даними будується модель поведінки у вигляді деякої математичної функції.

Іншими словами, функціональна (або поведінкова) модель описує динаміку функціонування: стан системи, події, перехід з одного стану в інший, умови переходу, послідовність подій.

Геометрична модель (geometric model) задається описами об’єктів (наприклад, математичними формулами), які задають їх геометричні форми. Але вона містить не тільки різні геометричні об’єкти, але й встановлює зв’язок між ними та властивостями матеріалів, граничними умовами тощо. Найчастіше використовується математичний опис геометричних фігур. Геометричні моделі мають графічне представлення. Графічні побудови геометричних моделей використовуються для вирішення тих чи інших математичних задач.

Геометричне моделювання – моделювання об’єктів за допомогою геометричних типів даних.

Структурна модель (structural model) подає об’єкт моделювання з точки зору його складу та взаємозв’язку частин (елементів системи) між собою та з зовнішнім середовищем. Структурна модель може бути описана на різних рівнях розгляду. Найбільш загальний опис структури – це топологічний опис за допомогою теорії графів.

Структурне моделювання – це створення і дослідження моделі, структура якої (елементи, зв’язку) подібна до структури об’єкта, що моделюється.

Інформаційна модель (information model) – модель об’єкта, що представлено у вигляді інформації, яка описує суттєві для даного розгляду параметри і змінні величини про об’єкт, зв’язки між ними, входи і виходи

об’єкта і дозволяє шляхом подачі на модель інформації про зміни вхідних величин моделювати можливі стани об’єкта.

Детермінована модель (deterministic model) відображає процеси, в яких відсутній випадковий вплив. Така модель має стани, що точно визначені в будь-який момент часу, і, відповідно, можуть бути описані математичними формулами і заздалегідь визначеними параметрами. У таких моделях не враховуються можливі випадкові відхилення характеристик об’єктів і вхідних впливів від номінальних значень.

Стохастична модель (stochastic model) відображає імовірнісні процеси і події. Значення деяких параметрів такої моделі неможливо передбачити. Стохастичні моделі часто використовуються в економіці та фінансах, оскільки вони можуть прогнозувати різні варіанти розвитку ситуації залежно від випадкових факторів, таких як зміни ринкової кон’юнктури, політичні ризики та інші фактори, які неможливо передбачити точно.

Статична модель (static model) служить для опису стану об’єкта в будь-який момент часу. У такій моделі всі залежності віднесені до одного моменту часу. Такими моделями можуть описуватися статичні системи, але також і динамічні системи: шляхом характеристики їх стану в заданий момент. При моделюванні процесів у таких моделях зміна вхідних параметрів призводить до миттєвої одномоментної зміни залежних параметрів без будь-яких запізнювань у часі.

Динамічна модель (dynamic model) знаходиться у відношенні динамічної подоби до об’єкта, що моделюється, тобто відображає поведінку об’єкта в часі. Динамічна модель може містити опис етапів або фаз або діаграму станів підсистем та часто має математичний опис.

Дискретна модель (discreet model) відображає поведінку системи з дискретними станами. Змінні та параметри в таких моделях є дискретними величинами, тобто величинами, що приймають кінцеве чи лічильне число значень. Безліч допустимих рішень при використанні дискретних моделей також є дискретним.

Безперервна модель (continuous model) представляє систему з безперервними процесами. У такій моделі змінні можуть набувати будь-яких, а не тільки дискретних, значень у заданому діапазоні.

Таким чином, дискретні моделі використовуються для систем, поведінка яких змінюється лише в задані моменти часу, а безперервні моделі – для систем, поведінка яких змінюється безперервно в часі.

Дискретно-безперервні моделі будуються тоді, коли дослідника цікавлять обидва ці типи процесів. За допомогою таких моделей досліджуються складні паралельні та розподілені динамічні системи, що складаються з великої кількості елементів різної природи, а саме – з елементів, поведінка яких описується безперервними процесами, що мають кінцеву тривалість, та елементів, поведінка яких описується дискретними процесами, час реакції на події в яких несуттєвий для аналізу системи. Дискретно-безперервні моделі поєднують переваги дискретних та безперервних моделей та дозволяють більш точно та детально відтворювати поведінку складних процесів та явищ.

Для опису процесів, які змінюються в часі, використовується відповідно **дискретне**, **неперервне** та **дискретно-неперервне моделювання**.

Абстрактна або наукова **модель** (scientific model) є певною конструкцією із загальноприйнятих знаків на матеріальному носії або у вигляді комп'ютерної програми. Така модель представлена лише в розумі людини, вона нематеріальна.

Символьна (символічна) **модель** (symbolic model) – це логічний об'єкт, що заміщає реальний процес і виражає основні властивості його відносин за допомогою певної системи символів або символів. Ця модель може мати самостійне значення, але, зазвичай, її побудова є початковим етапом будь-якого іншого моделювання.

Символьне моделювання (s-моделювання) – побудова моделей довільних об'єктів у людино-машинному середовищі на основі систем візуальних, аудіо- та інших символів. S-моделювання вивчає системи

символів та відповідні їм системи кодів; методи побудови, збереження, накопичення, пошуку та передачі найчастіше за допомогою обчислювальної техніки.

Математична модель (mathematical model) є наближеним описом об’єкта (явища, процесу), вираженим за допомогою математичної символіки.

Математичні моделі поділяють на класи залежно від:

- складності об’єкта моделювання (простий або складний);
- оператора моделі (модель є оператором, що визначається сукупністю рівнянь: алгебричних, систем звичайних диференціальних рівнянь тощо);
- вхідних і вихідних параметрів;
- способу дослідження моделі;
- цілей моделювання (дескриптивні або оптимізаційні).

Метою дескриптивних моделей є встановлення законів зміни параметрів моделі.

Оптимізаційні моделі призначені для визначення оптимальних з точки зору деякого критерія параметрів об’єкта, що моделюється, або для пошуку оптимального режиму управління процесом.

Якщо задано структурну або функціональну модель (рис. 1.6), то можна перейти до математичного її вигляду: $Y = M(X)$.

Рисунок 1.6 – Структурне зображення моделі

Модель – це закономірність, яка перетворює вхідні значення у вихідні. Як відомо з математики, з виразом $Y = M(X)$ можна розв’язати три види задач, що наведені в табл. 1.1.

Таблиця 1.1 – Форми запису моделі та типи розв'язуваних задач

	Відомі параметри	Невідомі параметри	Розв'язання
Пряма задача	X, M	Y	$Y = M(X)$
Зворотна задача	Y, M	X	$X = M^{-1}(Y)$
Задача налаштування	X, Y	M	$M = f(X, Y)$

Ряд моделей може бути невизначеним – це означає, що варіантів рішень багато. Якщо потрібне одне рішення, то проблему треба визначити, доповнити умовами. Якщо при побудові моделі щось не було враховано, то модель треба добудувати. Серед множини рішень також можна обирати найкраще рішення шляхом звуження області рішень, накладаючи певні обмеження для відсіву. Такі задачі часто називають задачами керування.

Частина визначень, яким треба задовольнити, називаються обмеженнями. Частина визначень, щодо яких висловлюють лише побажання, називаються критеріями.

Зворотна задача – те, що треба визначити, є керованою змінною. Тобто досліджується, як слід змінити вхідний параметр (управління), щоб забезпечити виконання законів, не вийти за обмеження і критерії набули найкращих значень.

Математичне моделювання – це процес встановлення відповідності модельованого об'єкта деякій математичній конструкції (математичній моделі), і дослідження цієї моделі, що дозволяє отримати характеристики об'єкта моделювання. Математичне моделювання – це ідеальне наукове знакове формальне моделювання, при якому опис об'єкта здійснюється мовою математики, а дослідження моделі проводиться з використанням тих чи інших математичних методів. На даний час математичне моделювання – це один з найрезультативніших і найчастіше застосовуваних методів наукового дослідження.

Аналітична модель (analytical model) – це математичне подання реального процесу або системи, що використовується для аналізу даних і прогнозування. Аналітичні моделі будуються за допомогою функціональних

співвідношень: систем алгебричних, диференціальних, інтегро-диференціальних рівнянь, статистичних методів, логічних умов. Основна мета аналітичної моделі – спростити реальність, зберігши при цьому основні характеристики, що впливають на результати, що дозволяє приймати обґрунтовані рішення на основі емпіричних даних. Аналітична модель, зазвичай, статична. Аналітичне уявлення підходить лише для дуже простих і сильно ідеалізованих завдань і об'єктів, які, як правило, мають мало спільного з реальною (складною) дійсністю, але мають високу спільність. Даний тип моделей зазвичай застосовують для опису фундаментальних властивостей об'єктів, оскільки фундамент простий за своєю суттю. Аналітичні моделі часто складні для формалізації та побудови, а іноді їх взагалі неможливо побудувати.

Аналітичне моделювання дозволяє отримати характеристики системи як деякі функції її параметрів функціонування. Аналітичні методи дають точні значення характеристик системи, але застосовуються для вирішення лише вузького класу завдань. Основні причини обмежень застосування – це те, що для більшості реальних систем їх закінчений математичний опис (модель) відсутній, або ще не розроблено аналітичні методи рішення створеної математичної моделі. Також, при виведенні формул, на яких ґрунтуються аналітичні методи, приймаються певні припущення, які не завжди відповідають реальній системі. У таких випадках від аналітичних методів моделювання доводиться відмовлятися.

Імітаційна модель (simulation model) – логіко-математичний опис об'єкта, який може бути використаний для експериментування на комп'ютері з метою проектування, аналізу та оцінки функціонування об'єкта. Основна відмінність імітаційних моделей від аналітичних полягає в тому, що замість аналітичного опису взаємозв'язків між входами та виходами досліджуваної системи будують алгоритм, що відображає послідовність розвитку процесів усередині об'єкта, що досліджується, а потім моделюють поведінку об'єкта засобами обчислювальної техніки. До імітаційних моделей вдаються тоді,

коли об’єкт моделювання настільки складний, що адекватно описати його поведінку математичними рівняннями неможливо чи важко.

Імітаційне моделювання – це метод, що дозволяє будувати моделі, що описують процеси так, як вони проходили б насправді [19]. Це окремий випадок математичного моделювання. Існує клас об’єктів, для яких з різних причин не розроблені аналітичні моделі або не розроблені методи розв’язування задач про такі моделі. У цьому випадку математична модель замінюється імітаційною моделлю.

Імітаційні моделі досить просто враховують вплив випадкових факторів. Для аналітичних моделей це серйозна проблема. За наявності випадкових факторів необхідні характеристики процесів, що моделюються, отримують багаторазовими прогонами (реалізаціями) імітаційної моделі та подальшою статистичною обробкою накопиченої інформації.

Статистичне моделювання – метод дослідження складних систем, заснований на описі процесів функціонування окремих елементів у взаємозв’язку з метою отримання безлічі результатів, що далі підлягають обробці методами математичної статистики для отримання кінцевих результатів.

Якщо дослідження об’єкта ускладнене використанням лише аналітичного чи імітаційного моделювання, то застосовують змішане (комбіноване), **аналітико-імітаційне моделювання**. При побудові **аналітико-імітаційних моделей** процеси функціонування об’єкта декомпозують на підпроцеси, для частини з яких використовують аналітичні моделі, а для решти – імітаційні моделі.

Комп’ютерна модель – це модель (математична чи інформаційна), що реалізована за допомогою програмних засобів. Для обробки комп’ютерних моделей використовуються існуючі програмні застосунки (математичні пакети, електронні таблиці, графічні редактори тощо) або розробляються оригінальні програми за допомогою як універсальних так і спеціалізованих мов програмування (C++, Python, Java, Modelica, Simulink тощо).

Матеріальна (фізична) **модель** (physical model) – це предметне подання об’єкта, процесу або явища, що реально існує, зі збереженням його геометричних і фізичних властивостей. Це така модель, в якій хоча б один компонент є фізичною копією реального об’єкта. На відміну від абстрактних, матеріальні моделі існують у природі, і з ними можна експериментувати.

Нерідко створюються **матеріально-абстрактні моделі**. Та частина операції, яка не піддається математичному опису, моделюється матеріально, інші – абстрактно.

Матеріальне моделювання засноване на застосуванні моделей, які представляють собою реальні технічні конструкції. Це може бути сам об’єкт або його елементи (**натурне моделювання**); спеціальний пристрій – модель, що має фізичну або геометричну подібність оригіналу; пристрій іншої фізичної природи, ніж оригінал, але процеси в якому описуються аналогічними математичними співвідношеннями.

Натурна модель (full-scale model) – це реальна досліджувана система, яка має повну адекватність з системою-оригіналом, що забезпечує високу точність і достовірність результатів моделювання; іншими словами, це – матеріальна копія об’єкта моделювання, це об’єкт, який створений самою природою. Істотний недолік натурної моделі – це неможливість моделювання критичних і аварійних режимів її роботи і висока вартість.

Масштабна (макетна) **модель** (scale model) – це система тієї ж фізичної природи, що і оригінал, але яка відрізняється від нього розмірами. В основі масштабних моделей лежить математичний апарат теорії подібності, який передбачає дотримання геометричної подоби оригіналу і моделі і відповідних масштабів для їх параметрів: параметри моделі і системи відрізняються між собою. Числове значення цієї відмінності називається масштабом моделювання, або коефіцієнтом схожості. Прості масштабні моделі – це пропорційно зменшені копії існуючих систем, які відтворюють основні властивості системи або об’єкта залежно від мети моделювання.

Масштабне моделювання полягає в побудові макету об’єкта, що вивчається, і визначенні на основі його аналізу тих чи інших корисних (прийнятних) властивостей оригіналу. Макет і оригінал можуть відрізнятися як геометричними розмірами, так і властивостями матеріалу, з якого вони виготовлені. Масштабне моделювання часто використовується при проектуванні великих об’єктів у літакобудуванні і суднобудуванні.

Аналогова модель (analog model) являє аналог досліджуваного об’єкта, який веде себе як реальний об’єкт, але не виглядає як такий.

Аналогове моделювання ґрунтується на аналогії процесів і явищ, що мають різну фізичну природу, але однаково описуваних формально (одними і тими ж математичними рівняннями, логічними схемами тощо). Має місце універсальна властивість природи: процеси різної фізичної природи описуються одними й тими ж математичними рівняннями. В якості аналогових моделей використовуються механічні, гідравлічні, пневматичні системи, але найбільш широке застосування отримали електричні та електронні аналогові моделі, в яких сила струму або напруга є аналогами фізичних величин, що мають іншу природу.

Якщо порівнювати **матеріальне** та **математичне моделювання**, то можна відзначити такі переваги математичного моделювання:

1) економічність (зокрема, заощадження ресурсів реальної системи);

2) можливість моделювання гіпотетичних, тобто не реалізованих у природі об’єктів (насамперед на різних етапах проектування);

3) можливість реалізації небезпечних чи важковідтворюваних режимів у природі (критичний режим ядерного реактора, робота системи протиракетної оборони);

4) можливість зміни масштабів часу;

5) простота багатоаспектного аналізу;

6) велика прогностична сила внаслідок можливості виявлення загальних закономірностей;

7) універсальність технічного та програмного забезпечення роботи (обчислювальна техніка, системи програмування та пакети прикладних програм широкого призначення).

1.2. Етапи моделювання

Моделювання – це творчий процес. Укласти його у формальні рамки дуже важко. У найбільш загальному вигляді поетапне виконання моделювання наведено на рис. 1.7.

Результати проведення кожного етапу моделювання необхідно документувати. Моделювання – це процес ітеративний, з кожного етапу може бути виконане повернення на будь-який із попередніх етапів для уточнення та модифікації інформації, і наявність документації може зберегти результати попередніх обчислень.

Також, при дослідженні складних систем, коли задіяні різні групи дослідників для різних етапів, результати, які отримані на кожному етапі, необхідно передавати між групами. Крім того, результати, отримані на кожному з етапів, можуть бути самоцінними продуктами, які, наприклад, можна використовувати в інших дослідженнях.

Опис основних етапів моделювання.

Етап постановки задачі. Взагалі-то це головний етап будь-якої діяльності. На цьому етапі виконується опис задачі, визначення мети моделювання, аналіз об’єкта.

Опис задачі виконується звичайною мовою. Аналіз об’єкта має на увазі чітке виділення модельованого об’єкта та його основних властивостей.

При визначенні мети моделювання необхідно серед багатьох характеристик (параметрів) об’єкта виділити суттєві. Зазвичай цілями моделювання є:

- створення об’єкта з заданими властивостями або прогноз поведінки існуючого об’єкта за нових режимів, сполучень чинників тощо;

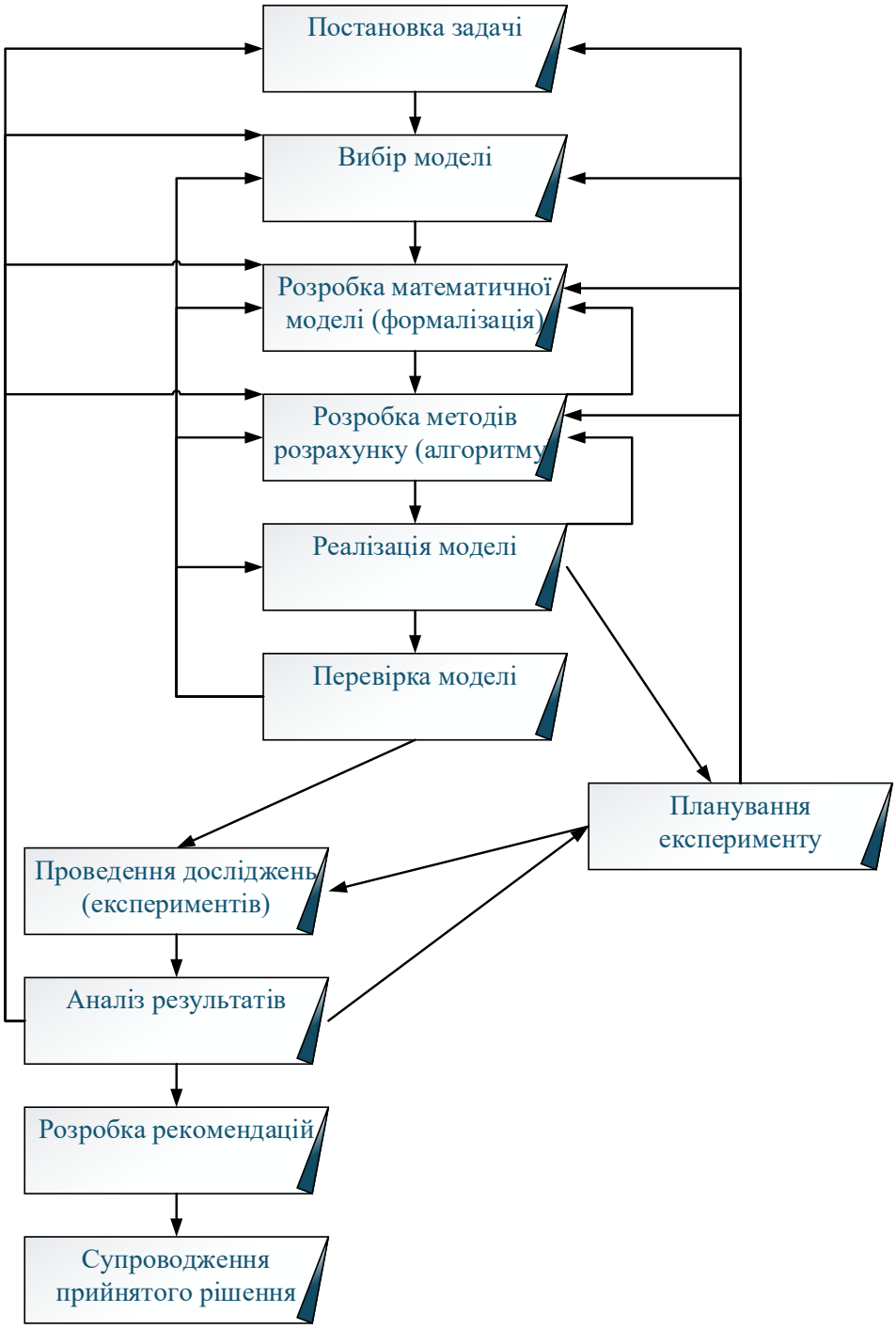


Рисунок 1.7 – Етапи моделювання

- ефективність керування об’єктом (або процесом) – підбір поєднання і значень факторів, що забезпечують оптимальне значення показників ефективності процесу;
- аналіз чутливості системи на зміну тих чи інших чинників;

- перевірка різного роду гіпотез про характеристики випадкових параметрів досліджуваного процесу;
- визначення наслідків впливу на об’єкт та прийняття правильного рішення – визначення функціональних зв’язків між поведінкою («реакцією») системи і факторами, що на неї впливають, що може сприяти прогнозу поведінки або аналізу чутливості;
- пізнання навколишнього світу. З’ясування сутності, кращого розуміння об’єкта дослідження, а також формування перших навичок для експлуатації модельованої або діючої системи.

При постановці задачі вирішуються такі взаємозалежні завдання:

1. З’ясування мети (завдання) дослідження. Будь-яке серйозне дослідження починається з вивчення системи, її цілей, структури, взаємодії з довкіллям, перспектив існування та розвитку. Причому формування завдання замовником, якщо такий є, слід розглядати лише як перше наближення.

2. Вивчення об’єкта моделювання (системи, процесу). Характер вивчення системи вирішальним чином залежить від таких обставин: фізичної природи системи, її цілей та задачі (мети) дослідження. При вивченні системи необхідно зрозуміти її призначення, структуру, межі, взаємодію з довкіллям, перспективи існування та розвитку. Особливу увагу слід приділяти відносинам, реально існуючим між елементами системи, суті існуючих у системі проблем і причин, що викликали ці проблеми.

3. Аналіз доступної інформації. Інформація потрібна не будь-яка, а лише та, яка справді забезпечить вирішення задачі. Типовою помилкою є початок досліджень із збирання інформації, тобто до того, як з’ясовано постановку задачі та намічено методику дослідження: існує велика ймовірність, що буде зібрано багато непотрібної інформації, а важливі дані – будуть відсутні. Важливо також враховувати чинник старіння інформації. Характеристики доступної інформації (повнота, достовірність, точність) впливають структуру моделі, методику проведення експерименту.

4. Виявлення релевантних чинників – факторів, які суттєвим, вирішальним чином впливають на результати дослідження. Після визначення релевантних факторів проводиться вибір тих, які можуть бути описані кількісно, уточнення списку цих факторів шляхом об’єднання їх за загальними ознаками та виключення істотно залежних між собою факторів. Після уточнення списку релевантних факторів потрібно переконатися, чи не призведе відмова від деяких факторів до неприпустимого зниження точності розв’язання задачі.

5. Формулювання альтернатив. На початковому етапі необхідно зберегти всі можливі альтернативи розв’язання задачі У найпростішому випадку замість вибору альтернатив потрібно визначити діапазони змінних та параметрів моделі. Існує кілька організаційних форм генерування альтернатив:

- мозковий штурм – колективний метод генерування ідей, під час якого фахівці висувають альтернативні рішення без критики, з акцентом на розвиток і взаємозв’язок пропозицій;
- синектика – метод генерування ідей через асоціативне мислення, коли для вирішення проблеми використовуються аналогії з інших галузей або явищ;
- розробка сценаріїв – опис можливого розвитку подій за різних альтернатив за однакових стартових умов із урахуванням усіх релевантних чинників впливу;
- морфологічний аналіз – метод, за допомогою якого визначаються всі можливі значення ключових змінних і розглядаються всі їхні допустимі комбінації для пошуку оптимального рішення;
- ділові ігри – моделювання реальних ситуацій у формі людино-машинної системи, де учасники приймають рішення, щоб проаналізувати наслідки різних варіантів розвитку подій.

6. Визначення обмежень та припущень. Відмова від факторів, віднесених до нерелевантних, свідоме спрощення низки залежностей, обмеження області зміни деяких змінних тощо, сприяють спрощенню моделі, здешевленню експерименту. Однак при спрощенні моделі може бути втрачено адекватність моделі.

7. Вибір критеріїв якості досягнення мети. Критерії мають забезпечити найкраще, у певному сенсі, рішення. Проблема вибору критерію – це встановлення ознак, яким віддається перевага. У певному вигляді критерій може бути і не сформульований, але його характер та вид переваги буде визначений. Задача спрощується, якщо вдається обмежитися одним критерієм, але для реальних задач більш властива багатокритеріальність.

8. Встановлення масштабу майбутнього експерименту. Остаточне визначення масштабу експерименту проводиться на етапах планування і проведення експериментів. Однак вже при постановці завдання необхідно оцінити передбачуваний їх масштаб і, у тому числі, обмеження, які можуть виникнути через брак ресурсів і коштів (досить типові для прикладних досліджень обмеження). При розробці моделі враховується і передбачуваний характер використання моделі – чи використовуватиметься модель повторно.

Етап вибору моделі передбачає визначення типу та структури моделі, яка найкраще відповідає цілям дослідження та об’єкту, що моделюється. На цьому етапі розглядаються різні типи моделей: математичні, інформаційні, структурно-функціональні тощо, та їх відповідність поставленій задачі.

Етап формалізації задачі. Виконується створення формалізованої інформаційної моделі, тобто моделі, записаної якоюсь формальною мовою. У такій моделі зв’язок між вихідними даними та кінцевими результатами фіксується за допомогою різних формул, а також накладаються обмеження на допустимі значення параметрів.

На цьому етапі досліджується об’єкт, встановлюються необхідні спрощення та апроксимації. Виявляються суттєві аспекти, виключаються другорядні. Встановлюються одиниці виміру та діапазони варіювання

змінних моделі. Визначається поведінка моделі в крайніх та граничних ситуаціях. Якщо можливо, то модель представляють у вигляді відомих систем: масового обслуговування, управління, авторегулювання, різного роду автоматів тощо.

Формалізація задачі істотно залежить від знання об’єкта, що досліджується, задачі дослідження, а також виду створюваної моделі. Загального методу підбору залежностей (відносин, функцій) немає. Чим більше функціональних залежностей відомо досліднику, чим більше він може залучити і критично осмислити аналогій, тим успішнішою буде його діяльність з розробки моделі.

Результатом є узагальнена схема моделі, повністю підготовлена для математичного опису – побудови математичної моделі.

У практиці моделювання на початкових етапах формалізації об’єктів використовують типові математичні схеми, до яких відносять такі добре опрацьовані (розроблені) математичні об’єкти, як диференціальні алгебричні рівняння, кінцеві ймовірнісні автомати тощо (табл. 1.2).

Таблиця 1.2 – Форми запису моделі та типи розв’язуваних задач

Процес функціонування системи	Типова математична схема	Позначення
безперервно-детермінований підхід	стандартні диференціальні рівняння	D-схема
дискретно-детермінований підхід	скінченні автомати	F-схема
дискретно-стохастичний підхід	ймовірнісні автомати	P-схема
безперервно-стохастичний підхід	система масового обслуговування	Q-схема
узагальнений (універсальний) підхід	агрегативна система	A-схема

При агрегативному підході спочатку дається формальне визначення об’єкта моделювання – агрегативної системи, яка є математичною схемою, що відображає системний характер об’єктів, що досліджуються. При агрегативному описі складний об’єкт (система) розбивається на кінцеве число частин (підсистем), зберігаючи зв’язки, що забезпечують взаємодію підсистем. Якщо деякі з отриманих підсистем виявляються, у свою чергу, ще досить складними, то процес їхнього розбиття триває до того часу, доки

утворюються підсистеми, які в умовах задачі моделювання можуть вважатися зручними для математичного опису. У результаті такої декомпозиції складна система представляється у вигляді багаторівневої конструкції із взаємозалежних елементів, об'єднаних у підсистеми різних рівнів.

Етап розробки математичної моделі.

У порівнянні з лабораторним (натурним) експериментом комп'ютерний експеримент дешевший, безпечніший, може проводитися в тих випадках, коли натурний експеримент принципово неможливий.

Існують такі основні принципи підходу до розробки моделей за різної доступності до інформації щодо структури системи та процесів, що протікають у ній:

1. **Система** досить **проста і прозора**, тому її можна обстежити і зрозуміти, наприклад, шляхом спостереження чи розпитувань людей, які з нею працюють. За результатами вивчення системи можна сконструювати її модель.

2. Якщо **структура системи очевидна**, але **методи опису не зрозумілі**, можна скористатися подібністю досліджуваної системи з іншою, зокрема, можливо, простішою, опис якої відомий.

3. **Структура** системи **невідома**, але її **можна визначити** шляхом аналізу даних про **функціонування** системи. Фактично буде отримано гіпотезу про структуру, яку потім необхідно перевірити експериментально.

4. **Аналіз** даних роботи системи **не дозволяє визначити вплив окремих змінних** на показники роботи системи, виникає необхідність у проведенні експерименту з метою виявлення релевантних факторів та їх впливу на роботу системи. У цьому випадку передбачається можливість проведення відповідного експерименту на системі.

5. Достатні описові **дані про систему відсутні**, **проведення експерименту на системі не припустимо**. Може бути побудована досить докладна модель штучної дійсності, використовувана для накопичення

статистики про можливе функціонування системи шляхом статистичних випробувань гіпотез про реальний світ.

Рекомендується під час створення математичної моделі:

- підшукати аналогії;
- підібрати та розглянути спеціальні приклади, характерні для розв’язуваної задачі;
- прийняти рішення про вибір класу (типу) моделі, у тому числі обрати тип моделі: аналітична, імітаційна, комбінована;
- записати міркування, що характеризують закономірності, які є в системі, за необхідності провести додаткові дослідження;
- якщо модель не піддається опису, знайти способи спрощення проблеми.

Результатом етапу розробки математичної моделі є програма, складена найзручнішою для моделювання та дослідження мовою – універсальною або спеціальною.

Етап планування експерименту. Математична модель є об’єктом експерименту. Експеримент повинен бути в максимально можливій мірі інформативним, задовольняти обмеженням, забезпечувати отримання даних з необхідною точністю і достовірністю.

При плануванні експерименту ставиться задача забезпечити досягнення мети дослідження за мінімальних витрат ресурсів всіх видів. Труднощі отримання необхідної достовірності результатів пов’язані з наявністю різноманітних перешкод, у тому числі – помилок вимірювання вхідної інформації.

При плануванні обчислювального експерименту необхідно, як мінімум, визначити область існування параметрів та змінних, оцінити хоча б якісно чи грубо кількісно вплив зміни всіх параметрів та змінних на результати моделі та обрані критерії якості розв’язання задачі, підібрати приклади для аналізу залежностей результатів моделі від параметрів моделі та вхідних змінних.

Якщо модель статистична, необхідно прийняти рішення про те, як задаватиметься вхідна інформація та порядок обробки результатів. З урахуванням цих міркувань розробляється методика проведення експерименту, що містить порядок проведення експериментів та кількість випробувань у кожному експерименті, порядок обробки результатів, способи контролю перебігу експерименту та порядок його коригування залежно від проміжних та кінцевих результатів.

Може використовуватись стратегічне та тактичне планування.

Стратегічне планування має на меті створення загального плану економного експерименту і, відповідно, він передбачає розумну послідовність експериментів і проміжних перевірок.

Тактичне планування пов’язане з розв’язанням задач двох типів:

- визначення початкових умов тією мірою, якою вони впливають на досягнення встановленого режиму, мінімізація витрат на перехідний режим;
- мінімізація дисперсії наслідків при одночасному зменшенні, по можливості, обсягу вибірок.

Результат етапу планування експерименту – план проведення експерименту.

Етап перевірки моделі. Слід зауважити, що модель необхідно перевіряти (випробовувати) постійно з моменту її створення до отримання необхідного результату.

Перевірка моделі перед проведенням експерименту – останній етап розробки моделі. Випробування проводиться з метою:

- виявлення правдоподібності моделі в першому наближенні, щоб переконатися, що модель поводиться, як і передбачалося, тобто існує якісна відповідність між поведінкою модельованої системи та моделі, у тому числі – збігається порядок їх результатів, а також поведінка та результати в граничних умовах;

- перевірки кількісної адекватності – точності перетворення інформації, що досягається калібруванням моделі.

Калібрування моделі – визначення (уточнення) коефіцієнтів моделі, що пов’язують змінні моделі. Калібрування здійснюється шляхом порівняння результатів, отриманих на моделях з результатами, одержуваними при випробуваннях реальної системи або з результатами аналітичних розрахунків, для чого використовуються еталонні приклади та завдання. Модель системи загалом перевіряється еталонними задачами, що охоплюють всі властивості моделі.

Однією з задач випробування є перевірка моделі на чутливість, тобто – наскільки результати моделі чутливі до зміни вхідних змінних.

При випробуваннях широко застосовують такі статистичні методи, як регресійний, кореляційний і дисперсійний аналізи. Важливо пам’ятати, що статистичні методи можуть призвести до невірних результатів, якщо дослідник не має ясного уявлення про систему, що моделюється, і характеристики використовуваної інформації.

Для забезпечення адекватності моделі передбачаються при її розробці та експлуатації такі види контролю:

- контроль розмірностей;
- контроль порядків;
- контроль характеру залежностей між змінними;
- контроль екстремальних ситуацій;
- контроль граничних умов;
- контроль математичної замкнутості;
- контроль стійкості моделі;
- контроль відповідності значень змінних їх фізичному змісту.

Оскільки випробування моделей складних систем пов’язані з суттєвими витратами, необхідно до планування випробувань відноситися так само суворо як і до планування експериментів.

Результати випробувань повинні забезпечити необхідний рівень адекватності моделі на всіх етапах її використання. При обґрунтованому виборі тестових прикладів та еталонних завдань ця задача вирішується за мінімальних витрат коштів та ресурсів.

Етап виконання експерименту з моделлю. Якщо модель аналітична, то експеримент зводиться до виконання розрахунків при варійованих вихідних даних. При імітаційному моделюванні модель реалізують засобами обчислювальної техніки з фіксацією та подальшою обробкою одержуваних даних статистичними та іншими методами. Експерименти проводять відповідно до плану, який може бути включено в алгоритм моделі.

Комп'ютерний експеримент зазвичай містить дві стадії:

- тестування моделі – процес перевірки правильності побудови моделі. На цій стадії перевіряється розроблений алгоритм побудови моделі та адекватність отриманої моделі об'єкта та меті моделювання;
- дослідження моделі – до цієї стадії комп'ютерного експерименту можна переходити лише після того, як тестування моделі пройшло успішно, і є впевненість, що створено саме ту модель, яку потрібно дослідити.

Етап обробки, аналізу та інтерпретації результатів експерименту. Відповідно до мети моделювання застосовують різноманітні методи обробки: визначення різного роду характеристик випадкових величин і процесів, виконання аналізів – дисперсійного, регресійного, факторного тощо. Багато з цих методів входять у системи моделювання (GPSS World, AnyLogic та ін.) і можуть застосовуватися автоматично.

При аналізі результатів необхідно:

1. Переконатися, що результати експерименту цілком зрозумілі, як якісно, так і кількісно.

2. Повернутися до зроблених припущень. Уточнити можливі впливи припущень на результати. За потреби також провести додаткові експерименти.

3. Оцінити точність отриманих результатів. Якщо такі оцінки заздалегідь були заплановані, слід їх зробити. Переконатися, що точність результатів є достатньою для вироблення рекомендацій, прийняття рішення.

Якщо результати не відповідають цілям поставленого завдання, це означає, що на попередніх етапах були допущені помилки. Це може бути або занадто спрощена побудова інформаційної моделі, або невдалий вибір методу або середовища моделювання, або порушення технологічних прийомів при побудові моделі. У таких випадках необхідно коригувати модель, тобто повертатися до одного з попередніх етапів. Процес повторюється до тих пір, поки результати комп'ютерного експерименту не відповідатимуть цілям моделювання.

Етап розробки рекомендацій. Після аналізу результатів моделювання здійснюється їхня інтерпретація, тобто переведення результатів у терміни предметної області. Це необхідно тому, що зазвичай фахівець предметної області (той, кому потрібні результати досліджень) не володіє термінологією математики та моделювання і може виконувати свої завдання, оперуючи лише добре знайомими йому поняттями.

Аналіз результатів моделювання може завершитися виробленням рекомендацій по суті розв'язуваної задачі, проте можлива незадоволеність результатами та підготовка пропозицій щодо проведення додаткових випробувань або уточнення моделі. Не виключається і висновок про непридатність моделі внаслідок її неадекватності досліджуваній системі чи неможливості проведення на моделі необхідного для отримання обґрунтованих висновків обсягу випробувань. Всі результати аналізу повинні представлятися в зручному для використання вигляді.

Етап супроводження. Впровадження прийнятих рекомендацій, одержаних на моделі, має відбуватися за участю осіб, які проводили

модельний експеримент. Тільки в процесі реалізації рекомендацій стає до кінця зрозумілим, наскільки адекватною була модель, наскільки коректно було проведено обчислювальний експеримент та обґрунтовано рекомендації. І лише за участю дослідників можливо найбільш грамотно реалізувати рекомендації, переконатися в їхній справедливості, а в іншому випадку – своєчасно виявити недостатність чи помилковість рекомендацій та ввести необхідні корективи. Процес реалізації рекомендацій може бути керованим, для чого необхідно передбачити оперативний зворотний зв’язок. Запланована участь дослідників у реалізації отриманих рекомендацій забезпечить більш відповідальне ставлення всіх осіб, які брали участь у дослідженнях.

1.3. Вимоги, що висуваються до моделей

Після побудови моделі та отримання на її основі розв’язання задачі наступним важливим кроком є оцінювання її якості. Цей етап є завершальним у процесі створення моделі і має дві основні функції: перевірити, наскільки модель відповідає меті дослідження, та визначити надійність результатів, отриманих у процесі моделювання. На точність цих результатів впливають кілька ключових чинників, зокрема:

- правильний вибір математичного інструментарію;
- методичні похибки, притаманні обраному методу;
- робота моделі в умовах нестаціонарного режиму;
- необхідність узгодження дій окремих складових моделі;
- достовірність моделі зовнішніх впливів.

Загальні зовнішні вимоги до моделей:

1. Модель має бути актуальною. Це означає, що модель має бути націлена на важливі проблеми для осіб, які приймають рішення.

2. Модель має бути результативною. Це означає, що отримані результати моделювання можуть знайти успішне застосування. Цю вимогу

може бути реалізовано тільки у випадку правильного формулювання необхідного результату.

3. Модель має бути достовірною. Це означає, що результатам моделювання можна не сумніватись.

4. Модель має бути економічною. Це означає, що ефект від використання результатів моделювання перевищує витрати ресурсів на її створення і дослідження.

Ці вимоги здійсненні за умови наявності в моделі таких внутрішніх властивостей:

- суттєвість моделі – дає змогу розкрити сутність поведінки системи, розкрити неочевидні, нетривіальні деталі;
- потужність моделі – дає змогу отримати широкий набір істотних відомостей;
- простота моделі у вивченні та використанні – легко прораховується засобами обчислювальної техніки;
- відкритість моделі – дозволяє її модифікацію.

Деякі властивості моделі, що дозволяють у тій чи іншій мірі або розрізняти, або ототожнювати модель з оригіналом (об’єктом, процесом):

- цілеспрямованість – модель завжди відображує якусь систему, тобто має певну мету;
- обмеженість (повнота) – у моделі повинні бути передбачені всі основні зв’язки і відношення, необхідні для забезпечення цілі моделювання;
- складність (спрощеність) – модель відображає тільки суттєві сторони об’єкта і, крім того, повинна бути простою для дослідження і відтворення;
- прийнятна трудомісткість розробки моделі для існуючого часу та програмних засобів;

- точність (наближеність) по відношенню до реальної системи – модель повинна містити достатню інформацію про систему і повинна надавати можливість одержувати нову інформацію;
- гнучкість – можливість відтворення різних ситуацій у всьому діапазоні зміни умов та параметрів;
- адекватність – слугує для оцінки рівня виконання вимог повноти та точності, необхідного для досягнення мети моделювання. Ступінь адекватності моделі перевіряється експериментальним шляхом на основі введення міри адекватності.

Важливо пам’ятати, що системні аналітики використовують моделі для ухвалення рішень і пошуку оптимальних способів побудови або вдосконалення модельованої системи. Інформація, отримана з моделі, насамперед має бути корисною для замовника – саме він є її основним користувачем. Тому створення моделі, яка не буде впроваджена або використана на практиці, є недоцільним. Робота з моделлю має бути максимально автоматизована і адаптована під потреби замовника, щоб той міг застосовувати її в межах своєї професійної сфери. Отже, між моделлю та користувачем необхідно забезпечити зручний інтерфейс, найчастіше реалізований у вигляді спеціального меню, адаптованого до конкретної предметної області. Рівень деталізації моделі слід визначати, виходячи з цілей моделювання, доступності вхідних даних і наявних ресурсів.

Модель призначена для заміни оригіналу під час досліджень, яким піддавати оригінал не можна або недоцільно. Але заміна оригіналу моделлю можлива, якщо вони достатньою мірою схожі або адекватні.

Саме тому однією з найважливіших вимог до моделі є вимога її адекватності, тобто відповідності реальному об’єкту (процесу, системі тощо) за певним набором ключових характеристик і властивостей.

Адекватність моделі означає правильне якісне та кількісне відображення досліджуваного об’єкта відповідно до обраних параметрів із прийнятним рівнем точності [12]. Важливо розуміти, що мова йде не про

загальну відповідність, а саме про адекватність тих властивостей, які мають значення для дослідника. Повна адекватність передбачає повне збігання моделі з оригіналом, тобто з прототипом.

Математична модель може бути адекватною для одних умов (комбінації внутрішніх станів системи та зовнішнього середовища), але непридатною для інших. Наприклад, функціональна модель типу «чорного ящика» вважається адекватною, якщо вона з певною точністю імітує поведінку реальної системи, тобто перетворює вхідні сигнали у вихідні так само, як і об’єкт, що моделюється.

Існує поняття ступеня (міри) адекватності, яке може змінюватися в діапазоні від 0 (повна неадекватність) до 1 (повна відповідність). Ця величина характеризує, наскільки модель відповідає реальному об’єкту за обраними ознаками. Введення такої кількісної міри дозволяє розв’язувати важливі завдання, пов’язані з ідентифікацією, стійкістю, чутливістю, адаптацією та навчанням моделі.

У деяких випадках чисельно оцінити ступінь адекватності неважко – наприклад, під час апроксимації експериментальних даних функцією.

Варто пам’ятати, що будь-яка адекватність є відносною та обмеженою в сфері застосування. Основні труднощі при оцінюванні адекватності полягають у нечіткості критеріїв і складності вибору параметрів, за якими її слід визначати. Оскільки адекватність є раціональною характеристикою, її підвищення можливе через логічні, методичні і експериментальні кроки. У процесі дослідження модель повинна перевірятись, уточнюватись та аналізуватись на прикладах, аналогіях та експериментах. Така перевірка дозволяє зрозуміти, чи призводять зроблені спрощення до прийнятної втрати точності, чи, навпаки, до істотного погіршення результатів.

Перевірку адекватності проводять на всіх етапах побудови моделі, починаючи з найпершого етапу. Якщо опис системи буде складено не адекватно реальній системі, то і модель, як би точно вона не відображала опис системи, не буде адекватною оригіналу. Якщо вивчення системи

проведено якісно і формалізована модель досить точно відображає реальний стан справ, то далі перед розробниками стоїть лише проблема еквівалентного перетворення одного опису в інший.

Також існує поняття побічної адекватності – коли модель, створена для імітації певних властивостей, несподівано точно відображає і додаткові характеристики, важливість яких може проявитися пізніше. Такий ефект посилюється, якщо модель ґрунтується на перевірених фізичних законах, системних принципах, геометричних закономірностях або добре апробованих методах. Зазвичай структурні моделі мають вищу побічну адекватність, ніж функціональні.

Деякі дослідники розглядають ціль як об’єкт моделювання. У такому випадку адекватність моделі визначається як ступінь наближення до цієї мети або як ефективність її досягнення.

Отже, можна говорити про адекватність моделі в будь-якій її формі та оригіналу, якщо:

- опис поведінки, створений на певному етапі, досить точно збігається з поведінкою модельованої системи в однакових ситуаціях;
- опис переконливо показовий щодо властивостей системи, які мають прогнозуватися за допомогою моделі.

Для математичної моделі вихідний варіант піддається перевіркам на:

- урахування в моделі всіх істотних параметрів;
- відсутність у моделі несуттєвих параметрів;
- слушність відображення функціональних зв’язків між параметрами;
- слушність визначених обмежень на значення параметрів;
- відсутність нелогічної поведінки моделі при граничних значеннях її параметрів.

Така попередня оцінка адекватності моделі дає змогу виявити в ній найбільш грубі помилки. Але усі ці рекомендації мають неформальний, рекомендаційний характер. Остаточне судження про адекватність моделі може дати лише практика, тобто порівняння моделі з оригіналом на основі експериментів з об'єктом і моделлю, а створення точних і надійних методів перевірки адекватності залишається складною і до кінця не вирішеною проблемою.

Спрощеність (простота) **і складність моделі.** Вимоги простоти і адекватності моделі зазвичай суперечать одна одній. Очевидно, що якщо орієнтуватися на вимогу адекватності моделі, то доведеться віддавати перевагу складним моделям, оскільки вони дозволяють врахувати більше чинників, що впливають на досліджувані властивості об'єкта. Але дуже часто ускладнення моделей призводить до неможливості їх дослідження: такі моделі точніше відображають реальний прототип, але водночас вони стають більш громіздкими, важкими для сприйняття та складними в практичному використанні.

Таким чином, достатньо важливими є вимоги достатньої простоти моделі по відношенню до обраної системи характеристик. Тому певною мірою є мистецтвом вибір оптимального співвідношення між адекватністю та простотою моделі. З цієї причини дослідники часто прагнуть спростити модель, оскільки з простими моделями легше працювати. При цьому слід дотримуватися основного принципу: модель можна спрощувати лише до того рівня, поки зберігаються основні характеристики, властивості й закономірності оригіналу.

На рис. 1.8 наведено принцип вибору моделі з розумним співвідношенням між простотою і адекватністю.

Слід враховувати, що поняття простоти або складності моделі є відносним. Модель вважається достатньо простою, якщо сучасні методи (математичні, інформаційні чи фізичні) дозволяють якісно і кількісно її дослідити з необхідною точністю. З розвитком технологій і зростанням

обчислювальних можливостей ті задачі, які раніше вважалися складними, можуть зараз сприйматися як прості. До того ж простота моделі часто визначається і її психологічним сприйняттям дослідником.

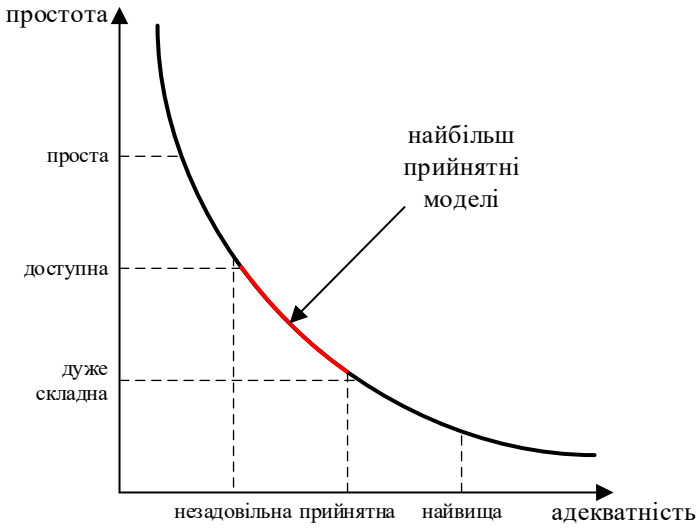


Рисунок 1.8 – Принцип вибору моделі

Щоб уникнути значної втрати адекватності, спрощення моделей бажано здійснювати на таких рівнях:

- на фізичному рівні, зберігаючи основні фізичні закономірності;
- на структурному рівні, з урахуванням ключових системних властивостей.

Спрощення на математичному (абстрактному) рівні може суттєво знизити достовірність моделі.

Загалом, при синтезі систем часто використовуються спрощені моделі, тоді як при аналізі систем потрібні детальніші й точніші моделі.

Обмеженість (повнота) моделей. Під час дослідження будь-якого об’єкта завжди обмежуються скінченним числом його властивостей, зв’язків, характеристик і доступних ресурсів.

Обмеженість моделі проявляється в двох аспектах:

- обмежене відображення реальності – модель охоплює лише певну кількість властивостей, зв’язків та структур

досліджуваного об’єкта в межах поточного рівня аналізу й наявних ресурсів;

- обмеженість ресурсів – моделювання завжди обмежене наявними інформаційними, фінансовими, енергетичними, часовими та технічними ресурсами, а також обсягом знань.

Розмірність моделі (тобто кількість змінних станів або ступенів свободи) тісно пов’язана з класом розв’язуваних задач. Збільшення розмірності зазвичай підвищує адекватність, але водночас ускладнює модель. Рівень складності обмежується можливістю дослідника оперувати з моделлю, тобто залежить від обраних засобів моделювання. Переходячи від грубої моделі до точнішої, доводиться вводити додаткові змінні, які були проігноровані на ранньому етапі. Ці змінні зазвичай належать до трьох категорій:

- швидкоплинні змінні – зміни відбуваються настільки швидко, що в спрощеній моделі вони враховувалися лише через усереднені або інтегральні показники;
- повільнозмінні змінні – змінюються настільки повільно, що в грубій моделі вважалися постійними;
- малі змінні (параметри) – настільки незначні за впливом, що ними нехтували при спрощеному моделюванні.

Розподіл складного руху системи на швидкі та повільні компоненти дозволяє вивчати їх окремо в грубому наближенні, що спрощує аналіз. Малі змінні зазвичай ігнорують при синтезі, але враховують при аналізі системи, оскільки вони можуть впливати на її точні властивості.

У процесі моделювання прагнуть відокремити невелику кількість ключових факторів, які мають вплив одного порядку і можуть бути описані відносно просто математично. Вплив решти факторів враховується через усереднені, інтегральні або зафіксовані характеристики. Варто зазначити, що

один і той самий фактор може по-різному впливати на різні властивості системи.

У більшості випадків врахування трьох наведених класів змінних є достатнім для створення моделі, що забезпечує належний рівень достовірності та ефективності аналізу.

Точність (наближеність) моделей. Обмеженість і простота (або спрощеність) моделі визначають якісну відмінність між моделлю та її оригіналом на структурному рівні. У свою чергу, наближеність моделі відображає кількісну сторону цієї відмінності.

Можна ввести кількісну міру наближеності, порівнюючи, наприклад, спрощену (грубу) модель із більш точною, еталонною (повною, ідеальною) моделлю або з реальною системою. Модель, як окремий об’єкт, завжди відтворює лише окремі властивості свого прототипу. Ступінь наближеності (або точності, відповідності) моделі до оригіналу визначається метою моделювання та постановкою задачі. Спроба досягти максимальної точності часто призводить до надмірного ускладнення моделі. А це, у свою чергу, знижує її практичну цінність, тобто можливість ефективного застосування на практиці. Основною причиною цього є невизначеність і нечіткість знань про сам об’єкт дослідження: про його поведінку, властивості, впливи зовнішнього середовища, мислення і поведінку людини, механізми формування цілей, засоби й способи їх досягнення тощо.

Завдання дослідника полягає у виборі оптимального балансу між точністю та практичністю, залежно від цілей і ресурсів моделювання.

Якщо модель і об’єкт піддаються однаковим впливам і порівнюються їхні реакції, то, якщо ці реакції однакові (у межах допустимої точності), то робиться висновок, що модель адекватна оригіналу. Однак треба мати на увазі, що:

- впливи на об’єкт мають обмежений характер через можливе руйнування об’єкта, недоступність до елементів системи тощо;

- впливи на об’єкт мають фізичну природу, а на математичну модель – це числові аналоги фізичних впливів.

Для оцінки ступеня подібності структур об’єктів (фізичних або математичних) існує поняття ізоморфізму. Дві системи ізоморфні, якщо існує взаємно однозначна відповідність між елементами і відносинами (зв’язками) цих систем.

Під час моделювання складних систем досягти повної відповідності важко, та й недоцільно. Під час моделювання абсолютна подібність не має місця. Прагнуть лише до того, щоб модель досить добре відображала досліджувану сторону функціонування об’єкта: при абсолютній подібності модель за складністю може стати аналогічною досліджуваній системі і ніякого спрощення дослідження не буде.

Для оцінки подібності в поведінці (функціонуванні) систем існує поняття ізофункціоналізму: дві системи довільної, а часом невідомої структури ізофункціональні, якщо за однакових впливів вони виявляють однакові реакції. Моделювання при цьому називається функціональним або кібернетичним. Функціональні моделі не копіюють структури.

Під час побудови математичних моделей об’єктів, систем чи процесів доцільно дотримуватися таких рекомендацій:

- починати моделювання слід зі створення найпростішої (грубої) моделі, зосереджуючи увагу лише на найбільш суттєвих факторах. При цьому важливо чітко усвідомлювати як мету моделювання, так і пізнавальні задачі, які розв’язуються за допомогою моделі;
- слід уникати використання штучних або складних гіпотез, щоб не ускладнювати перевірку адекватності моделі;
- потрібно контролювати розмірність змінних, дотримуючись правила: додавати або порівнювати можна лише величини

однакової розмірності. Це правило варто застосовувати на всіх етапах виведення математичних співвідношень;

- слід контролювати порядок величин, щоб виділити головні фактори та змінні і відкинути другорядні. Це дозволяє зберігати простоту моделі: відкидання незначних величин не повинно суттєво змінювати кількісні результати і порушувати якісні висновки;
- важливо перевіряти характер функціональних залежностей, дотримуючись правила: зміна напрямку і швидкості одних змінних повинна відповідати зміні інших. Це дозволяє краще зрозуміти фізичну суть і коректність отриманих формул чи рівнянь;
- необхідно аналізувати поведінку моделі при наближенні її параметрів до критичних значень або особливих точок. У таких умовах модель часто спрощується, набуває наочності, а результати легше перевірити або підтвердити альтернативними методами. Аналіз таких граничних випадків може дати уявлення про асимптотичну поведінку системи;
- модель повинна коректно поводитися за відомих умов: вона повинна задовольняти граничним умовам задачі, а також демонструвати адекватну реакцію на стандартні вхідні сигнали, що дозволяє перевірити її працездатність у типових ситуаціях;
- слід звертати увагу на побічні результати, які можуть з’являтися в процесі дослідження. Такий аналіз може підказати нові напрями наукових пошуків або вказати на потребу змінити структуру моделі.

Постійний контроль за коректністю функціонування моделі на всіх етапах моделювання допомагає уникати серйозних помилок у кінцевих результатах. При цьому, недоліки не передбачаються наперед, а виявляються

та усуваються в процесі дослідження, що є природною складовою розвитку та уточнення моделі.

Контрольні запитання

1. Що таке модель? Розкрийте зміст фрази: «модель є об’єкт і засіб експерименту».

2. Обґрунтуйте необхідність моделювання.

3. На основі якої теорії ґрунтується моделювання.

4. Що таке система? Що визначають стани системи?

5. Які існують зв’язки між входами та виходами системи?

6. Яким чином можуть описуватись системи?

7. Назвіть загальні класифікаційні ознаки моделей.

8. Чи потрібно прагнути до абсолютної подібності моделі й оригіналу?

9. Наведіть приклади об’єктів і можливих їхніх моделей у своїй предметній області.

10. Що означає структурна модель?

11. Що таке функціональна модель?

12. Класифікація моделей за характером процесів, що протікають у модельованих об’єктах.

13. Сутність математичного моделювання та його основних класів: аналітичного та імітаційного.

14. Назвіть етапи моделювання і дайте їм коротку характеристику.

15. Що таке адекватність моделі? Дайте поняття ізоморфізму та ізофункціоналізму.

16. Які існують форми запису моделей? Наведіть приклади опису.

17. Для чого виконується калібрування моделі?

18. Загальні вимоги (зовнішні) до моделей.

19. Внутрішні властивості моделі.

20. Які властивості моделі дозволяють її ототожнювати з оригінальним об’єктом?

2. ТИПОВІ МАТЕМАТИЧНІ МОДЕЛІ

2.1. Дискретні марковські процеси. Схема загибелі та розмноження

Марковська модель – це математична концепція, яка використовується для моделювання систем, що переходять із одного стану в інший, при цьому ймовірність кожного переходу залежить лише від поточного стану, а не від попередньої послідовності подій [8]. Ця властивість робить такі моделі надзвичайно корисними в статистиці, аналізі даних та Data Science. Існує кілька різновидів марковських моделей, серед яких виділяють дискретні марковські ланцюги та неперервні марковські процеси – вибір залежить від типу даних і характеру завдання.

Марковська модель складається з кількох ключових складових:

- стани $s_i, i = \overline{1,n}$ – можливі конфігурації або ситуації, в яких може перебувати система;
- ймовірності переходу $p_i(t)$ – числові значення, що вказують ймовірність зміни одного стану на інший у момент часу t;
- початковий розподіл – визначає, з якого стану і з якою ймовірністю починається процес.

Разом ці елементи формують структуру моделі і забезпечують її функціональність, дозволяючи здійснювати аналіз і прогнозування.

Як було вказано, існують два основні типи марковських моделей:

- марковські ланцюги з дискретним часом – використовуються, коли зміни відбуваються у визначені моменти часу;
- марковські процеси з неперервним часом – підходять для систем, що змінюються постійно, без фіксованих інтервалів.

Процес називається процесом з дискретними станами, якщо його можливі стани s_1, s_2, ... можна заздалегідь перерахувати, тобто стани системи належать скінченній множині $E = \{s_1, s_2, \dots, s_n\}$, і перехід системи з одного стану в інший відбувається миттєво. У загальному випадку число станів

може бути як скінченним, так і нескінченним (але зліченним). Властивість відсутності післядії називають також властивістю відсутності пам’яті, а марковські процеси – процеси без пам’яті.

Процес називається процесом з безперервним часом, якщо зміна станів може статися будь-якої миті.

Відповідно, дискретним марковським ланцюгом називається випадковий процес, в якому зміна дискретних станів відбувається в певні моменти часу, а безперервним марковським процесом – випадковий процес, в якому зміна дискретних станів відбувається у випадкові моменти часу.

Марковські моделі широко застосовуються в:

- фінансах – для прогнозування динаміки цін на ринку;
- економіці – для моделювання змін у станах економічних агентів;
- генетиці – для аналізу послідовностей ДНК;
- машинному навчанні – особливо в навчанні з підкріпленням, де агент вчиться оптимальним діям шляхом взаємодії з середовищем.

Дискретні моделі ефективні в теорії масового обслуговування, тоді як неперервні – у популяційній динаміці чи фінансовому аналізі.

Попри свою ефективність, марковські моделі мають деякі обмеження:

- вони базуються на припущенні, що майбутній стан залежить лише від поточного, що не завжди відповідає реальним умовам;
- у складних системах матриця переходів може мати дуже велику розмірність, що ускладнює обчислення.

До марковських дискретних моделей належить широкий клас найрізноманітніших об’єктів, що мають загальну назву – системи масового обслуговування.

В основі теорії масового осблуговування лежить поняття приладу, який може виконувати скінченну множину операцій. Прилад виконує операцію, коли виникає заявка – вимога виконання операції. Якщо прилад виконує

будь-яку операцію, то вважається, що він зайнятий (працює), інакше прилад вільний.

Послідовність заявок у часі називається потоком заявок. Загальний потік заявок може складатися з кількох потоків. У випадках незалежності потоків, випадкових моментів надходження або завершення обслуговування заявок у системі можуть виникати черги. Черга – це заявки, які очікують на обслуговування, коли прилад зайнятий. Прилад може складатися з кількох елементів (каналів), кожен з яких здатний обслужити будь-яку заявку. Сукупність приладу, потоків заявок і черг до нього називають системою масового обслуговування (СМО). Теорія масового обслуговування добре розроблена. Тому вона знайшла широке застосування для створення математичних моделей.

Найповніше дослідження процесу функціонування систем можна отримати, якщо відомі явні математичні залежності, що пов'язують шукані показники з початковими умовами, параметрами і змінними досліджуваної системи. Але для багатьох сучасних систем таких математичних залежностей немає або вони малопридатні, і слід застосовувати імітаційне моделювання.

Для багатоелементних систем з великим числом станів аналітичне моделювання на основі теорії марковських процесів стає досить громіздким. У цьому разі використовується так званий метод динаміки середніх, який в основі має також марковість процесу. Цей метод істотно спрощує аналітичне моделювання для випадків визначення середніх характеристик станів модельованої системи.

Для опису поведінки системи S у вигляді марковської моделі слід визначити поняття стану системи; виявити всі стани $s_i,\ i = \overline{1,n}$, в яких може перебувати система S; вказати, в якому стані знаходиться система S у початковий момент часу $t = 0$; побудувати граф станів, тобто зобразити всі стани, наприклад, колами та можливі переходи зі стану в стан – стрілками, що з'єднують стани; розмітити граф, тобто для кожного переходу зі стану s_i

у стан s_j вказати ймовірності переходів p_{ij}. Можливість системи залишитися в наступний момент часу в тому ж самому стані зображується петлею, що виходить і входить в одну й ту ж саму вершину (рис. 2.1).

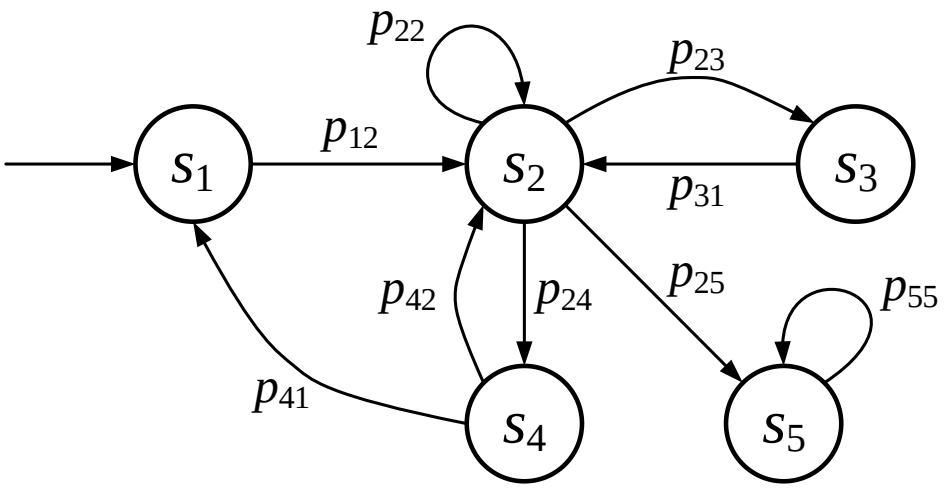


Рисунок 2.1 – Розмічений граф станів системи S

Наявність на розміченому графі стрілок і відповідних їм ймовірностей з одного стану в інший означає, що ці перехідні ймовірності відмінні від нуля. Навпаки, відсутність стрілок з одного стану в інший вказує на те, що відповідні їм перехідні ймовірності дорівнюють нулю.

Випадкова послідовність називається ланцюгом Маркова, якщо для кожного кроку ймовірність переходу з будь-якого стану s_i в будь-який стан s_j не залежить від того, коли і як система S виявилася в стані s_i.

Очевидно, що реалізацію дискретного випадкового процесу з дискретним часом за будь-який (скінченний) проміжок часу можна представити невипадковою скінченною послідовністю (за індексом k) розглядуваних подій $S_i(k)$ $(i = \overline{1,n};\ k = 0,1,2,...)$. Наприклад, для графа станів системи S, наведеного на рис. 2.1, в якій відбувається випадковий процес із дискретним часом у початковий момент $k = 0$ система знаходилася в стані s_1; у момент першого кроку перейшла з нього стрибком у стан s_2, з якого на другому кроці перейшла в s_4, на третьому кроці перейшла в s_2, на четвертому кроці перейшла в s_3, після п’ятого кроку перейшла в s_2,

після шостого – у s_2, після сьомого – у s_5. Тоді реалізація випадкового процесу блукання по станах (траєкторія ланцюга Маркова) має такий вигляд: $S_1(0), S_2(1), S_4(2), S_2(3), S_3(4), S_2(5), S_2(6), S_5(7)$.

Більш змістовне визначення ланцюга Маркова можна сформулювати так: послідовність випадкових величин $S(k),\ k = 0,1,2, \ldots$, називається ланцюгом Маркова з множиною станів $E = \{1, 2, \ldots, n\}$, якщо

$$\sum_{i=1}^{n} P(S(k) = i) = 1, k = 0, 1, 2, \ldots$$

і для будь-яких $0 \le k_1 < k_2 < \cdots < k_m < l < k\ \ (m = 1, 2, \ldots)$, будь-яких $i, j \in E$ та будь-яких підмножин $B_1, \ldots, B_m$ множини E виконується рівність

$$P(S(k) = j | S(k_1) \in B_1, \ldots, S(k_m) \in B_m, S(l) = i) = P(S(k) = j | S(l) = i).$$

Ця властивість означає, що при фіксованому значенні системи в момент часу l поведінка системи в майбутньому не залежить від поведінки системи в минулому і має назву марковської властивості.

Звідси, ланцюг Маркова може бути заданий розподілом ймовірностей переходу за один крок.

Ймовірністю переходу (перехідною ймовірністю) $p_{ij}(k)$ на k–му кроці зі стану s_i у стан s_j $(i,\ j = 1, 2, \ldots, n;\ k = 1, 2, \ldots)$ називається умовна ймовірність того, що система S після k–го кроку виявиться в стані s_j за умови, що безпосередньо перед цим (після $(k-1)$-го кроку) вона знаходилася в стані s_i:

$$p_{ij}(k) = P\big(S(k) = s_j | S(k-1) = s_i\big), i, j = 1, \ldots, n.$$

У випадку, коли $i = j$, то перехідна ймовірність $p_{ij}(k) = p_{ii}(k)$ називається ймовірністю затримки системи S у стані s_i. Відповідно, на графі станів, якщо стрілка виходить із стану s_i та повертається в стан s_i, то це означає затримку системи в цьому стані.

Стан s_i називається джерелом, якщо система може вийти з цього стану, але повернутися в нього вже не може.

Якщо перехідні ймовірності $p_{ij}(k)$ $(i,j=1,\dots,n;k=1,2,\dots)$ не залежать від номера кроку k (від часу), а залежать лише від того, з якого стану в який здійснюється перехід, то ланцюг Маркова – однорідний. Інакше, якщо хоча б одна ймовірність змінюється зі зміною кроку k, то ланцюг Маркова – неоднорідний.

Для однорідних ланцюгів Маркова перехідні ймовірності можна позначати через p_{ij} замість $p_{ij}(k)$, а всю сукупність ймовірностей переходу p_{ij} $(i,j=1,\dots,n)$ можна зобразити у вигляді однорідної матриці переходів (перехідних ймовірностей):

$$P=\begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{pmatrix},$$

усі елементи якої невід'ємні та елементи кожного рядка задовольняють умові

$$\sum_{j=1}^{n} p_{ij}=1, i=1,\dots,n.$$

Деякі з перехідних ймовірностей можуть дорівнювати нулю (це означає, що перехід системи за один крок зі стану s_i у стан s_j неможливий), на головній діагоналі матриці P розташовані ймовірності затримки системи в тому ж стані на наступному кроці.

Перехідні ймовірності, що відповідають стрілкам на розміченому графі і виходять зі станів s_i $(i=1,\dots,n)$, розміщені в i–му рядку матриці P перехідних ймовірностей, їх сума дорівнює одиниці. Тому ймовірності затримок $p_{ii}(i=1,\dots,n)$ можна обчислити за формулою

$$p_{ii}=1-\sum_{\substack{j=1 \\ i\neq j}}^{n} p_{ij}\,, i=1,\dots,n.$$

Це дає змогу не відзначати на графі петлі (рис. 2.2), що може в певних випадках значно зменшити завантаженість зображення графічними елементами.

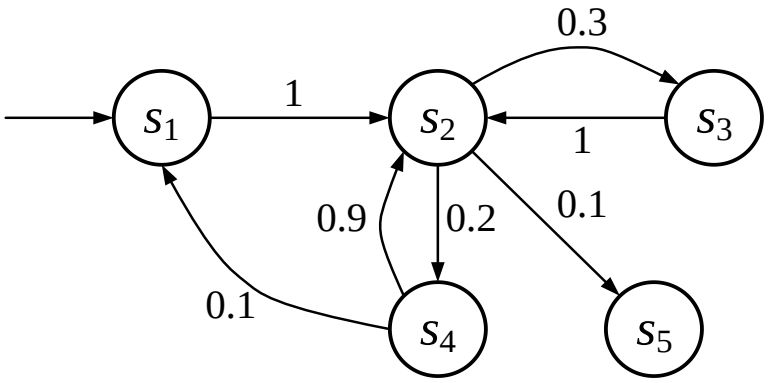


Рисунок 2.2 – Розмічений граф станів системи S (варіант 2)

Матриця переходів для розміченого графу станів системи, наведеного на рис. 2.2, виглядатиме таким чином:

$$P = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0.4 & 0.3 & 0.2 & 0.1 \\ 0 & 1 & 0 & 0 & 0 \\ 0.1 & 0.9 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Слід зазначити, що кінцевий дискретний ланцюг Маркова є подальшим узагальненням схеми Бернуллі.

Схемою Бернуллі називається послідовність незалежних у сукупності випробувань, у кожному з яких можливі лише два результати – «успіх» і «невдача», при цьому успіх в одному випробуванні відбувається з ймовірністю p, а невдача з ймовірністю $q = 1 - p$.

Заміна умови незалежності дослідів у схемі Бернуллі умовою залежності кожного випробування тільки від попереднього призводить до ланцюгів Маркова. Схема Бернуллі розглядається як частковий випадок ланцюга Маркова з двома станами: s_1 – наслідком окремого випробування є подія A і s_2 – наслідком окремого випробування є подія $\overline{A}$, в якому матриця перехідних ймовірностей P має вигляд

$$P = \begin{pmatrix} p & q \\ p & q \end{pmatrix}.$$

Відповідний розмічений граф ланцюга Маркова для схеми Бернуллі наведено на рис. 2.3.

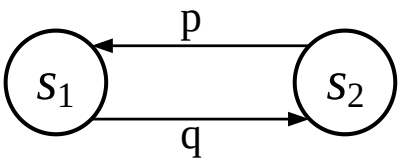


Рисунок 2.3 – Розмічений граф станів схеми Бернуллі

Існує тип ланцюга Маркова, в якому при досягненні певного стану вийти з цього стану неможливо: стан s_i називається кінцевим (або поглинаючим), якщо система може попасти в цей стан, але вийти з нього вже не може. Ланцюги Маркова, що мають хоча б один поглинаючий стан, та з будь-якого іншого стану можна потрапити за один або кілька кроків у один із поглинаючих станів, називаються **поглинаючими ланцюгами Маркова**. На рис. 2.2 стан s_5 є поглинаючим, і в нього є перехід, тому наведений ланцюг Маркова також – поглинаючий.

Поглинаючі ланцюги Маркова використовуються для розв'язання задачі «розорення азартного гравця», вперше вирішеній Бернуллі та Гюйгенсом.

Формулювання задачі: гравець G_m (з капіталом m) грає в азартну гру з гравцем G_M (з капіталом M), беручи участь у серії послідовних партій гри. У результаті кожної партії капітал гравця G_m з ймовірністю p збільшується на 1 (за рахунок гравця G_M) та з ймовірністю $q = 1 - p$ зменшується на 1 (на користь гравця G_M). Результат кожної партії залежить від результатів попередніх партій. Якщо капітал одного з гравців стає рівним нулю, гравець розоряється, гра припиняється. Знайти ймовірність розорення гравця G_M.

Ланцюг Маркова, що відображає цю задачу, має $N + 1$ можливих станів, де $N = m + M$ – сумарний капітал гравців, а саме: $E = \{0,1,2, \dots, N\}$.

Перехідні ймовірності в цьому випадку дорівнюють: $p_{0i} = 0$ для $i \neq 0$, $p_{00} = 1$; $p_{Ni} = 0$ для $i \neq N$, $p_{NN} = 1$; $p_{i,i+1} = p$, $p_{i,i-1} = q$ для $i = 1, \dots, N-1$; $p_{ij} = 0$ для $i = 1,2, \dots, N-1$; $j \neq i-1, i+1$. Матриця перехідних ймовірностей P має вигляд:

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ q & 0 & p & 0 & \dots & 0 & 0 & 0 \\ 0 & q & 0 & p & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & q & 0 & p \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & 1 \end{pmatrix}.$$

Стани $1, 2, \dots, N-1$ – несуттєві. Стани 0 і N є поглинаючими, ланцюг, потрапивши в поглинаючий стан, залишається в ньому назавжди.

Розмічений граф станів ланцюга Маркова для вирішення задачі «розорення азартного гравця» наведено на рис. 2.4.

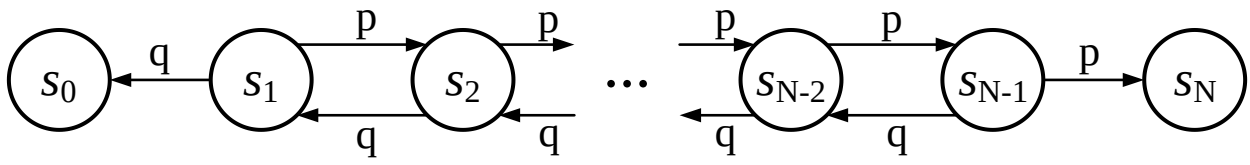


Рисунок 2.4 – Розмічений граф станів ланцюга Маркова для вирішення задачі «розорення азартного гравця»

Під час аналізу ланцюгів Маркова доводиться мати справу з низкою його похідних характеристик. Найчастіше використовуються:

- ймовірність переходу за k кроків;
- безумовна (абсолютна) ймовірність $p_j(k)$ того, що система після k-го кроку знаходиться в стані s_j;
- ймовірності першого досягнення стану s_j при виході зі стану s_i;
- ймовірність f_{ij} того, що, починаючи зі стану s_i, система коли-небудь потрапить в стан s_j;
- середнє число кроків μ_{ij} до першого досягнення стану s_j при виході зі стану s_i.

Ймовірність переходу за k кроків. Припустимо, що стан s_i отримано після l-го кроку. Необхідно розрахувати ймовірність переходу $p_{ij}^{(k)}$ з цього стану в стан s_j за k кроків (після $(l+k)$-кроку). Внаслідок однорідності ланцюга Маркова ця ймовірність залежить тільки від k (не залежить від l):

$$p_{ij}^{(k)} = P\{S(l+k) = j | S(l) = i\}.$$

Якщо визначити через $p_{iv}^{(m)}$ ймовірність переходу за m кроків зі стану s_i у проміжний стан s_v та через $p_{vj}^{(k-m)}$ ймовірність переходу за $(k-m)$ кроків з проміжного стану s_v у стан s_j, використовуючи формулу повної ймовірності

$$p_j(k) = \sum_{i=1}^{n} p_i(k-1)p_{ij}\,, \qquad j = 1,2,\dots,n; k = 1,2,\dots,$$

та враховуючи, що проміжні стани $s_1,\dots,s_n$ на $(l+m)$–му кроці утворюють повну групу попарно несумісних подій, то

$$p_{ij}^{(k)} = \sum_{v=1}^{n} p_{iv}^{(m)} p_{vj}^{(k-m)},$$

де m – будь-яке ціле значення від 1 до $k-1$.

Цю формулу називають рівністю Маркова.

Позначимо через P_k матрицю, яка складається з ймовірностей $p_{ij}^{(k)}$, $i,j = 1,\dots,n$, тобто P_k – матриця переходів через k кроків (при чому $P_1 = P$).

Ергодична теорема Маркова: якщо при деякому досить великому значенні $k > 0$ всі елементи матриці переходів P_k за k кроків позитивні, тобто немає 0, тоді існують такі постійні значення p_j, то справедлива рівність: $p_{ij}^{(k)} = p_j, j = \overline{1,n}$.

Суть теореми: ймовірності p_j можна розглядати як ймовірності того, що система перебуває в стані s_j після k кроків. Враховуючи формулу для перемноження квадратних матриць, рівність Маркова можна записати в матричному вигляді:

$$P_k = P_m \cdot P_{k-m}, \qquad m = 1,\dots,k-1,$$

при послідовному застосуванні якого можна отримати:

$$P_k = P_1 \cdot P_{k-1} = P_1 \cdot P_1 \cdot P_{k-2} = \dots = P^k.$$

Тобто для того, щоб отримати матрицю ймовірностей переходів за k кроків, потрібно просто піднести до k-го ступеню вихідну матрицю ймовірностей переходів P.

Матриця P^k при будь-якому натуральному k є стохастичною.

Приклад. Необхідно визначити ймовірність переходу за 2 кроки ($k = 2$) зі стану s_1 у стан s_3 для розміченого графу станів ланцюга Маркова, наведеного на рис. 2.5. Петлі на графі зображені для наочності.

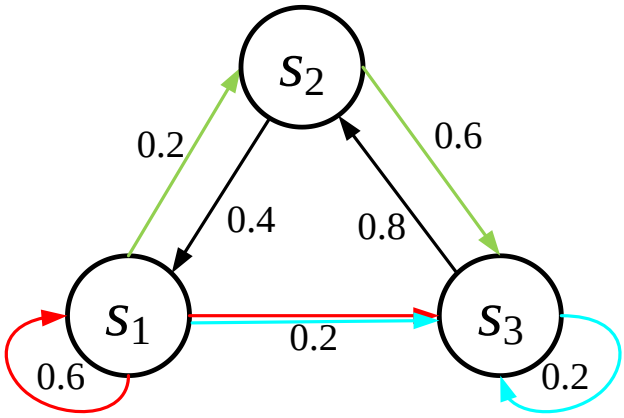


Рисунок 2.5 – Розмічений граф станів ланцюга Маркова для прикладу

Зі стану s_1 у стан s_3 можна потрапити трьома шляхами (відмічені різними кольорами на рис. 2.5). Матриця переходів виглядатиме таким чином:

$$P = \begin{pmatrix} 0.6 & 0.2 & 0.2 \\ 0.4 & 0 & 0.6 \\ 0 & 0.8 & 0.2 \end{pmatrix}.$$

Тоді:

$$p_{13}^{(2)} = p_{11}p_{13} + p_{12}p_{23} + p_{13}p_{33} = 0.6 \cdot 0.2 + 0.2 \cdot 0.5 + 0.2 \cdot 0.2 = 0.28.$$

Або, як результат добутку першого рядка та третього стовпця матричної операції (в обчисленні наведені тільки задіяні в розрахунку елементи):

$$P^2 = \begin{pmatrix} 0.6 & 0.2 & 0.2 \\ . & . & . \\ . & . & . \end{pmatrix} \cdot \begin{pmatrix} . & . & 0.2 \\ . & . & 0.6 \\ . & . & 0.2 \end{pmatrix} = \begin{pmatrix} . & . & 0.28 \\ . & . & . \\ . & . & . \end{pmatrix}.$$

Безумовна (абсолютна) ймовірність $p_j(k)$ того, що система після k-го кроку знаходиться в стані s_j. Для розрахунку використовується наслідок формули повної ймовірності:

$$p_j(k) = \sum_{i=1}^{n} p_i(0) p_{ij}^{(k)}, \qquad j = 1, \dots, n; k = 1,2, \dots .$$

Також можна отримати матричну форму запису. Якщо $p^{(k)} = \left(p_1(k), p_2(k), \dots, p_n(k)\right)$ $(k = 0,1,2, \dots)$, то

$$p^{(k)} = p^{(k-1)} P, \qquad k = 1,2, \dots ,$$

або

$$p^{(k)} = p^{(0)} P^k , \qquad k = 1,2, \dots .$$

Ймовірності першого досягнення стану s_j при виході зі стану s_i. Введемо до розгляду ймовірність $f_{ij}^{(k)}$ того, що, починаючи зі стану i, ланцюг Маркова вперше досягає стан j на k-му кроці. $\left(f_{ij}^{(0)} = 0, i \neq j\right)$, тобто

$$f_{ij}^{(k)} = P\{S(k) = j, S(k-1) \neq j, \dots, S(1) \neq j | S(0) = i\}.$$

Спираючись на раніше наведену формулу повної ймовірності, $f_{ij}^{(k)}$ можуть бути обчислені через ймовірності $p_{ij}^{(k)}$ внаслідок рекурентного застосування формули:

$$p_{ij}^{(k)} = \sum_{\nu=1}^{k} f_{ij}^{(\nu)} p_{jj}^{(k-\nu)}, \qquad i, j = 1, \dots, n; k = 1,2, \dots .$$

Ймовірність f_{ij} того, що, починаючи зі стану s_i, система коли-небудь потрапить у стан s_j:

$$f_{ij} = \sum_{k=1}^{\infty} f_{ij}^{(k)}, \qquad i, j = 1, \dots, n.$$

Середнє число кроків μ_{ij} до першого досягнення стану s_j при виході зі стану s_i:

$$\mu_{ij} = \sum_{k=1}^{\infty} k f_{ij}^{(k)}, \qquad i,j = 1, \dots, n.$$

Різновидом марковської моделі з дискретним числом станів та безперервним часом є **модель «загибелі і розмноження»**. Назва цього процесу пов'язана з низкою біологічних задач, де він є математичною моделлю зміни чисельності біологічних популяцій. Назва моделі «загибель і розмноження» пов'язана з уявленням, що стрілки праворуч означають перехід до станів, для яких виконується зростання номерів станів («народження»), а стрілки ліворуч – зменшення номерів станів («загибель»).

Модель характерна тим, що її граф станів має вигляд ланцюга (рис. 2.6). Особливість цього графа полягає в тому, що кожен із середніх станів s_1, ..., s_{n-1} пов'язаний прямою і зворотною стрілками з кожним із сусідніх станів – правим і лівим, а крайні стани s_0 і s_n – тільки з одним сусіднім станом.

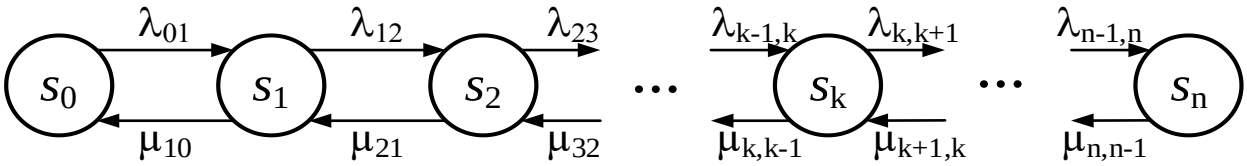


Рисунок 2.6 – Розмічений граф моделі «загибелі і розмноження»

Замість перехідних ймовірностей p_{ij} вводяться інтенсивності переходу λ_{ij} та μ_{ij}. Зазвичай символом λ позначаються інтенсивності переходів у напрямку зростання номерів станів, а символом μ – інтенсивності переходів у напрямку зменшення номерів станів. Інтенсивністю переходу системи зі стану s_i у стан s_j називається границя відношення ймовірності переходу системи за час Δt з стану s_i у стан s_j до довжини Δt:

$$\lambda_{ij} = \lim_{\Delta t \to 0} \frac{p_{ij}\Delta t}{\Delta t},$$

де $p_{ij}\Delta t$ – ймовірність того, що система, яка знаходилася в момент часу t у стані s_i за час Δt перейде з нього в стан s_j.

З формули випливає, що при малому Δt ймовірність переходу $p_{ij}\Delta t$ (з точністю до нескінченно малих вищих порядків) дорівнює $\lambda_{ij}\Delta t$.

Якщо всі інтенсивності переходів λ_{ij} не залежать від t, то марковський процес називається однорідним, а якщо всі інтенсивності представляють собою функції часу, то процес є неоднорідним.

Складати кожного разу рівняння Колмогорова для схеми «загибелі і розмноження» немає необхідності, оскільки структура регулярна (необхідні формули часто наводяться в довідковій літературі). У цій моделі формули для визначення ймовірностей станів, отримані в результаті розв'язання рівнянь Колмогорова, мають вигляд:

$$p_k = \frac{\lambda_{k-1,k}\dots\lambda_{12}\lambda_{01}}{\mu_{k,k-1}\dots\mu_{21}\mu_{10}}p_0;$$

$$p_0 = \left(1 + \frac{\lambda_{01}}{\mu_{10}} + \frac{\lambda_{12}\lambda_{01}}{\mu_{21}\mu_{10}} + \dots + \frac{\lambda_{n-1,n}\dots\lambda_{12}\lambda_{01}}{\mu_{n,n-1}\dots\mu_{21}\mu_{10}}\right)^{-1}.$$

Приклад. Процес загибелі та розмноження представлений графом, наведеним на рис. 2.7. Необхідно знайти граничні ймовірності станів.

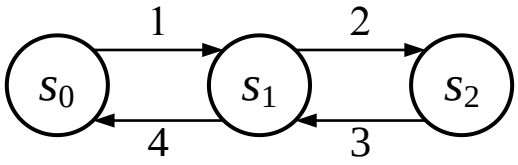


Рисунок 2.7 – Розмічений граф моделі «загибелі і розмноження» для прикладу

$$p_0 = \left(1 + \frac{1}{4} + \frac{2\cdot 1}{3\cdot 4}\right)^{-1} = 0.706;$$

$$p_1 = \frac{\lambda_{01}}{\mu_{10}}p_0 = \frac{1}{4}0.706 = 0.176;$$

$$p_2 = \frac{\lambda_{12}\lambda_{01}}{\mu_{21}\mu_{10}}p_0 = \frac{2\cdot 1}{3\cdot 4}0.706 = 0.118.$$

Тобто, у сталому, стаціонарному режимі в середньому 70,6 % часу система буде перебувати в стані s_0, 17,6 % – у стані s_1, і 11,8 % – у стані s_2.

2.2. Елементи систем масового обслуговування. Коротка характеристика

Більшість систем, з якими людина має справу, є стохастичними системами. Спроба їх математичного опису за допомогою детермінованих моделей приводить до надто грубої оцінки реальності. Під час розв'язання задач аналізу і проектування таких систем необхідно враховувати те, що випадковість є визначальною в процесах, що протікають у них. При цьому, нехтування цією випадковістю призводить до помилок у висновках і практичних рекомендаціях.

У системах масового обслуговування (СМО) розглядаються черги і вирішуються питання по обслуговуванню ряду (потоку) вимог від людей, приладів, подій.

Приклади СМО: черга в магазині (касир розглядається як «прилад», що обслуговує чергу); телефонна станція (розглядається як прилад, що обслуговує потік замовлень на телефонні розмови); оператори ЕОМ (вони розглядаються як прилад, що обслуговує потік інформації від приладів, підприємств та інших операторів); n робітників цеху, які можуть звернутися з вимогами інструменту до комірника («приладу обслуговування») тощо.

Вимоги на виконання робіт поступають у випадкові моменти часу, обслуговуючі пристрої задовольняють вимоги (обробляють їх) за випадковий термін. Кількість вимог є статистично оціненою величиною. СМО мають дві головні ознаки: обслуговуючий пристрій і чергу.

Тому характерним для подібних задач є:

- умови «подвійної» випадковості – випадковий момент часу надходження замовлення на обслуговування та випадкова тривалість часу обслуговування;
- проблема черги.

Системами масового обслуговування [2] називають системи, що складаються з будь-якої кількості каналів обслуговування K_1, K_2, …, K_n, які

призначені для обслуговування будь-якого потоку заявок, що надходять на систему у випадкові моменти часу (рис. 2.8).

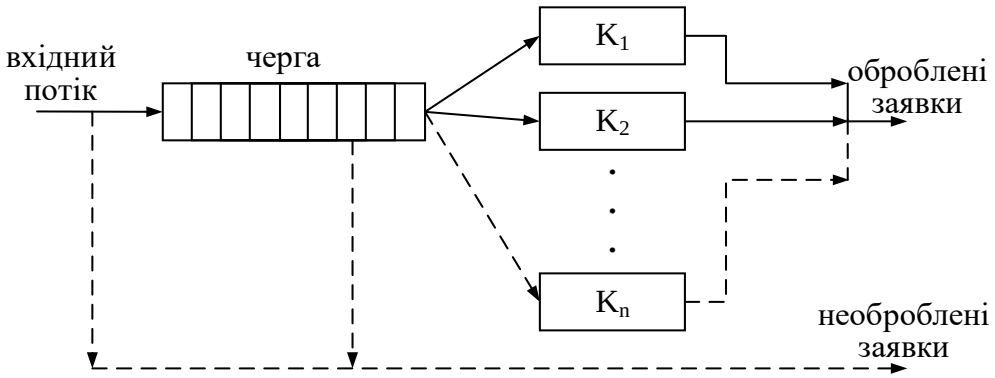


Рисунок 2.8 – Система масового обслуговування

Елементами СМО є [11]:

- вхідний потік заявок на обслуговування;
- прилади (канали) обслуговування;
- черга заявок, що очікують на обслуговування;
- вихідний потік оброблених заявок;
- потік необроблених заявок;
- черга вільних каналів (для багатоканальних СМО).

СМО розрізняються:

1. За конструкцією обслуговуючого пристрою:
   - одноканальна – є один канал обслуговування;
   - багатоканальна. Кожен канал у багатоканальній СМО одночасно може обслуговувати лише одну заявку і, якщо не обговорено спеціально, кожна заявка, що знаходиться під обслуговуванням, обслуговується тільки одним каналом. Багатоканальні СМО можуть складатися з однорідних каналів або різнорідних, що відрізняються тривалістю обслуговування однієї заявки.
2. За алгоритмом обслуговування черги:
   - традиційний алгоритм FIFO – найбільше розповсюджений алгоритм;

- пріоритетне обслуговування – спочатку необхідно вирішити окрему задачу розбиття загального вхідного потоку приладу на класи (пріоритети). Далі заявка розміщується в черзі відповідно заданому пріоритетному класу;
- зважене обслуговування – робиться спроба надати всім класам певний мінімум пропускної здатності і гарантувати деякі вимоги до затримок. Під вагою для кожного класу вважається відсоток від повного значення пропускної здатності, яка надається класу. Алгоритм, що використовується для призначення ваги, називається налаштованою чергою.

3. За дисципліною обслуговування:

- СМО з відмовами, в яких заявка, що надійшла на вхід СМО в момент, коли всі канали зайняті, отримує «відмову» і залишає СМО («зникає»). Щоб ця заявка все ж таки була оброблена, вона повинна знову надійти на вхід СМО і розглядатися при цьому буде як заявка, яка надійшла вперше. Прикладом СМО з відмовами може бути робота АТС: якщо набраний телефонний номер (заявка, що надійшла на вхід) зайнятий, то заявка отримує відмову, і, щоб додзвонитися за цим номером, слід набрати його ще раз (заявка надходить на вхід як нова);
- СМО з очікуванням (необмеженим очікуванням чи чергою). У таких системах заявка, що надійшла в момент зайнятості всіх каналів, стає в чергу і чекає на звільнення каналу, який прийме її до обслуговування. Кожна заявка, що надійшла на вхід, зрештою, буде обслугована. Такі СМО часто зустрічаються в сфері побутового та медичного обслуговування, на підприємствах (наприклад, обслуговування верстатів бригадою наладчиків);

- СМО змішаного типу (з обмеженим очікуванням). Це такі системи, де на перебування заявки в черзі накладаються деякі обмеження.

При дослідженні СМО можуть вирішуватися такі задачі:

- аналіз – визначення характеристик якості СМО з заданою структурою і параметрами для відомого вхідного потоку заявок;
- параметричний синтез – визначення необхідних значень параметрів СМО з фіксованою структурою для забезпечення заданої якості обслуговування для відомого вхідного потоку заявок;
- структурний синтез – визначення необхідної структури СМО з фіксованими параметрами для забезпечення заданої якості обслуговування для відомого вхідного потоку заявок;
- оптимізація – визначення необхідних структури і параметрів СМО для забезпечення мінімальної вартості її функціонування при підтримці заданої якості обслуговування.

Критерії ефективності функціонування СМО залежать від характеру поставленої задачі, та найбільш часто розраховуються такі:

- ймовірність негайного обслуговування заявки;
- ймовірність відмови в обслуговуванні заявки;
- відносна і абсолютна пропускна здатність системи;
- середній відсоток заявок, які отримали відмову в обслуговуванні;
- середній час очікування в черзі;
- середня довжина черги тощо.

При дослідженні СМО намагаються одержати такі характеристики:

- середню довжину черги;
- середній термін обслуговування;
- середній час, за який обслуговуючий пристрій не працює.

Відповідно, для отримання математичної моделі СМО потрібно знати:

- конструкцію СМО;
- математичний опис потоку заявок, які поступають у СМО;
- опис дисципліни черги, способу обслуговування;
- математичний опис обробки заявок.

Основними показниками, що характеризують роботу СМО, є:

$p_k, k = 0, 1, \dots, n$ – ймовірність того, що k з n каналів системи зайнято (p_0 – ймовірність простоювання системи (усі канали вільні); p_n – ймовірність того, що всі канали зайняті);

$p_{\text{обсл}}$ – ймовірність обслуговування;

$p_{\text{відм}}$ – ймовірність відмови;

$\overline{l_{\text{оч}}}$ – середнє число заявок у черзі;

$\overline{t_{\text{оч}}}$ – середній час очікування обслуговування (очікування в черзі);

$p_{\text{ч}}$ – ймовірність утворення черги;

$\overline{t_{\text{с}}}$ – середній час перебування заявки в системі;

λ – інтенсивність надходження заявок у СМО;

μ – інтенсивність обслуговування заявок;

$\rho = \frac{\lambda}{\mu}$ – показник (коефіцієнт) завантаженості системи (трафік) – середня кількість заявок, які приходять у СМО за середній час обслуговування однієї заявки;

A – абсолютна пропускна здатність системи – середнє число заявок, що обслуговується системою за час T;

Q – відносна пропускна здатність системи – середня доля заявок, що обслуговується системою (відношення середнього числа оброблених заявок до середнього числа отриманих за час T);

$\overline{t_{\text{пр}}}$ – середній час простоювання каналу;

$\overline{n_{\text{з}}}$– середнє число зайнятих каналів;

$\overline{n_{\text{в}}}$ – середнє число вільних каналів;

$\overline{t_{\text{обсл}}}$ – середній час обслуговування заявки.

Задача функціонування **одноканальної СМО з відмовами**: на вхід одноканальної системи подається найпростіший потік заявок з інтенсивністю λ. Потік обслуговування – також найпростіший з інтенсивністю μ. У початковий момент часу канал вільний. Якщо заявка поступає в момент часу, коли канал зайнятий, то вона отримує відмову в обслуговуванні і покидає систему. Необхідно знайти граничні ймовірності станів системи та показники її ефективності.

Найпростіший потік (стаціонарний ординарний потік без наслідків – потік Пуассона) з інтенсивністю λ (або μ) у загальному випадку залежить від часу і підпорядковується пуассонівському закону розподілу зі щільністю розподілу $f_1(t) = \lambda e^{-\lambda t}$ та $f_2(t) = \mu e^{-\mu t}$.

Система може знаходитись у двох станах (розмічений граф наведено на рис. 2.9):

s_0 – канал вільний (очікування);

s_1 – канал зайнятий (йде обслуговування заявки).

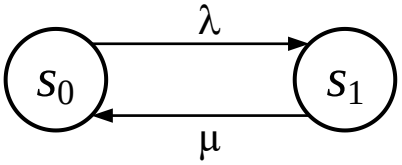


Рисунок 2.9 – Граф станів одноканальної СМО з відмовами

Основні характеристики системи (ймовірність простоювання системи $p_0(t)$ та ймовірність зайнятості каналу обслуговування (відмови) $p_1(t)$, при цьому справедлива рівність $p_0(t) + p_1(t) = 1$) можна визначити, використовуючи рівняння Колмогорова (для початкових умов $p_0(0) = 1$, $p_1(0) = 0$):

$$\begin{cases} p_0' = -\lambda p_0 + \mu p_1 \\ p_1' = -\mu p_1 + \lambda p_0 \end{cases},$$

При $t \to \infty$:

$$p_0 = p_{\text{обсл}} = \frac{\mu}{\lambda + \mu}, \qquad p_1 = p_{\text{відм}} = \frac{\lambda}{\lambda + \mu}.$$

Також, для аналізованої СМО можна визначити відносну Q та абсолютну A пропускну здатність, які розраховуються за такими формулами:

$$Q = \frac{\mu}{\lambda + \mu} = p_0, \qquad A = \lambda Q = \frac{\lambda \cdot \mu}{\lambda + \mu},$$

і очевидно, що $Q = 1 - p_{\text{відм}} = 1 - p_1$.

Середній час обслуговування каналом однієї заявки визначається як:

$$\overline{t_{\text{обсл}}} = \frac{1}{\mu}.$$

Середній час очікування надходження чергової заявки $\overline{t_{\text{пр}}}$, тобто середнє значення інтервалу часу між надходженням двох сусідніх заявок:

$$\overline{t_{\text{пр}}} = \frac{1}{\lambda}.$$

Приклад. До технічної підтримки ІТ-компанії надходять запити користувачів через онлайн-чат із середньою інтенсивністю 90 звернень на годину. Один спеціаліст служби підтримки опрацьовує звернення в середньому за 2 хвилини. Визначити основні показники ефективності СМО для даного випадку, якщо в системі працює лише один оператор (один віртуальний канал зв'язку).

Розв'язання:

$$\lambda = 90 \ {}^{1}/_{\text{год}}, \qquad t_{\text{обсл}} = 2 \text{ хв} = \frac{1}{30} \text{ год}, \qquad \mu = \frac{1}{t_{\text{обсл}}} = 30 \ {}^{1}/_{\text{год}},$$

$$Q = p_0 = \frac{\mu}{\lambda + \mu} = \frac{30}{90 + 30} = 0.25, \qquad p_1 = p_{\text{відм}} = \frac{\lambda}{\lambda + \mu} = \frac{90}{90 + 30} = 0.75,$$

$$A = \lambda Q = 90 \cdot 0.25 = 22.5.$$

Висновок: у середньому за годину буде обслуговано 22,5 заявки з 90, що становить 25 % від всіх, що надходять – СМО погано справляється з потоком заявок.

В **одноканальній СМО з обмеженою чергою** кількість місць n у черзі обмежена. Отже, заявка, що надійшла в момент часу, коли всі місця в черзі зайняті, відхиляється та залишає систему. Класичним прикладом такої системи є, наприклад, кінотеатр із однією квитковою касою, де придбати

квитки на конкретний сеанс можна тільки в такій кількості, скільки місць є в залі.

Граф станів такої СМО наведено на рис. 2.10 і він являє собою схему «загибелі і розмноження».

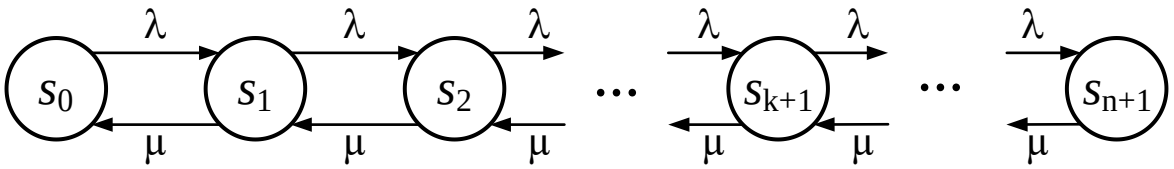


Рисунок 2.10 – Граф станів одноканальної СМО з обмеженою чергою

Можливі стани системи:

s_0 – канал вільний (очікування);

s_1 – канал зайнятий, черги немає;

s_2 – канал зайнятий, у черзі одна заявка;

. . . . . ;

s_{k+1} – канал зайнятий, у черзі k заявок;

. . . . . ;

s_{n+1} – канал зайнятий, всі n місць у черзі зайняті.

Ймовірність відмови:

$$p_{\text{відм}} = p_{n+1} = \rho^{n+1} \cdot p_0,$$

відносна пропускна здатність:

$$Q = 1 - p_{n+1} = 1 - \rho^{n+1} \cdot p_0,$$

абсолютна пропускна здатність:

$$A = Q \cdot \lambda.$$

Інші показники системи залежать від показника завантаженості системи ρ.

СМО працює за схемою «загибелі і розмноження» та має кінцеву кількість станів. З кожного стану за кінцеве число кроків можна перейти до будь-якого іншого стану:

$$p_0 = (1 + \rho + \rho^2 + \cdots + \rho^{n+1})^{-1}.$$

Загальний випадок ($\rho \neq 1$). На основі загального розв'язання для схеми «загибелі і розмноження», граничні ймовірностей станів визначаються таким чином:

$$\begin{cases} p_0 = \dfrac{1-\rho}{1-\rho^{n+2}}, \rho \neq 1 \\ p_k = \rho^k p_0, (k = 1, 2, \dots, n+1) \end{cases}.$$

Ймовірність утворення черги:

$$p_{\text{ч}} = p_2 + \cdots + p_{n+1} = (\rho^2 + \cdots + \rho^{n+1})p_0 = \rho^2 \frac{1-\rho^n}{1-\rho} p_0 = \rho^2 \frac{1-\rho^n}{1-\rho^{n+2}}.$$

Середня кількість заявок $\overline{l_{\text{оч}}}$, що знаходяться в черзі, визначається таким чином:

$$\overline{l_{\text{оч}}} = \rho^2 \frac{1-\rho^n(n-n\rho+1)}{(1-\rho)^2} p_0.$$

Середнє число заявок, які знаходяться на обслуговуванні:

$$\overline{l_{\text{обсл}}} = 0 \cdot p_0 + 1 \cdot (1-p_0) = 1 - p_0 ,$$

середнє число заявок у системі:

$$\overline{l_{\text{СМО}}} = \overline{l_{\text{обсл}}} + \overline{l_{\text{оч}}}.$$

Середній час очікування обслуговування

$$\overline{t_{\text{оч}}} = \frac{\overline{l_{\text{оч}}}}{\lambda},$$

середній час перебування заявки в системі:

$$\overline{t_{\text{с}}} = \frac{\overline{l_{\text{СМО}}}}{\lambda}.$$

Окремий випадок ($\rho = 1$). Ймовірність простоювання системи:

$$p_0 = (1 + \rho + \rho^2 + \cdots + \rho^{n+1})^{-1} = (1 + 1 + 1 + \cdots + 1)^{-1},$$

тобто

$$p_0 = \frac{1}{n+2},$$

а ймовірності всіх інших без винятку станів системи рівні одна одній та рівні ймовірності вільного стану системи p_0:

$$p_0 = p_1 = p_2 = \cdots = p_{n+1} = \frac{1}{n+2}.$$

Середня кількість заявок $\overline{l_{\text{оч}}}$, що знаходяться в черзі:

$$\overline{l_{\text{оч}}} = \frac{n(n+1)}{2} p_0.$$

Ймовірність утворення черги:

$$p_{\text{ч}} = p_2 + \cdots + p_{n+1} = \frac{n}{n+2}.$$

Середнє число заявок, які знаходяться на обслуговуванні:

$$\overline{l_{\text{обсл}}} = 0 \cdot p_0 + 1 \cdot (1 - p_0) = 1 - p_0 = \frac{n+1}{n+2},$$

середня кількість заявок у черзі:

$$\overline{l_{\text{оч}}} = \sum_{i=1}^{n} i p_{i+1} = \frac{1}{n+2} \sum_{i=1}^{n} i = \frac{n(n+1)}{2(n+2)},$$

середнє число заявок у системі:

$$\overline{l_{\text{СМО}}} = \overline{l_{\text{обсл}}} + \overline{l_{\text{оч}}} = \frac{n+1}{2}.$$

Середній час очікування обслуговування:

$$\overline{t_{\text{оч}}} = \frac{\overline{l_{\text{оч}}}}{\lambda} = \frac{n(n+1)}{\lambda \cdot 2(n+2)},$$

середній час обслуговування заявки:

$$\overline{t_{\text{обсл}}} = \frac{\overline{l_{\text{обсл}}}}{\lambda} = \frac{1}{\mu},$$

середній час перебування заявки в системі (для розглянутого окремого випадку $\lambda = \mu$):

$$\overline{t_{\text{с}}} = \frac{\overline{l_{\text{СМО}}}}{\lambda} = \frac{n+1}{2\mu}.$$

Приклад. На сервері встановлено одне обчислювальне ядро, яке обробляє великі пакетні завдання. Інтенсивність надходження завдань становить $\lambda = 0.4$ завдань/хвилину, а середній час обробки одного завдання – $\overline{t_{\text{об}}} = 2$ хвилини. Якщо в черзі на обробку вже чекає більше двох завдань,

нове завдання відхиляється та не потрапляє до системи. Визначити показники ефективності роботи обчислювального ядра.

Розв'язання: маємо одноканальну СМО з обмеженою чергою ($n = 3$).

$$\mu = \frac{1}{\overline{t_{\text{об}}}} = \frac{1}{2} = 0.5\ {}^{1}/_{\text{хв}}, \quad \rho = \frac{\lambda}{\mu} = \frac{0.4}{0.5} = 0.8.$$

$$p_0 = (1 + \rho + \rho^2 + \rho^3 + \rho^4)^{-1} = \frac{1-\rho}{1-\rho^{n+2}} = \frac{1-0.8}{1-0.8^5} \approx 0.297,$$

$$p_{\text{відм}} = \rho^4 \cdot p_0 = 0.8^4 \cdot 0.297 \approx 0.122,$$

$$p_{\text{ч}} = \rho^2 \frac{1-\rho^n}{1-\rho} p_0 = 0.8^2 \frac{1-0.8^{n3}}{1-0.8} 0.297 \approx 0.464,$$

$$Q = 1 - p_{n+1} = 1 - p_{\text{відм}} = 1 - 0.122 = 0.878,$$

$$A = Q \cdot \lambda = 0.878 \cdot 0.4 \approx 0.351,$$

$$\overline{l_{\text{оч}}} = \rho^2 \frac{1-\rho^n(n-n\rho+1)}{(1-\rho)^2} p_0 = \rho^2 \frac{1-\rho^n(n-n\rho+1)}{(1-\rho)(1-\rho^{n+2})} =$$

$$= 0.8^2 \frac{1-0.8^3(3-3\cdot 0.8+1)}{(1-0.8)(1-0.8^5)} \approx 0.861,$$

$$\overline{t_{\text{оч}}} = \frac{\overline{l_{\text{оч}}}}{\lambda} = \frac{0.861}{0.4} \approx 2.153 \text{ хв},$$

$$\overline{l_{\text{обсл}}} = 1 - p_0 = 1 - 0.297 = 0.703, \qquad \overline{t_{\text{обсл}}} = \frac{\overline{l_{\text{обсл}}}}{\lambda} = \frac{0.703}{0.4} \approx 1.758,$$

$$\overline{l_{\text{СМО}}} = \overline{l_{\text{обсл}}} + \overline{l_{\text{оч}}} = 0.861 + 0.703 = 1.564, \qquad \overline{t_{\text{с}}} = \frac{\overline{l_{\text{СМО}}}}{\lambda} = \frac{1.564}{0.4} = 3.91 \text{ хв}.$$

В **одноканальній СМО без обмежень**, на відміну від одноканальної СМО з очікуванням, при обмеженому числі n місць у черзі, довжина черги (теоретично) може збільшуватись до нескінченності. Граф такої системи наведено на рис. 2.11.

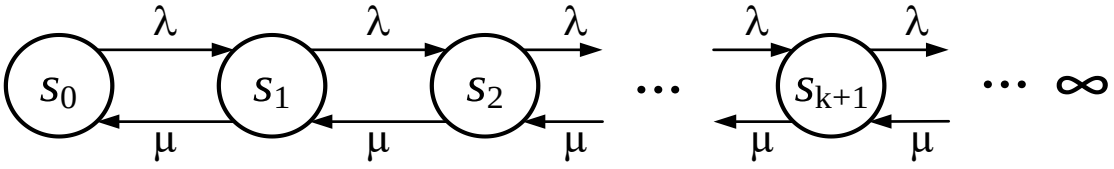


Рисунок 2.11 – Граф станів одноканальної СМО без обмежень

Можливі стани системи будуть такими:

s_0 – канал вільний (очікування);

s_1 – канал зайнятий, черги немає;

s_2 – канал зайнятий, у черзі одна заявка;

. . . . . ;

s_{k+1} – канал зайнятий, у черзі k заявок;

. . . . . ;

Всі характеристики такої СМО можна отримати з формул для одноканальної СМО з обмеженою чергою, вважаючи $n \to \infty$. При цьому необхідно відрізняти два істотно різних випадки: при $\rho \geq 1$ і при $\rho < 1$. У першому випадку при $t \to \infty$ черга необмежено збільшується, тобто цей випадок не має практичної цінності.

Оскільки відсутнє обмеження на довжину черги, то будь-яка заявка може бути обслугована, тобто відносна пропускна здатність:

$$Q = p_{\text{обсл}} = 1.$$

Абсолютна пропускна здатність:

$$A = \lambda \cdot Q = \lambda.$$

Ймовірність відмови в такій системі неможлива, тому приймається як $p_{\text{відм}} = 0$, інші ймовірності:

$$\begin{cases} p_0 = (1 + \rho + \rho^2 + \cdots + \rho^{n+1})^{-1} \\ p_k = \rho^k p_0, (k = 1, 2, \dots) \end{cases}.$$

Сума в дужках для p_0 є геометричною прогресією, яка збігається тільки тоді, коли $\rho < 1$, тобто для СМО з очікуванням існує граничний встановлений режим, і $p_0 = (1 - \rho)$ при $n \to \infty$.

Наявність черги означає, що в черзі є як мінімум одна заявка, відповідно СМО знаходиться в будь-якому стані, крім s_0 і s_1, отже, ймовірність того, що почне утворюватися черга:

$$p_{\text{ч}} = 1 - p_0 - p_1 = 1 - (1 - \rho) - \rho(1 - \rho) = \rho^2.$$

Для граничного встановленого режиму визначається також середня кількість заявок $\overline{l_{\text{оч}}}$, що очікують у черзі:

$$\overline{l_{\text{оч}}} = \frac{\rho^2}{1-\rho},$$

і середня кількість заявок у системі:

$$\overline{l_{\text{СМО}}} = \overline{l_{\text{оч}}} + \rho = \frac{\rho}{1-\rho}.$$

Середнє число заявок, які знаходяться на обслуговуванні:

$$\overline{l_{\text{обсл}}} = \rho \cdot Q = \rho.$$

Середній час обслуговування заявки:

$$\overline{t_{\text{обсл}}} = \frac{\rho}{\lambda} = \frac{1}{\mu}.$$

Середній час очікування обслуговування в черзі:

$$\overline{t_{\text{оч}}} = \frac{\rho}{\mu(1-\rho)} = \frac{\rho^2}{\lambda(1-\rho)} = \frac{\overline{l_{\text{оч}}}}{\lambda},$$

середній час перебування заявки в системі:

$$\overline{t_{\text{с}}} = \overline{t_{\text{оч}}} + \overline{t_{\text{обсл}}} = \frac{\rho}{\mu(1-\rho)} + \frac{1}{\mu} = \frac{1}{\mu(1-\rho)} = \frac{\rho}{\lambda(1-\rho)} = \frac{\overline{l_{\text{СМО}}}}{\lambda}.$$

Приклад. На сервері обробки даних доступний один обчислювальний вузол (ресурс) для обробки пакетів завдань. Інтенсивність надходження пакетів завдань становить 0.4 пакета за хвилину, а середній час обробки одного пакета – 2 хвилини. Припускаючи, що черга завдань може мати необмежену довжину, потрібно: визначити основні показники ефективності обчислювального вузла (СМО), знайти ймовірність того, що в черзі на обробку очікують не більше ніж 2 пакети завдань.

Розв'язання:

$$\lambda = 0.4\ {}^{1}/_{\text{хв}}, \qquad t_{\text{обсл}} = 2\ \text{хв}, \qquad \mu = \frac{1}{\overline{t_{\text{обсл}}}} = \frac{1}{2} = 0.5\ {}^{1}/_{\text{хв}},$$

$$\rho = \lambda \cdot t_{\text{обсл}} = 0.4 \cdot 2 = 0.8 < 1.$$

СМО справляється з потоком заявок. Стан, коли в черзі є дві заявки – s_3, відповідно необхідно розрахувати ймовірності p_0, p_1, p_2, p_3 та визначити сумарну ймовірність перебування СМО в станах $s_0 - s_3$:

$p_0 = (1-\rho) = 1 - 0.8 = 0.2, \qquad p_1 = \rho p_0 = 0.8 \cdot 0.2 = 0.16,$

$p_2 = \rho^2 p_0 = 0.8^2 \cdot 0.2 = 0.128, \qquad p_3 = \rho^3 p_0 = 0.8^3 \cdot 0.2 = 0.1024.$

$p\left(\overline{l_{\text{СМО}}} \le 3\right) = p_0 + p_1 + p_2 + p_3 \approx 0.59.$

$\overline{l_{\text{СМО}}} = \frac{\rho}{1-\rho} = \frac{0.8}{1-0.8} = 4, \qquad t_{\text{с}} = \frac{\overline{l_{\text{СМО}}}}{\lambda} = \frac{4}{0.4} = 10 \text{ хв},$

$\overline{l_{\text{оч}}} = \frac{\rho^2}{1-\rho} = \frac{0.8^2}{1-0.8} = 3.2, \qquad \overline{t_{\text{оч}}} = \frac{\overline{l_{\text{оч}}}}{\lambda} = \frac{3.2}{0.4} = 8 \text{ хв},$

$\overline{l_{\text{обсл}}} = \rho = 0.8, \qquad \overline{t_{\text{обсл}}} = \frac{\rho}{\lambda} = \frac{0.8}{0.4} = 2 \text{ хв} = t_{\text{обсл}}.$

$A = \lambda = 0.4, \qquad Q = 1, \qquad p_{\text{ч}} = \rho^2 = 0.8^2 = 0.64.$

Ймовірність того, що обчислювальний вузол зайнятий, а в черзі не більше двох пакетів завдань:

$p\left(\overline{l_{\text{оч}}} \le 2\right) = p_1 + p_2 + p_3 \approx 0.39.$

Висновок: ефективність роботи серверу обробки даних – невисока. Для її підвищення необхідно зменшення середнього часу обробки пакетів або збільшення числа обчислювальних вузлів.

У переважній більшості випадків на практиці СМО є багатоканальними, і отже, моделі з $n > 1$ каналами обслуговування викликають безсумнівну зацікавленість.

Багатоканальна СМО з відмовами містить n каналів, інтенсивність потоку вхідного заявок дорівнює λ, а інтенсивність обслуговування заявки кожним каналом дорівнює μ. Система працює без черги, тобто з відмовами в обслуговуванні заявок, що прийшли в той момент, коли всі канали системи вже зайняті обслуговуванням заявок. Розмічений граф станів системи наведено на рис. 2.12.

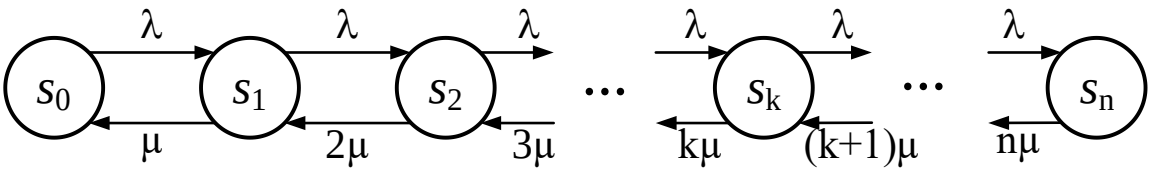


Рисунок 2.12 – Граф станів багатоканальної СМО з відмовами

Можливі стани системи:

s_0 – система вільна (простоює);

s_1 – один канал зайнятий;

s_2 – два канали зайнято;

. . . . . ;

s_k –зайнято k каналів.

. . . . . ;

s_n – всі канали системи зайняті обслуговуванням.

Випадковий процес, що протікає в СМО, є окремим випадком процесу «загибелі і розмноження» і описується системою диференціальних рівнянь Ерланга, які дозволяють отримати вирази для граничних ймовірностей стану системи – формулами Ерланга:

$$\begin{cases} p_0 = \left(\sum_{i=0}^{n} \frac{\rho^i}{i!} \right)^{-1} \\ p_k = \frac{\rho^k}{k!} p_0, (k = 1, 2, \dots, n) \end{cases}.$$

Ймовірність відмови в обслуговуванні заявки $p_{\text{відм}}$ дорівнює ймовірності того, що всі канали зайняті, тобто система перебуває в стані s_n:

$$p_{\text{відм}} = p_n = \frac{\rho^n}{n!} p_0.$$

Відносна пропускна здатність:

$$Q = p_{\text{обсл}} = 1 - p_{\text{відм}} = 1 - \frac{\rho^n}{n!} p_0.$$

Абсолютна пропускна здатність:

$$A = \lambda \cdot Q = \lambda \cdot p_{\text{обсл}} = \lambda \left(1 - \frac{\rho^n}{n!} p_0 \right).$$

Середню кількість зайнятих каналів $\overline{n_з}$ можна визначити, виходячи з того, що кожний зайнятий канал в одиницю часу обслуговує в середньому μ заявок:

$$\overline{n_з} = \frac{A}{\mu} = \frac{\lambda(1-p_n)}{\mu} = \rho(1-p_n) = \rho\left(1-\frac{\rho^n}{n!}p_0\right) = \sum_{i=0}^{n} ip_i.$$

Середня кількість вільних каналів:

$$\overline{n_в} = n - \overline{n_з}.$$

Середній час перебування заявки в системі визначається формулою Літтла

$$\overline{t_с} = \frac{\overline{n_з}}{\lambda}.$$

Приклад. Відомо, що звернення до служби онлайн-підтримки великого IT-сервісу надходять з інтенсивністю 90 запитів на годину, а середній час обробки одного звернення оператором становить 2 хвилини. Визначити оптимальну кількість операторів технічної підтримки (віртуальних каналів обслуговування), якщо за умову оптимальності вважається обробка не менше ніж 90 звернень зі 100. Оцінити економічну ефективність системи підтримки, якщо: кожне успішно оброблене звернення приносить 10 грошових одиниць доходу; утримання одного оператора коштує 50 грошових одиниць на годину; тривалість робочого дня – 8 годин.

Розв'язання:

$$\mu = \frac{1}{\overline{t_{обсл}}} = \frac{60}{2} = 30\ {}^{1}/_{год}, \qquad \rho = \frac{\lambda}{\mu} = \frac{90}{30} = 3,$$

тобто за час середньої за тривалістю телефонної розмови надходить у середньому 3 звернення. Рішення задачі можна знайти поступовим збільшенням кількості каналів:

$$n = 2: p_0 = (1 + 3 + 4.5)^{-1} \approx 0.12,$$

$$Q = 1 - p_{відм} = 1 - 4.5 \cdot 0.12 = 0.45,$$

$$A = \lambda \cdot Q = 90 \cdot 0.46 = 41.4.$$

$$n = 3: p_0 = (1 + 3 + 4.5 + 4.5)^{-1} \approx 0.077,$$

$$Q = 1 - 4.5 \cdot 0.077 \approx 0.65,$$

$$A = \lambda \cdot Q = 90 \cdot 0.65 = 58.5.$$

$n = 4$: $p_0 = (1 + 3 + 4.5 + 3.375)^{-1} \approx 0.061,$

$$Q = 1 - 3.375 \cdot 0.061 \approx 0.79,$$

$$A = \lambda \cdot Q = 90 \cdot 0.79 = 71.5.$$

$n = 5$: $p_0 = (1 + 3 + 4.5 + 3.375 + 2.025)^{-1} \approx 0.054,$

$$Q = 1 - 2.025 \cdot 0.054 \approx 0.89,$$

$$A = \lambda \cdot Q = 90 \cdot 0.89 = 80.1.$$

$n = 6$: $p_0 = (1 + 3 + 4.5 + 3.375 + 2.025 + 1.0125)^{-1} \approx 0.0515,$

$$Q = 1 - 1.0125 \cdot 0.0515 \approx 0.95,$$

$$A = \lambda \cdot Q = 90 \cdot 0.95 = 85.5.$$

Таким чином, при значенні $n = 6$ було досягнуто необхідну умову оптимальності ($Q \geq 0.9$).

Економічна ефективність СМО розраховується за формулою:

$$П = С \cdot А - В \cdot n,$$

де П – чистий прибуток, С – прибуток від обслуговування однієї заявки, В – витрати на утримання одного каналу (оператора) в одиницю часу, n – кількість каналів:

$П = 10 \cdot 85.5 - 50 \cdot 6 = 555$ грош. од./год.

Висновок: оптимальна кількість операторів дорівнює шести. При цьому отримують обслуговування 95 % заявок, що надходять. Чистий дохід за робочий день 6 операторів становить $555 \cdot 8 = 4440$ грош. од.

Багатоканальна СМО з обмеженням довжини черги – на вхід СМО, що має n каналів обслуговування, надходить пуассонівський потік заявок з інтенсивністю λ. Інтенсивність обслуговування заявки кожним каналом дорівнює μ, а максимальна кількість місць у черзі дорівнює m. На рис. 2.13 наведено граф станів такої системи.

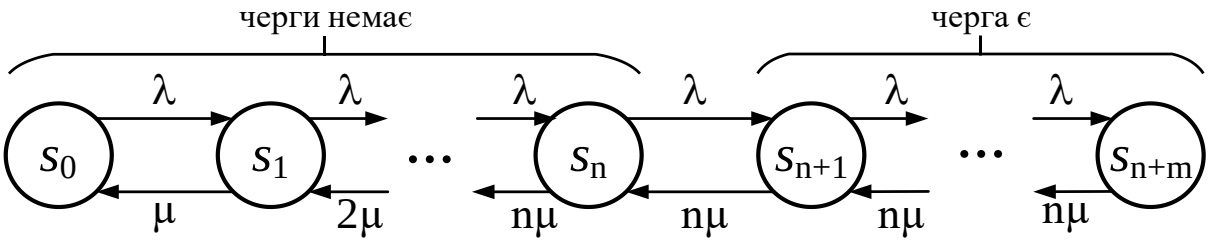


Рисунок 2.13 – Граф станів багатоканальної СМО з обмеженою довжиною черги

Можливі стани системи:

s_0 – система вільна (простоює);

s_1 – один канал зайнятий;

s_2 – два канали зайнято;

. . . . . ;

s_n – всі канали системи зайняті обслуговуванням.

s_{n+1} – всі канали зайняті обслуговуванням, у черзі одна заявка;

s_{n+2} – всі канали зайняті обслуговуванням, у черзі дві заявки;

. . . . . ;

s_{n+m} – всі канали зайняті, всі m місць у черзі зайняті.

Усі показники СМО з обмеженням на довжину черги залежать від показника завантаження системи та кількості каналів обслуговування. У цих системах, так само як і в одноканальних системах з обмеженням, виділяють загальний та окремий випадки.

<u>Загальний випадок</u> (при $\rho \neq n$).

Граф станів СМО з обмеженням на довжину черги також являє собою схему «загибелі і розмноження», тому можна отримати:

$$\begin{cases} p_0 = \left(\sum_{i=0}^{n} \frac{\rho^i}{i!} + \frac{\rho^{n+1} \cdot \left(1 - \left(\frac{\rho}{n}\right)^m\right)}{n! \cdot (n - \rho)} \right)^{-1} \\ p_k = \frac{\rho^k}{k!} p_0 , \qquad k = 1,2, \dots, n \\ p_{n+l} = \frac{\rho^{n+l}}{n^l \cdot n!} p_0 , \qquad l = 1,2, \dots, m \end{cases} .$$

Утворення черги відбувається, коли в момент надходження до СМО чергової заявки всі n каналів зайняті, тобто коли в системі буде або n, або $n+1$, …, або $(n+m-1)$ заявок. Оскільки ці події несумісні, то ймовірність утворення черги $p_{\text{ч}}$ дорівнює сумі відповідних ймовірностей p_n, p_{n+1}, ..., p_{n+m-1}:

$$p_{\text{ч}} = \sum_{i=0}^{m-1} p_{n+i} = \frac{\rho^n}{n!} \cdot \frac{1-\left(\frac{\rho}{n}\right)^m}{1-\frac{\rho}{n}} p_0.$$

Відмова в обслуговуванні заявки відбувається, коли всі m місць у черзі зайняті:

$$p_{\text{відм}} = p_{n+m} = \frac{\rho^{n+m}}{n^m n!} p_0.$$

Відносна пропускна здатність:

$$Q = p_{\text{обсл}} = 1 - p_{\text{відм}} = 1 - \frac{\rho^{n+m}}{n^m n!} p_0.$$

Абсолютна пропускна здатність:

$$A = \lambda \cdot Q = \lambda \cdot p_{\text{обсл}} = \lambda \cdot \left(1 - \frac{\rho^{n+m}}{n^m n!} p_0\right).$$

Середня кількість заявок в черзі (довжина черги):

$$\overline{l_{\text{оч}}} = \sum_{i=1}^{m} i p_{n+i} = \frac{\rho^{n+1}}{n \cdot n!} \cdot \frac{1-\left(\frac{\rho}{n}\right)^m \cdot \left(m+1-m\frac{\rho}{n}\right)}{\left(1-\frac{\rho}{n}\right)^2} p_0.$$

Середнє число зайнятих каналів (середня кількість заявок під обслуговуванням):

$$\overline{n_{\text{з}}} = \frac{A}{\mu} = \rho\left(1 - \frac{\rho^{n+m}}{n^m n!} p_0\right),$$

середнє число вільних каналів:

$$\overline{n_{\text{в}}} = n - \overline{n_{\text{з}}}$$

і середня кількість заявок у системі:

$$\overline{l_{\text{СМО}}} = \overline{l_{\text{оч}}} + \overline{n_{\text{з}}}.$$

Середній час очікування обслуговування в черзі:

$$\overline{t_{\text{оч}}} = \frac{\overline{l_{\text{оч}}}}{\lambda}.$$

Середній час обслуговування заявки:

$$\overline{t_{\text{обсл}}} = \frac{\overline{n_{\text{з}}}}{\lambda}.$$

Середній час перебування заявки в системі:

$$\overline{t_{\text{с}}} = \frac{\overline{l_{\text{СМО}}}}{\lambda}.$$

Окремий випадок ($\rho = n$). Ймовірність простоювання системи та інших станів:

$$\begin{cases} p_0 = \left(\sum_{i=0}^{n} \frac{\rho^i}{i!} + \frac{\rho^n \cdot m}{n!} \right)^{-1} \\ p_k = \frac{\rho^k}{k!} p_0 \,, \qquad k = 1,2, \dots, n \\ p_{n+1} = p_{n+2} = \cdots = p_{n+m} = \frac{\rho^n}{n!} p_0 \end{cases}.$$

Відмова в обслуговуванні заявки відбувається, коли всі m місць у черзі зайняті:

$$p_{\text{відм}} = p_{n+m} = \frac{\rho^n}{n!} p_0.$$

Ймовірність утворення черги:

$$p_{\text{ч}} = m p_n = m \cdot \frac{\rho^n}{n!} p_0.$$

Відносна пропускна здатність:

$$Q = 1 - p_{\text{відм}} = 1 - \frac{\rho^n}{n!} p_0.$$

Абсолютна пропускна здатність:

$$A = \lambda \cdot Q = \lambda \cdot \left(1 - \frac{\rho^n}{n!} p_0\right).$$

Середня кількість заявок під обслуговуванням (середнє число зайнятих каналів):

$$\overline{n_{\text{з}}} = \frac{A}{\mu} = \rho Q = \rho \left(1 - \frac{\rho^n}{n!} p_0\right),$$

Середня кількість заявок у черзі для окремого випадку:

$$\overline{l_{\text{оч}}} = \sum_{i=1}^{m} i p_{n+i} = \sum_{i=1}^{m} i p_n = \frac{\rho^n}{n!} \cdot p_0 \sum_{i=1}^{m} i = \frac{m(m+1)}{2} \cdot \frac{\rho^n}{n!} p_0.$$

Середня кількість заявок у системі:

$$\overline{l_{\text{СМО}}} = \overline{l_{\text{оч}}} + \overline{n_{\text{з}}}.$$

Середній час очікування обслуговування в черзі $\overline{t_{\text{оч}}}$:

$$\overline{t_{\text{оч}}} = \frac{\overline{l_{\text{оч}}}}{\lambda}.$$

Середній час обслуговування заявки:

$$\overline{t_{\text{обсл}}} = \frac{\overline{n_{\text{з}}}}{\lambda}.$$

Середній час перебування заявки в системі $\overline{t_{\text{с}}}$

$$\overline{t_{\text{с}}} = \frac{\overline{l_{\text{СМО}}}}{\lambda}.$$

Приклад. На вузол обробки даних надходять запити на обслуговування в середньому з інтенсивністю $\lambda = 3$ запити/хв. У системі працює дві незалежні обчислювальні служби (двоядерний сервер), які можуть одночасно обробляти два запити. Середній час обробки одного запиту – 40 сек. Для очікування обробки передбачено два буферні місця (черга обмежена двома запитами). Якщо при надходженні нового запиту всі служби зайняті, і черга вже заповнена – запит втрачається. Вартість обробки одного запиту $C = 1.4$ грн. Провести аналіз ефективності роботи СМО. Оцінити втрату прибутку через відмови в обслуговуванні.

Розв'язання:

$$n = 2, \qquad m = 2, \qquad \lambda = 3 \text{ запити/хв}, \qquad \overline{t_{\text{обсл}}} = 40 \text{ сек} = \frac{2}{3} \text{ хв},$$

$$\mu = \frac{3}{2} \text{ 1/хв}, \qquad \rho = 2, \qquad \frac{\rho}{n} = 1.$$

Маємо окремий випадок ($\rho = n$).

$$p_0 = \left(\sum_{i=0}^{n} \frac{\rho^i}{i!} + \frac{\rho^n \cdot m}{n!}\right)^{-1} = \left(1 + \rho + \frac{\rho^2}{2!} + \frac{\rho^2 \cdot m}{2!}\right)^{-1} =$$

$$= \left(1 + 2 + \frac{4}{2} + \frac{4 \cdot 2}{2}\right)^{-1} = (1 + 2 + 2 + 4)^{-1} = \frac{1}{9}.$$

$$p_1 = \frac{\rho}{1!} p_0 = \frac{2}{1} \cdot \frac{1}{9} \approx 0.222, \qquad p_2 = p_3 = p_4 = \frac{\rho^2}{2!} p_0 = \frac{4}{2} \cdot \frac{1}{9} \approx 0.222,$$

$$p_{\text{відм}} = p_{n+m} = p_4 \approx 0.222, \qquad p_{\text{ч}} = m \cdot \frac{\rho^n}{n!} p_0 = 2 \cdot \frac{2^2}{2!} \cdot \frac{1}{9} \approx 0.444,$$

$$Q = 1 - p_{\text{відм}} = 1 - 0.222 \approx 0.778, \qquad A = \lambda \cdot Q = 3 \cdot 0.778 \approx 2.333,$$

$$\overline{l_{\text{оч}}} = \frac{m(m+1)}{2} \cdot \frac{\rho^n}{n!} p_0 = \frac{2(2+1)}{2} \cdot \frac{2^2}{2!} \cdot \frac{1}{9} \approx 0.667,$$

$$\overline{t_{\text{оч}}} = \frac{\overline{l_{\text{оч}}}}{\lambda} = \frac{0.667}{3} \approx 0.222 \text{ хв}, \qquad \overline{n_{\text{з}}} = \rho Q = 2 \cdot 0.778 \approx 1.556,$$

$$\overline{t_{\text{обсл}}} = \frac{\overline{n_{\text{з}}}}{\lambda} = \frac{1.556}{3} \approx 0.63 \text{ хв}, \qquad \overline{l_{\text{СМО}}} = \overline{l_{\text{оч}}} + \overline{n_{\text{з}}} \approx 2.222,$$

$$\overline{t_{\text{с}}} = \frac{\overline{l_{\text{СМО}}}}{\lambda} = \frac{2.222}{3} \approx 0.741.$$

Кількість втрачених запитів

$$k_{\text{втр}} = \lambda p_{\text{відм}} = \lambda - A = 3 - \frac{7}{3} = \frac{2}{3}.$$

Втрата прибутку за годину роботи:

$$S = C \cdot k_{\text{втр}} \cdot t = 1.4 \cdot \frac{2}{3} \cdot 60 = 56 \text{ грн.}$$

Багатоканальна СМО з необмеженою чергою – у таких системах теоретично передбачається, що черга зростатиме нескінченно, отже, ймовірність відмови в обслуговуванні дорівнюватиме нулю. Граф станів наведено на рис. 2.14.

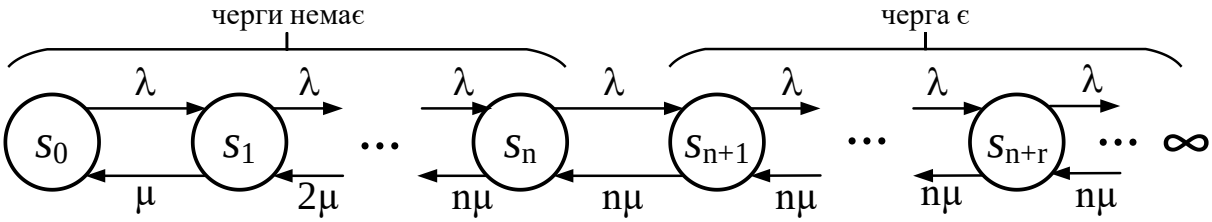


Рисунок 2.14 – Граф станів багатоканальної СМО з необмеженою чергою

Можливі стани системи:

s_0 – система вільна (простоює);

s_1 – один канал зайнятий;

s_2 – два канали зайнято;

. . . . . ;

s_n – всі канали системи зайняті обслуговуванням.

s_{n+1} – всі канали зайняті обслуговуванням, у черзі одна заявка;

s_{n+2} – всі канали зайняті обслуговуванням, у черзі дві заявки;

. . . . . ;

$s_{n+\infty}$ – всі канали зайняті, у черзі нескінченна кількість заявок.

Формули для ймовірностей можна отримати з формул для багатоканальної СМО з обмеженою чергою при $m \to \infty$. Слід пам'ятати, що при $\rho/n \geq 1$ ймовірності $p_0 = p_1 = \cdots = p_n = 0$, тобто черга необмежено зростає. Цей випадок практичного інтересу не становить і розглядається лише випадок $\rho/n < 1$:

$$\begin{cases} p_0 = \left(\sum_{i=0}^{n} \frac{\rho^i}{i!} + \frac{\rho^{n+1}}{n! \cdot (n-\rho)} \right)^{-1} \\ p_k = \frac{\rho^k}{k!} p_0 , \qquad k = 1,2, \dots, n \\ p_{n+l} = \frac{\rho^{n+l}}{n^l \cdot n!} p_0 , \qquad l = 1,2, \dots \end{cases} .$$

Вираз для ймовірності утворення черги заявок:

$$p_{\text{ч}} = \frac{\rho^{n+1}}{n!} \cdot \frac{1}{n-\rho} p_0 .$$

Середня кількість заявок, що перебувають у черзі:

$$\overline{l_{\text{оч}}} = \sum_{i=1}^{\infty} i p_{n+i} = \frac{\rho^{n+1}}{n \cdot n!} \cdot \frac{p_0}{\left(1 - \frac{\rho}{n}\right)^2} = \frac{n\rho^{n+1}}{n!\,(n-\rho)^2} p_0$$

або

$$\overline{l_{\text{оч}}} = \frac{n}{n-\rho} p_{\text{ч}}.$$

Оскільки черга не обмежена, то ймовірність відмови в обслуговуванні заявки ротка дорівнює нулю:

$$p_{\text{відм}} = 0,$$

відносна пропускна здатність Q дорівнює одиниці:

$$Q = p_{\text{обсл}} = 1 - p_{\text{відм}} = 1,$$

а абсолютна пропускна здатність A дорівнює

$$A = \lambda \cdot Q = \lambda.$$

Інтенсивність вихідного потоку обслужених заявок дорівнює середній кількості обслуговуваних заявок в одиницю часу і дорівнює абсолютній пропускній здатності системи A.

Середній час очікування заявки в черзі:

$$\overline{t_{\text{оч}}} = \frac{\overline{l_{\text{оч}}}}{\lambda}.$$

Середня кількість зайнятих каналів:

$$\overline{n_{\text{з}}} = \frac{A}{\mu} = \frac{\lambda}{\mu} = \rho.$$

Середня кількість вільних каналів:

$$\overline{n_{\text{в}}} = n - \overline{n_{\text{з}}} = n - \rho.$$

Середня кількість заявок, що знаходяться в СМО:

$$\overline{l_{\text{СМО}}} = \overline{l_{\text{оч}}} + \overline{n_{\text{з}}} = \overline{l_{\text{оч}}} + \rho.$$

Середній час перебування заявки в системі:

$$\overline{t_{\text{с}}} = \frac{\overline{l_{\text{СМО}}}}{\lambda}.$$

Приклад. На хмарний сервер надходить потік запитів від користувачів з інтенсивністю $\lambda = 81$ запит/хв. Середній час обробки одного запиту одним обчислювальним ядром становить $\overline{t_{\text{обсл}}} = 2$ сек. Потрібно визначити мінімальну кількість обчислювальних ядер n_{min}, за якої черга запитів не зростатиме безмежно, а також знайти відповідні характеристики обслуговування при цьому значенні n_{min} та знайти ймовірність того, що в черзі очікують не більше трьох запитів.

Розв'язання:

$$\lambda = 81 \text{ запит}/_{\text{хв}} = \frac{81}{60} = 1.35 \text{ запит}/_{\text{сек}},$$

$$\rho = \frac{\lambda}{\mu} = \lambda\overline{t_{\text{обсл}}} = 1.35 \cdot 2 = 2.7.$$

Для того, щоб черга не зростала до нескінченності, необхідно, щоб $\rho/n < 1$, тобто $n > \rho$. Отже, $n_{min} = 3$.

Розрахунок характеристик обслуговування.

$$p_0 = \left(1 + \rho + \frac{\rho^2}{2!} + \frac{\rho^3}{3!} + \frac{\rho^4}{3! \cdot (3 - 2.7)}\right)^{-1} =$$

$$= \left(1 + 2.7 + \frac{2.7^2}{2} + \frac{2.7^3}{6} + \frac{2.7^4}{6 \cdot 0.3}\right)^{-1} =$$

$$= 0.024907 \approx 0.025.$$

Ймовірність виникнення черги заявок:

$$p_{\text{ч}} = \frac{\rho^{n+1}}{n!} \cdot \frac{1}{n - \rho} p_0 = \frac{2.7^4}{3!} \cdot \frac{1}{3 - 2.7} 0.025 \approx 0.738.$$

Середня кількість заявок, що обслуговуються:

$$\overline{n_{\text{з}}} = \rho = 2.7.$$

Середня довжина черги:

$$\overline{l_{\text{оч}}} = \frac{n}{n - \rho} p_{\text{ч}} = \frac{3}{3 - 2.7} 0.738 \approx 7.38.$$

Середній час очікування заявки в черзі:

$$\overline{t_{\text{оч}}} = \frac{\overline{l_{\text{оч}}}}{\lambda} = \frac{7.38}{1.35} \approx 5.47 \text{ сек}.$$

$\overline{l_{\text{СМО}}} = \overline{l_{\text{оч}}} + \overline{n_{\text{з}}} = 7.38 + 2.7 = 10.08, \qquad \overline{t_{\text{с}}} = \dfrac{\overline{l_{\text{СМО}}}}{\lambda} = \dfrac{10.08}{1.35} \approx 7.47$ сек.

$A = \lambda = 1.35\ ^{\text{запит}}/_{\text{хв}}, \qquad Q = 1.$

Ймовірність того, що в черзі буде не більше трьох запитів для випадку обчислювальних ядер дорівнює:

$$p(r \le 3) = p_0 + p_1 + p_2 + p_3 + p_{3+1} + p_{3+2} + p_{3+3} =$$

$$= (1 - p_{\text{ч}}) + p_{3+1} + p_{3+2} + p_{3+3} =$$

$$= (1 - 0.738) + \left(\frac{2.7^4}{3^1 \cdot 3!} + \frac{2.7^5}{3^2 \cdot 3!} + \frac{2.7^6}{3^3 \cdot 3!}\right) 0.025 =$$

$$= 0.262 + (2.952 + 2.657 + 2.391) 0.025 = 0.262 + 0.2 = 0.462.$$

Висновок: при мінімальній кількості обчислювальних ядер $n_{min} = 3$ ймовірність того, що в черзі буде не більше трьох запитів від користувачів, дорівнює 0.462.

Багатоканальна СМО з необмеженою чергою і обмеженим часом очікування – СМО в якій час перебування заявки в черзі обмежено деяким випадковим строком $t_{\text{оч}}$. Відмінність такої СМО від СМО з обмеженням черги, полягає в тому, що час очікування обслуговування, коли заявка знаходиться в черзі, вважається випадковою величиною, розподіленою за показниковим законом розподілу (пуассонівський потік) з параметром $\nu = 1/\overline{t_{\text{оч}}}$, де ν – інтенсивність потоку вибуття заявок із черги.

При $\nu \to \infty$ багатоканальна СМО з обмеженою чергою і обмеженим часом очікування з кількістю каналів обслуговування n переходить у багатоканальну СМО з відмовами, а при $\nu \to 0$ – у багатоканальну СМО з необмеженою чергою.

Такі СМО зустрічаються часто на практиці (ще мають назву – СМО з «нетерплячими» заявками). Заявки можуть піти з черги, якщо час очікування перевищує деяку величину. Наприклад, різні технологічні системи, в яких затримка початку обслуговування може призвести до втрати якості продукції, системи оперативного управління, коли термінові повідомлення

втрачають цінність, якщо вони не надходять на обслуговування протягом певного часу.

Такі СМО мають три найпростіших потоки, що впливають на стани системи: потік заявок з інтенсивністю λ, потік обслуговування з інтенсивністю μ і потік заявок, що залишають чергу, з інтенсивністю ν.

Граф станів СМО з обмеженим часом очікування наведено на рис. 2.15.

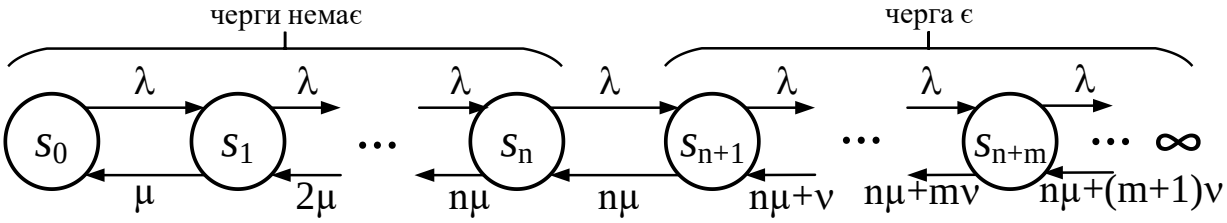


Рисунок 2.15 – Граф станів багатоканальної СМО з обмеженим часом очікування

Можливі стани системи:

s_0 – система вільна (простоює);

s_1 – один канал зайнятий;

s_2 – два канали зайнято;

. . . . . ;

s_n – всі канали системи зайняті обслуговуванням.

s_{n+1} – всі канали зайняті обслуговуванням, у черзі одна заявка;

s_{n+2} – всі канали зайняті обслуговуванням, у черзі дві заявки;

. . . . . ;

s_{n+m} – всі канали зайняті, у черзі m заявок;

. . . . . .

Вводиться величина $\beta = {}^{\nu}/_{\mu}$ – приведена інтенсивність потоку вибуття заявок із черги (середня кількість вибуття з черги необслужених заявок за середній час обслуговування однієї заявки), причому:

$$n\mu + m\nu = (n + m\beta)\mu \quad \text{і} \quad \frac{\lambda}{n\mu + m\nu} = \frac{\lambda}{(n + m\beta)\mu} = \frac{\rho}{n + m\beta}.$$

Тоді граничні ймовірності:

$$\begin{cases} p_0 = \left(\sum_{i=0}^{n} \frac{\rho^i}{i!} + \frac{\rho^n}{n!} \sum_{j=1}^{\infty} \frac{\rho^j}{\prod_{l=1}^{j}(n+l\beta)} \right)^{-1} \\ p_k = \frac{\rho^k}{k!} p_0 \,, \qquad k = 1,2,\dots,n \\ p_{n+m} = \frac{\rho^n}{n!} \cdot \frac{\rho^m}{\prod_{i=1}^{m}(n+i\beta)} p_0 = \frac{\rho^n}{n!} \cdot \frac{\lambda^m}{\prod_{i=1}^{m}(n\mu+i\nu)} p_0, \qquad m = 1,2,\dots \end{cases}.$$

У практичних розрахунках використовують лише декілька членів ряду, так як він швидко спадає. При заданій точності ε знаходять значення m (на практиці обмеження зазвичай доходить до десяти членів ряду) з виразу:

$$\frac{1}{m!}\left(\frac{\rho}{\beta}\right)^m e^{\frac{\rho}{\beta}} \le \varepsilon.$$

Середня кількість заявок у черзі:

$$\overline{l_{\text{оч}}} = \sum_{i=1}^{\infty} i p_{n+i} = \frac{\rho^n}{n!} p_0 \sum_{i=1}^{\infty} \frac{i\rho^i}{\prod_{j=1}^{i}(n+j\beta)}$$

визначається також з урахуванням знайденої кількості членів нескінченного ряду m для досягнення заданої точності ε.

Абсолютна пропускна здатність A дорівнює

$$A = \lambda - \nu \cdot \overline{l_{\text{оч}}},$$

де $\nu \cdot \overline{l_{\text{оч}}}$ – заявки, що вибули з черги в одиницю часу.

Відносна пропускна здатність Q дорівнює

$$Q = \frac{A}{\lambda} = 1 - \frac{\nu}{\lambda}\overline{l_{\text{оч}}}.$$

Ймовірність того, що заявка буде обслуговувана:

$$p_{\text{обсл}} = 1 - \frac{\nu}{\lambda}\overline{l_{\text{оч}}}.$$

Ймовірність вибуття заявки з черги по причині перевищення часу перебування в системі:

$$p_{\text{виб}} = 1 - p_{\text{обсл}} = \frac{\nu}{\lambda}\overline{l_{\text{оч}}}.$$

Середня кількість зайнятих каналів:

$$\overline{n_з} = \frac{A}{\mu} = \rho - \beta\overline{l_{оч}},$$

або ще середнє число зайнятих каналів $\overline{n_з}$ можна знайти як математичне сподівання випадкової величини:

$$\overline{n_з} = 0 \cdot p_0 + 1 \cdot p_1 + 2p_2 + \ldots + (n-1)p_{n-1} + n[1 - (p_0 + p_1 + \cdots + p_{n-1})],$$

або:

$$\overline{n_з} = n - \sum_{i=0}^{n}(n-i)p_i.$$

Це дозволяє обчислити середню кількість заявок у черзі $\overline{l_{оч}}$ та інші показники, не виконуючи додавання нескінченного ряду.

$$\overline{l_{оч}} = \frac{\rho}{\beta} - \frac{\overline{n_з}}{\beta}, \qquad A = \mu\overline{n_з}, \qquad Q = \frac{\overline{n_з}}{\rho}, \qquad p_{виб} = 1 - \frac{\overline{n_з}}{\rho}.$$

Інтенсивність вибуття заявок без обслуговування:

$$\omega = \overline{l_{оч}} \cdot \nu.$$

Середня кількість заявок, що знаходяться в СМО:

$$\overline{l_{СМО}} = \overline{l_{оч}} + \overline{n_з} = \frac{\rho + (\beta - 1)\overline{n_з}}{\beta}.$$

За формулами Літтла знаходяться середній час очікування заявки в черзі $\overline{t_{оч}}$, середній час обслуговування заявки $\overline{t_{обсл}}$ та середній час перебування заявки в системі $\overline{t_с}$:

$$\overline{t_{оч}} = \frac{\overline{l_{оч}}}{\lambda} = \frac{1-Q}{\nu}, \qquad \overline{t_{обсл}} = \frac{\overline{n_з}}{\lambda}, \qquad \overline{t_с} = \frac{\overline{l_{СМО}}}{\lambda} = \frac{\rho + (\beta-1)\overline{n_з}}{\lambda\beta}.$$

Ймовірність виникнення черги заявок:

$$p_ч = \sum_{i=1}^{\infty} p_{n+i} = 1 - p_0 \sum_{i=0}^{n} p_i = 1 - p_0 \sum_{i=0}^{n} \frac{\rho^i}{i!}.$$

Приклад. У центрі кіберзахисту працюють дві автоматизовані системи реагування на кіберзагрози. Спеціальні системи моніторингу постійно сканують вхідний трафік на наявність шкідливих атак. Якщо в момент виявлення нової загрози обидві системи обробки вже задіяні на обробці попередніх атак, виявлена загроза тимчасово відстежується системою

моніторингу. Щойно одна з систем звільняється, моніторинг передає їй очікуючу загрозу для негайної обробки – за умови, що з моменту її виявлення не пройшло більш як 2 хвилини (тобто вона ще активна в мережі). Відомо, що: кожна система обробки атак справляється з однією атакою за одну хвилину ($\mu = 1$ атака/хв), атаки надходять з інтенсивністю ($\lambda = 0.8$ атак/хв), середній час, упродовж якого загроза залишається активною (тобто може бути перехоплена після виявлення), становить 2 хв. Потрібно визначити основні характеристики ефективності СМО.

Розв'язання. Параметри системи:

- кількість каналів обслуговування $n = 2$;
- інтенсивність вхідного потоку заявок $\lambda = 0.8$ атак/хв;
- інтенсивність потоку обслуговування $\mu = 1$ атака/хв;
- середній час, що обмежує перебування заявки в черзі, $\overline{t_{\text{оч}}} = 2$ хв;
- інтенсивність потоку вибуття $\nu = \frac{1}{\overline{t_{\text{оч}}}} = 0.5$ атак/хв.

$$\rho = \frac{\lambda}{\mu} = \frac{0.8}{1} = 0.8, \qquad \beta = \frac{\nu}{\mu} = \frac{0.5}{1} = 0.5,$$

$$p_0 = \left(\sum_{i=0}^{n} \frac{\rho^i}{i!} + \frac{\rho^n}{n!} \sum_{j=1}^{\infty} \frac{\rho^j}{\prod_{l=1}^{j} (n + l\beta)} \right)^{-1} =$$

$$= \left(\sum_{i=0}^{2} \frac{0.8^i}{i!} + \frac{0.8^2}{2!} \sum_{j=1}^{\infty} \frac{0.8^j}{\prod_{l=1}^{j} (2 + l \cdot 0.5)} \right)^{-1}.$$

Обмежимося точністю $\varepsilon = 0.01$:

$$\frac{1}{m!} \left(\frac{0.8}{0.5} \right)^m e^{\frac{0.8}{0.5}} \leq 0.01,$$

або

$$\frac{(1.6)^m}{m!} \leq \frac{0.01}{4.953} = 0.02,$$

і, послідовно перебираючи значення m, отримуємо, що необхідна точність досягається при $m = 7$:

$$\frac{(1.6)^7}{7!} = 0.005 \le 0.02.$$

$$p_0 = \left(1 + 0.8 + \frac{0.8^2}{2!} + \frac{0.8^2}{2!}S\right)^{-1}, \text{де } S = \sum_{j=1}^{8} \frac{0.8^j}{\prod_{l=1}^{j}(2 + l \cdot 0.5)}.$$

Знайдемо S як суму $S = a_1 + a_2 + a_3 + a_4 + a_5 + a_6 + a_7$:

$$a_1 = \frac{0.8}{2 + 0.5} = 0.32,$$

$$a_2 = \frac{0.8^2}{(2 + 0.5)(2 + 2 \cdot 0.5)} = 0.085,$$

$$a_3 = \frac{0.8^3}{(2 + 0.5)(2 + 2 \cdot 0.5)(2 + 3 \cdot 0.5)} = 0.020,$$

$$a_4 = \frac{0.8^4}{(2 + 0.5)(2 + 2 \cdot 0.5)(2 + 3 \cdot 0.5)(2 + 4 \cdot 0.5)} = 0.004,$$

$$a_5 = \frac{0.8^5}{(2 + 0.5)(2 + 2 \cdot 0.5)(2 + 3 \cdot 0.5)(2 + 4 \cdot 0.5)(2 + 5 \cdot 0.5)} = 0.0007,$$

$$a_6 = \frac{0.8^6}{(2 + 0.5)(2 + 2 \cdot 0.5)(2 + 3 \cdot 0.5)(2 + 4 \cdot 0.5)(2 + 5 \cdot 0.5)(2 + 6 \cdot 0.5)} =$$

$$= 0.0001,$$

$$a_7 = \frac{0.8^7}{(2 + 0.5)(2 + 2 \cdot 0.5)(2 + 3 \cdot 0.5)(2 + 4 \cdot 0.5)(2 + 5 \cdot 0.5)(2 + 6 \cdot 0.5)} \times$$

$$\times \frac{1}{(2 + 7 \cdot 0.5)} = 0.00002,$$

$$S = 0.32 + 0.085 + 0.02 + 0.004 + 0.0007 + 0.0001 + 0.00002 \approx 0.43,$$

$$p_0 = \left(1 + 0.8 + \frac{0.8^2}{2!} + \frac{0.8^2}{2!}0.43\right)^{-1} = 0.443.$$

$$p_1 = \frac{\rho^1}{1!}p_0 = 0.8 \cdot 0.443 = 0.354, \qquad p_2 = \frac{\rho^2}{2!}p_0 = 0.142.$$

$$p_{\text{ч}} = 1 - p_0 \sum_{i=0}^{2} \frac{\rho^i}{i!} = 1 - p_0\left(1 + \frac{\rho^1}{1!} + \frac{\rho^2}{2!}\right) = 1 - (p_0 + p_1 + p_2) =$$

$$= 1 - (0.443 + 0.354 + 0.142) = 0.061.$$

$$\overline{n_з} = n - \sum_{i=0}^{n}(n-i)p_i = 2 - (2p_0 + 1p_1 + 0) = 2 - 2 \cdot 0.443 - 0.354 = 0.76,$$

$$\overline{l_{оч}} = \frac{\rho}{\beta} - \frac{\overline{n_з}}{\beta} = \frac{0.8}{0.5} - \frac{0.76}{0.5} = 0.08, \qquad \overline{l_{СМО}} = \overline{l_{оч}} + \overline{n_з} = 0.08 + 0.76 = 0.84,$$

$$A = \mu\overline{n_з} = 1 \cdot 0.76 = 0.76, \qquad Q = \frac{\overline{n_з}}{\rho} = \frac{\overline{n_з}}{\rho} = \frac{0.76}{0.8} = 0.95,$$

$$p_{виб} = 1 - \frac{\overline{n_з}}{\rho} = 1 - \frac{0.76}{0.8} = 0.05,$$

$$\overline{t_{оч}} = \frac{\overline{l_{оч}}}{\lambda} = \frac{0.08}{0.8} = 0.1, \qquad \overline{t_{обсл}} = \frac{\overline{n_з}}{\lambda} = \frac{0.76}{0.8} = 0.95,$$

$$\overline{t_с} = \frac{\overline{l_{СМО}}}{\lambda} = \frac{0.84}{0.8} = 1.05.$$

Висновок: при заданих параметрах 95 % заявок, що надходять, обслуговуються системою і тільки 5 % «нетерплячих» заявок залишають її необслуженими, тобто 95 % кібератак обробляються системою, а 5 % кібератак пропускаються.

Одноканальна СМО з необмеженою чергою і обмеженим часом очікування є окремим випадком багатоканальної СМО з необмеженою чергою і обмеженим часом очікування ($n = 1$).

Розрахункові формули:

$$\begin{cases} p_0 = \left(1 + \rho + \rho\sum_{j=1}^{\infty}\frac{\rho^j}{\prod_{l=1}^{j}(1+l\beta)}\right)^{-1} \\ p_1 = \rho p_0 \\ p_{1+m} = \frac{\rho^{m+1}}{\prod_{i=1}^{m}(1+i\beta)}p_0, \qquad m = 1,2,\ldots \end{cases},$$

$$p_ч = 1 - p_0(1+\rho),$$

$$\overline{n_з} = 1 - p_0,$$

$$A = \mu\overline{n_з},$$

$$Q = \frac{\overline{n_з}}{\rho},$$

$$p_{\text{обсл}} = Q = \frac{1 - p_0}{\rho},$$

$$p_{\text{виб}} = 1 - p_{\text{обсл}} = 1 - \frac{1 - p_0}{\rho},$$

$$\omega = \lambda - A = \lambda - \mu(1 - p_0),$$

$$\overline{l_{\text{оч}}} = \frac{\rho - \bar{n}_{\text{з}}}{\beta},$$

$$\overline{l_{\text{СМО}}} = \overline{l_{\text{оч}}} + \bar{n}_{\text{з}} = \frac{\rho + (\beta - 1)(1 - p_0)}{\beta},$$

$$\overline{t_{\text{оч}}} = \frac{\overline{l_{\text{оч}}}}{\lambda} = \frac{\rho - \bar{n}_{\text{з}}}{\lambda\beta} = \frac{1 - Q}{\nu},$$

$$\overline{t_{\text{обсл}}} = \frac{\bar{n}_{\text{з}}}{\lambda} = \frac{Q}{\mu},$$

$$\bar{t}_{\text{с}} = \frac{\overline{l_{\text{СМО}}}}{\lambda} = \frac{1 - Q}{\nu} + \frac{Q}{\mu}.$$

Приклад. У системі технічної підтримки мобільного застосунку реалізовано один канал обробки звернень користувачів (один черговий спеціаліст або серверна служба). Аналіз показав, що: у святкові та вихідні дні нові звернення надходять у середньому кожні 15 хвилин (тобто з інтенсивністю $\lambda = 4$ звернення на годину), середній час обробки одного звернення – 12 хвилин ($\mu = 5$ звернень на годину), якщо користувач очікує відповіді більше 10 хвилин, він іде й скасовує звернення (відмова), середній дохід, що приносить успішно оброблене звернення (наприклад, оформлене підписання або продаж послуги) $\mathrm{C} = 200$ грн. Потрібно: визначити основні характеристики ефективності СМО середні втрати доходу через відмови користувачів.

Розв'язання. Параметри системи:

- кількість каналів обслуговування $n = 1$;
- інтенсивність вхідного потоку заявок $\lambda = 4$ звернень/год.;
- інтенсивність потоку обслуговування $\mu = 5$ звернень/год.

- середній час, що обмежує перебування заявки в черзі, $T_{\text{оч}} = 10$ хв $= 1/6$ год.

$$\nu = \frac{1}{T_{\text{оч}}} = 6, \qquad \rho = \frac{\lambda}{\mu} = \frac{4}{5} = 0.8, \qquad \beta = \frac{\nu}{\mu} = \frac{6}{5} = 1.2,$$

$$p_0 = (1 + \rho + \rho S)^{-1}, \text{де } S = \sum_{j=1}^{8} \frac{\rho^j}{\prod_{l=1}^{j}(1 + l \cdot \beta)},$$

Обмежимося точністю $\varepsilon = 0.01$:

$$\frac{1}{m!}\left(\frac{\rho}{\beta}\right)^m e^{\frac{\rho}{\beta}} \leq 0.01,$$

або

$$\frac{1}{m!}\left(\frac{0.8}{1.2}\right)^m \leq \varepsilon e^{-\frac{\rho}{\beta}} \approx 0.00513,$$

і, послідовно перебираючи значення m, отримуємо, що необхідна точність досягається при $m = 5$:

$$\frac{1}{5!}\left(\frac{0.8}{1.2}\right)^5 = 0.001 \leq 0.00513.$$

Знайдемо S як суму $S = a_1 + a_2 + a_3 + a_4 + a_5$:

$$a_1 = \frac{0.8}{1 + 1.2} = 0.3636,$$

$$a_2 = \frac{0.8^2}{(1 + 1.2)(1 + 2 \cdot 1.2)} = 0.0856,$$

$$a_3 = \frac{0.8^3}{(1 + 1.2)(1 + 2 \cdot 1.2)(1 + 3 \cdot 1.2)} = 0.0149,$$

$$a_4 = \frac{0.8^4}{(1 + 1.2)(1 + 2 \cdot 1.2)(1 + 3 \cdot 1.2)(1 + 4 \cdot 1.2)} = 0.0021,$$

$$a_5 = \frac{0.8^5}{(1 + 1.2)(1 + 2 \cdot 1.2)(1 + 3 \cdot 1.2)(1 + 4 \cdot 1.2)(1 + 5 \cdot 1.2)} = 0.0002,$$

$$S \approx 0.467,$$

$$p_0 = (1 + 0.8 + 0.8 \cdot 0.467)^{-1} \approx 0.46.$$

$$p_{\text{ч}} = 1 - 0.46(1 + 0.8) \approx 0.172,$$

$$\overline{n_{\text{з}}} = 1 - p_0 = 1 - 0.46 = 0.54,$$

$$A = \mu\overline{n_з} = 5 \cdot 0.54 = 2.7, \qquad Q = \frac{\overline{n_з}}{\rho} = \frac{0.54}{0.8} = 0.675,$$

$$\omega = \lambda - A = 4 - 2.7 = 1.3,$$

$$p_{\text{виб}} = 1 - \frac{1 - p_0}{\rho} = 1 - \frac{1 - 0.46}{0.8} = 0.325,$$

$$\overline{l_{\text{оч}}} = \frac{\rho - \overline{n_з}}{\beta} = \frac{0.8 - 0.54}{1.2} \approx 0.217,$$

$$\overline{l_{\text{СМО}}} = \frac{\rho + (\beta - 1)(1 - p_0)}{\beta} = \frac{0.8 + 0.2 \cdot 0.54}{1.2} \approx 0.757,$$

$$\overline{t_{\text{оч}}} = \frac{1 - Q}{\nu} = \frac{1 - 0.675}{6} \approx 0.054 \text{ год.} = 3.25 \text{ хв},$$

$$\overline{t_{\text{обсл}}} = \frac{Q}{\mu} = \frac{0.675}{5} = 0.135 \text{ год.} = 8.1 \text{ хв},$$

$$\overline{t_{\text{с}}} = \frac{\overline{l_{\text{СМО}}}}{\lambda} = \frac{1 - Q}{\nu} + \frac{Q}{\mu} = 11.35 \text{ хв}.$$

Середній прибуток, який отримується за годину роботи:

$\text{П} = \text{С} \cdot A = 200 \cdot 2.7 = 540$ грн.,

і середні втрати в результаті скасування замовлень за годину роботи

$\text{В} = \text{С} \cdot \omega = 200 \cdot 1.3 = 260$ грн.

Висновок: внаслідок недостатньої швидкості обслуговування втрачається приблизно третина клієнтів і, відповідно, третина можливого доходу.

Багатоканальні замкнуті СМО – такі системи, в яких інтенсивність потоку заявок, які надходять, залежить від станів самої СМО. Для таких систем характерним є те, що джерело заявок є обмеженим.

Як правило, потоком заявок у такій СМО є потік відмов (несправностей, збоїв) від групи пристроїв, що працюють (джерела), а λ – інтенсивність потоку відмов одного пристрою. Є m пристроїв, які можуть виходити з ладу через несправності, та n каналів для їх обслуговування. Припускається, що $m > n$. Стан замкнутої СМО пов’язують із кількістю пристроїв, що відмовили. Сумарна інтенсивність потоку відмов усієї групи

пристроїв залежить від кількості пристроїв, що вже відмовили. Джерело (пристрій), яке подало заявку, може подати наступну заявку на обслуговування лише тоді, якщо подана ним попередня заявка вже обслужена.

Таким чином, є лише два можливі стани джерела заявок:

- активний (пристрій працездатний) – джерело заявок може подати чергову заявку на обслуговування, тобто вже обслужено раніше подану їм останню заявку;
- пасивний (пристрій вийшов з ладу) – остання подана джерелом заявка ще обслуговується або стоїть у черзі, подати нову заявку джерело не може.

Граф станів замкнутої СМО наведено на рис. 2.16.

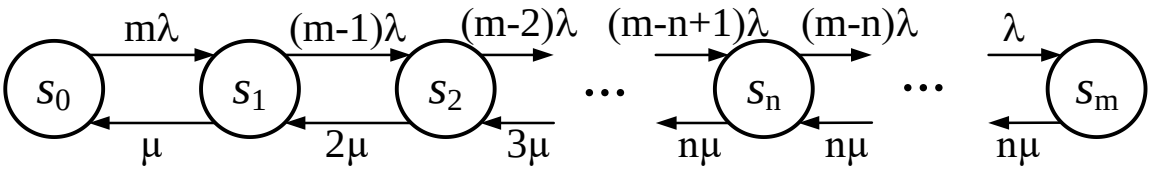


Рисунок 2.16 – Граф станів багатоканальної замкнутої СМО

Можливі стани системи:

s_0 – система простоює (всі пристрої працездатні);

s_1 – один канал зайнятий (один пристрій вийшов з ладу та виконується його ремонт);

s_2 – два канали зайнято (два пристрої вийшло з ладу та виконується їх ремонт);

. . . . . ;

s_n – всі канали системи зайняті обслуговуванням (n пристроїв вийшли з ладу).

s_{n+1} – всі канали зайняті обслуговуванням, у черзі одна заявка ($n+1$ пристроїв вийшли з ладу, n пристроїв ремонтуються, один чекає на ремонт);

s_{n+2} – всі канали зайняті обслуговуванням, у черзі дві заявки ($n+2$ пристроїв вийшли з ладу, n пристроїв ремонтуються, два чекають на ремонт);

. . . . . ;

s_m – всі канали зайняті обслуговуванням, у черзі $(m-n)$ заявок (всі m пристроїв вийшли з ладу, n пристроїв проходять ремонт, інші $(m-n)$ чекають на ремонт).

Граничні ймовірності:

$$\begin{cases} p_0 = \left(\sum_{i=1}^{n} \frac{m!}{(m-i)!} \frac{\rho^i}{i!} + \frac{\rho^n}{n!} \sum_{i=n+1}^{m} \frac{\rho^{i-n} \cdot m!}{(m-i)! \cdot n^{i-n}} \right)^{-1} \\ p_k = \frac{m!}{(m-k)!} \frac{\rho^k}{k!} p_0 \,, \qquad k = 1,2, \dots, n \\ p_k = \frac{m!}{(m-k)!} \frac{\rho^k}{n! \cdot n^{k-n}} p_0, \qquad k = n+1, n+2, \dots, m \end{cases}.$$

Ймовірність утворення черги заявок:

$$p_{\text{ч}} = \frac{\rho^n}{n!} p_0 \sum_{i=n+1}^{m} \frac{\rho^{i-n} \cdot m!}{(m-i)! \cdot n^{i-n}}.$$

Оскільки відмови в обслуговуванні заявки немає, то $p_{\text{відм}} = 0$ і відносна пропускна здатність Q дорівнює

$$Q = p_{\text{обсл}} = 1 - p_{\text{відм}} = 1.$$

Середня кількість зайнятих каналів $\overline{n_з}$ (відповідає середній кількості заявок, що знаходяться під обслуговуванням):

$$\overline{n_з} = n - \sum_{i=0}^{n} (n-i) p_i.$$

Решту всіх характеристик СМО можна знайти через отримане значення $\overline{n_з}$.

Абсолютна пропускна здатність системи дорівнює середній кількості зайнятих каналів, помноженій на продуктивність кожного каналу (інтенсивність потоку обслуговування):

$$A = \nu = \overline{n_з} \cdot \mu,$$

де ν – інтенсивність потоку обслужених заявок.

Середня кількість заявок, що знаходяться в системі:

$$\overline{l_{\text{СМО}}} = m - \frac{\overline{n_{\text{з}}}}{\rho} = \sum_{i=0}^{m} i p_i.$$

Середня кількість заявок, що перебувають у черзі – різниця середньої кількості заявок, що знаходяться в системі, та середньої кількості заявок, що знаходяться під обслуговуванням:

$$\overline{l_{\text{оч}}} = m - \overline{n_{\text{з}}}\left(\frac{1}{\rho} + 1\right).$$

Вводяться також нові характеристики для замкнутих СМО:

$N_{\text{пас}}$ – середня кількість джерел, що знаходяться в пасивному стані;

$N_{\text{акт}}$ – середня кількість джерел, що знаходяться в активному стані;

Λ – середня інтенсивність сумарного потоку заявок;

$P_{\text{акт}}$ – ймовірність того, що джерело знаходиться в активному стані.

Оскільки всі заявки будуть обслужені, середня інтенсивність потоку заявок Λ, тобто середня кількість поданих в одиницю часу заявок, дорівнює абсолютній пропускній здатності системи:

$$\Lambda = A = \overline{n_{\text{з}}} \cdot \mu = \lambda \cdot N_{\text{акт}},$$

звідки можна знайти

$$N_{\text{акт}} = \frac{A}{\lambda} = \frac{\overline{n_{\text{з}}}}{\rho}.$$

Середня кількість джерел, що знаходяться в пасивному стані:

$$N_{\text{пас}} = m - N_{\text{акт}} = m - \frac{\overline{n_{\text{з}}}}{\rho}.$$

Ймовірність того, що джерело заявок знаходиться в активному стані:

$$P_{\text{акт}} = \frac{N_{\text{акт}}}{m} = \frac{A}{m\lambda}.$$

За формулами Літтла знаходяться середній час очікування заявки в черзі $\overline{t_{\text{оч}}}$, середній час обслуговування заявки $\overline{t_{\text{обсл}}}$ та середній час перебування заявки в системі $\overline{t_{\text{с}}}$:

$$\overline{t_{\text{оч}}} = \frac{\overline{l_{\text{оч}}}}{\Lambda}, \qquad \overline{t_{\text{обсл}}} = \frac{\overline{n_{\text{з}}}}{\Lambda} = \frac{1}{\mu}, \qquad \overline{t_{\text{с}}} = \frac{\overline{l_{\text{СМО}}}}{\Lambda}.$$

Приклад. У технічному центрі два спеціалісти обслуговують групу з 9 автоматизованих робочих станцій (АРМ), які періодично потребують технічного втручання. Кожна станція в середньому один раз на годину потребує обслуговування ($\lambda = 1$ виклик/год.). Обслуговування однієї станції триває в середньому 6 хвилин ($\mu = 10$ обслуговувань/год.). Знайти ймовірність того, що принаймні три АРМ перебувають у справному стані (не очікують обслуговування).

Розв'язання: параметри системи:

$n = 2, \qquad m = 9, \qquad \lambda = 1$ виклик/год.,

$\mu = 10$ обслуговувань/год., $\qquad \rho = \dfrac{\lambda}{\mu} = \dfrac{1}{10} = 0.1.$

Враховуючи те, що

$$\begin{cases} p_0 = \left(\sum\limits_{i=1}^{n} \dfrac{m!}{(m-i)!} \dfrac{\rho^i}{i!} + \dfrac{\rho^n}{n!} \sum\limits_{i=n+1}^{m} \dfrac{\rho^{i-n} \cdot m!}{(m-i)! \cdot n^{i-n}} \right)^{-1} \\ p_k = \dfrac{m!}{(m-k)!} \dfrac{\rho^k}{k!} p_0, \qquad k = 1,2, \dots, n \\ p_k = \dfrac{m!}{(m-k)!} \dfrac{\rho^k}{n! \cdot n^{k-n}} p_0, \qquad k = n+1, n+2, \dots, m \end{cases},$$

можна спочатку знайти співвідношення всіх ймовірностей p_k до p_0, а потім знайти p_0:

$$\frac{p_k}{p_0} = \frac{9!}{(9-k)!} \frac{0.1^k}{k!}, \qquad k = 0, 1, 2,$$

$$\frac{p_k}{p_0} = \frac{9!}{(9-k)!} \frac{0.1^k}{2! \cdot 2^{k-2}}, \qquad k = 3, 4, \dots, 9.$$

Оскільки

$$\sum_{k=0}^{9} p_k = 1,$$

то

$$\sum_{k=0}^{9} \frac{p_k}{p_0} = \frac{1}{p_0},$$

звідки

$$p_0 = \frac{1}{\sum_{k=0}^{9} \frac{p_k}{p_0}}.$$

Розрахунок ймовірностей наведено в табл. 2.1.

Таблиця 2.1 – Проміжні розрахунки ймовірностей

k	$\frac{p_k}{p_0}$	p_k
0	1	0.410601102
1	0.9	0.369540992
2	0.36	0.147816397
3	0.126	0.051735739
4	0.0378	0.015520722
5	0.00945	0.00388018
6	0.00189	0.000776036
7	0.000284	0.000116611
8	0.00002835	0.0000116405
9	0.0000014175	0.000000582027
$\sum_{k=0}^{9}$	2.435453768	1

Ймовірність того, що не менше трьох АРМ перебувають у справному стані (не менше трьох джерел заявок будуть у активному стані):

$$P(N_{\text{акт}} \geq 3) = P(N_{\text{пас}} \leq 6) = \sum_{k=0}^{6} p_k = 0.999871167 \approx 1.$$

Висновок: $P(N_{\text{акт}} \geq 3) \approx 1$, тобто це практично достовірна подія.

Одноканальна замкнута СМО – найпростіший випадок ($n = 1$) замкнутої СМО. Загальна кількість станів такої системи – $m + 1$:

s_0 – система простоює (всі пристрої працездатні);

s_1 – канал зайнятий (один пристрій вийшов з ладу та виконується його ремонт);

s_2 – канал зайнято, у черзі знаходиться одна заявка (два пристрої вийшло з ладу, один ремонтується, один чекає на ремонт);

. . . . . ;

s_m – канал зайнято, у черзі знаходиться $(m-1)$ заявок, (всі m пристроїв вийшли з ладу, один пристрій проходить ремонт, інші $(m-1)$ чекають на ремонт).

Розмічений граф станів системи наведено на рис. 2.17.

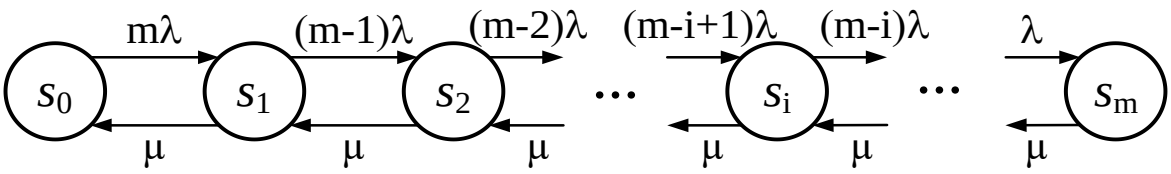


Рисунок 2.17 – Граф станів одноканальної замкнутої СМО

На відміну від багатоканальної замкнутої СМО вирази для граничних ймовірностей суттєво спрощуються:

$$\begin{cases} p_0 = (1 + m\rho + m(m-1)\rho^2 + \cdots + m(m-1)(m-2)\cdot \ldots \cdot 1\rho^m)^{-1} \\ p_1 = m\rho p_0 \\ p_2 = m(m-1)\rho^2 p_0 \\ p_3 = m(m-1)(m-2)\rho^3 p_0 \\ \quad \ldots \\ p_m = m(m-1)(m-2)\cdot \ldots \cdot (m-(m-1))\rho^m p_0 \end{cases},$$

або

$$\begin{cases} p_0 = \left(m! \sum_{i=0}^{m} \dfrac{\rho^i}{(m-i)!} \right)^{-1} \\ p_k = \dfrac{m!}{(m-k)!}\rho^k p_0, \qquad k = 1,2,\ldots,m \end{cases}.$$

Відмови в обслуговуванні заявки немає, тобто $p_{\text{відм}} = 0$ і відносна пропускна здатність Q дорівнює

$$Q = p_{\text{обсл}} = 1 - p_{\text{відм}} = 1.$$

Абсолютна пропускна здатність системи A дорівнює

$$A = (1 - p_0)\mu.$$

Оскільки всі заявки будуть обслужені, середня інтенсивність потоку заявок дорівнює абсолютній пропускній здатності системи:

$$\Lambda = A = (1 - p_0)\cdot \mu = \lambda \cdot N_{\text{акт}},$$

звідки:

$$N_{\text{акт}} = \frac{A}{\lambda} = \frac{1 - p_0}{\rho}.$$

Середня кількість джерел, що знаходяться в пасивному стані:

$$N_{\text{пас}} = m - N_{\text{акт}} = m - \frac{1 - p_0}{\rho}$$

і дорівнює середній кількості заявок, що знаходяться в системі:

$$\overline{l_{\text{СМО}}} = N_{\text{пас}} = m - \frac{1 - p_0}{\rho}.$$

Ймовірність наявності черги в системі:

$$p_{\text{ч}} = \sum_{i=2}^{m} p_i = m!\, p_0 \sum_{i=2}^{m} \frac{\rho^i}{(m-i)!} = 1 - p_0 - p_1.$$

Середня кількість заявок у черзі:

$$\overline{l_{\text{оч}}} = m - (1 - p_0)\left(\frac{1}{\rho} + 1\right).$$

Середня кількість зайнятих каналів:

$$\overline{n_{\text{з}}} = 1 - p_0.$$

Ймовірність того, що джерело заявок знаходиться в активному стані:

$$P_{\text{акт}} = \frac{N_{\text{акт}}}{m} = \frac{A}{m\lambda}.$$

Середній час очікування заявки в черзі $\overline{t_{\text{оч}}}$, середній час обслуговування заявки $\overline{t_{\text{обсл}}}$ та середній час перебування заявки в системі $\overline{t_{\text{с}}}$:

$$\overline{t_{\text{оч}}} = \frac{\overline{l_{\text{оч}}}}{\Lambda}, \qquad \overline{t_{\text{обсл}}} = \frac{\overline{n_{\text{з}}}}{\Lambda}, \qquad \overline{t_{\text{с}}} = \frac{\overline{l_{\text{СМО}}}}{\Lambda}.$$

Приклад. Група з 15 мобільних терміналів збору даних (наприклад, автономних IoT-пристроїв) періодично підключаються до єдиної базової станції для передачі зібраної інформації та підзарядки. Середній час автономної роботи одного термінала (між сесіями обслуговування) становить 5 діб. Для кожного підключення потрібно 8 годин на обслуговування, яке здійснюється через один порт обміну (один канал зв’язку/роз’єднання). Якщо порт зайнятий, інші термінали чекають на доступ у черзі. Необхідно

визначити: середній час простоювання термінала, середній час очікування терміналом початку обслуговування, середній час простоювання базової станції (тобто коли вона вільна та не обслуговує жоден термінал).

Розв'язання: параметри системи:

$$n = 1, \qquad m = 15, \qquad \lambda = \frac{1}{5} = 0.2 \text{ заявок/доба},$$

$$\mu = \frac{24}{8} = 3 \text{ заявки/доба}, \qquad \rho = \frac{\lambda}{\mu} = \frac{0.2}{3} = 0.067.$$

З виразу для граничних ймовірностей

$$\begin{cases} p_0 = \left(m! \sum_{i=0}^{m} \frac{\rho^i}{(m-i)!} \right)^{-1} \\ p_k = \frac{m!}{(m-k)!} \rho^k p_0, \qquad k = 1,2, \dots, m \end{cases}$$

можна отримати

$$\frac{p_k}{p_0} = 0.067^k \frac{15!}{(15-k)!}, \qquad k = 0, 1, \dots, 15.$$

Розрахунок ймовірностей наведено в табл. 2.2.

Таблиця 2.2 – Проміжні розрахунки ймовірностей

k	$\frac{p_k}{p_0}$	k	$\frac{p_k}{p_0}$
0	1	8	0.10535799
1	1.005	9	0.0494129
2	0.94269	10	0.01986399
3	0.82108299	11	0.00665444
4	0.66015072	12	0.00178339
5	0.48653108	13	0.00035846
6	0.32597583	14	0.000048034
7	0.19656342	15	0.0000032183

$$\sum_{k=0}^{15} \frac{p_k}{p_0} = 5.621476463.$$

Оскільки

$$\sum_{k=0}^{15} p_k = 1,$$

то

$$\sum_{k=0}^{15}\frac{p_k}{p_0}=\frac{1}{p_0}=5.621476463,$$

звідки

$$p_0=\frac{1}{5.621476463}=0.177889209.$$

$$\overline{l_{\text{СМО}}}=N_{\text{пас}}=m-\frac{1-p_0}{\rho}=15-\frac{1-0.177889209}{0.067}\approx 2.731,$$

$$\overline{l_{\text{оч}}}=m-(1-p_0)\left(\frac{1}{\rho}+1\right)=15-(1-0.177889209)\left(\frac{1}{0.067}+1\right)\approx 1.909,$$

$$\Lambda=A=(1-p_0)\cdot\mu=(1-0.177889209)\cdot 3\approx 2.466.$$

Середній час простоювання термінала:

$$\overline{t_{\text{с}}}=\frac{\overline{l_{\text{СМО}}}}{\Lambda}=\frac{2.731}{2.466}\approx 1.107 \text{ доби}.$$

Середній час очікування терміналом початку обслуговування:

$$\overline{t_{\text{оч}}}=\frac{\overline{l_{\text{оч}}}}{\Lambda}=\frac{1.909}{2.466}\approx 0.774 \text{ доби}.$$

Середній час простоювання базової станції дорівнює середньому часу очікування чергової заявки мінус середній час обслуговування заявки:

$$\overline{t_{\text{прост}}}=\frac{1}{\Lambda}-\frac{8}{24}=\frac{1}{2.466}-\frac{8}{24}\approx 0.073 \text{ доби}.$$

СМО з «взаємодопомогою» між каналами. На практиці зустрічаються СМО, де заявка може одночасно обслуговуватися двома і більше каналами. Наприклад, один і той самий комп'ютерний модуль, що вийшов з ладу, можуть одночасно обслуговувати два технічних спеціалісти. Така «взаємодопомога» між каналами може мати місце як у відкритих, так і замкнутих СМО. При розгляді СМО із взаємодопомогою між каналами необхідно враховувати два фактори:

- ступінь прискорення обслуговування заявки кількома каналами одночасно по зрівнянню з одним каналом;

- організація дисципліни «взаємодопомоги» – скільки каналів і коли вони беруть участь у обслуговування однієї й тієї заявки.

Якщо над обслуговуванням заявки працює не один, а кілька каналів, то інтенсивність потоку обслуговувань пропорційно збільшується їх кількості, але при деякому критичному значенні кількості каналів подальше збільшення числа зайнятих каналів вже не підвищує інтенсивності обслуговування.

СМО з відмовами і «взаємодопомогою» між каналами типу «всі як один». Найпростіший випадок дисципліни «взаємодопомоги» – «всі як один». Це означає, що з появою однієї заявки її починають обслуговувати всі канали одразу і вони залишаються зайнятими, доки не закінчиться обслуговування цієї заявки; потім всі канали перемикаються на обслуговування іншої заявки (якщо вона є) або чекають на її появу. Очевидно, що в цьому випадку всі канали працюють як один, СМО стає одноканальною, але з більш високою інтенсивністю обслуговування.

Інтенсивність спільного обслуговування заявки всіма каналами дорівнює $\varphi(n) = n\mu$. Середній час обробки однієї заявки всіма n каналами $1/n\mu$.

Розмічений граф станів системи наведено на рис. 2.18.

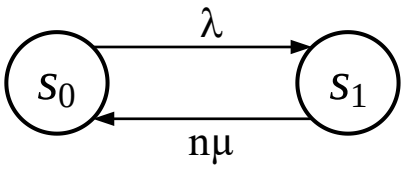


Рисунок 2.18 – Граф станів СМО з відмовами і «взаємодопомогою» між каналами типу «всі як один»

Стани системи:

S_0 – усі n каналів вільні;

S_1 – усі n каналів зайняті.

Вводиться позначення:

$$\psi = \frac{\lambda}{n\mu} = \frac{\rho}{n},$$

і тоді формули, що характеризують роботу даної СМО, можна отримати з формул для одноканальної СМО з відмовами, замінивши в них параметр ρ на ψ.

Граничні ймовірності:

$$p_0 = p_{\text{обсл}} = \frac{n\mu}{\lambda + n\mu} = \frac{n}{\rho + n}, \qquad p_1 = p_{\text{відм}} = \frac{\lambda}{n\mu} p_0 = \frac{\lambda}{\lambda + n\mu} = \frac{\rho}{\rho + n}.$$

Відносна Q та абсолютна A пропускна здатність:

$$Q = p_{\text{обсл}} = \frac{n}{\rho + n}, \qquad A = \lambda Q = \frac{\lambda \cdot n}{\rho + n}.$$

Середня кількість зайнятих каналів:

$$\overline{n_{\text{з}}} = 0 \cdot p_0 + n \cdot p_1 = \frac{n\rho}{\rho + n} = \rho Q.$$

Для СМО з «взаємодопомогою» між каналами середня кількість заявок під обслуговуванням не дорівнює середній кількості зайнятих каналів як у звичайних СМО:

$$\overline{l_{\text{обсл}}} = \overline{l_{\text{СМО}}} = \frac{\rho}{\rho + n}.$$

Середній час обслуговування заявки $\overline{t_{\text{обсл}}}$ і середній час перебування заявки в системі $\overline{t_{\text{с}}}$:

$$\overline{t_{\text{обсл}}} = \overline{t_{\text{с}}} = \frac{\overline{l_{\text{СМО}}}}{\lambda} = \frac{\rho}{\lambda(\rho + n)} = \frac{1}{\lambda + n\mu}.$$

Приклад. У центрі 3D-моделювання та візуалізації працює п’ять високопродуктивних графічних станцій. До центру надходять замовлення на візуалізацію складних інженерних моделей. У середньому надходить три замовлення за 12-годинний робочий день. У середньому повна обробка одного замовлення триває 12 годин. Центр приймає не більше одного замовлення за раз. Якщо всі станції зайняті – нове замовлення не приймається. Потрібно визначити: ймовірність того, що замовлення буде відхилене (всі ресурси зайняті); ймовірність обслуговування замовлення; середню кількість замовлень, що обробляються; середній час обслуговування

прийнятого замовлення; середній час обслуговування на одне замовлення з урахуванням і відхилених, і прийнятих.

Порівняти отримані параметри з варіантом використання багатоканальної СМО з відмовами (одне замовлення виконується тільки на одній графічній станції).

Розв'язання:

$$\lambda = \frac{3}{12} = 0.25 \text{ заявок/год.}, \qquad \mu = \frac{1}{12} = 0.083 \text{ заявок/год.},$$

$$n = 5, \qquad \rho = \frac{\lambda}{\mu} \approx 3.012.$$

$$p_0 = p_{\text{обсл}} = Q = \frac{n}{\rho + n} = \frac{5}{3.012 + 5} \approx 0.624,$$

$$p_1 = p_{\text{відм}} = \frac{\rho}{\rho + n} = \frac{3.012}{3.012 + 5} \approx 0.376.$$

$$A = \lambda Q = 0.25 \cdot 0.624 = 0.156,$$

Тобто за робочий день можна обслужити близько двох заявок $(0.156 \cdot 12 = 1.872)$.

Середній час обробки однієї заявки всіма n каналами:

$$\frac{1}{n\mu} = \frac{1}{5 \cdot 0.083} = 2.41 \text{ год.}$$

$$\overline{t_{\text{обсл}}} = \overline{t_{\text{с}}} = \frac{1}{\lambda + n\mu} = \frac{1}{0.25 + 5 \cdot 0.083} \approx 1.504 \text{ год.}$$

Рішення знайдене. Для дослідження ефективності роботи СМО з відмовами і «взаємодопомогою» між каналами типу «всі як один» слід порівняти її характеристики з характеристиками багатоканальної СМО з відмовами з п'ятьма каналами (кожне замовлення обробляється тільки на одній графічній станції):

$$p_0 = \left(\sum_{i=0}^{n} \frac{\rho^i}{i!} \right)^{-1} = \left(1 + 3.012 + \frac{3.012^2}{2!} + \frac{3.012^3}{3!} + \frac{3.012^4}{4!} + \frac{3.012^5}{5!} \right) =$$

$$= 0.054,$$

$$p_{\text{відм}} = \frac{\rho^n}{n!} p_0 = \frac{3.012^5}{5!} 0.054 = 0.112,$$

$$Q = p_{\text{обсл}} = 1 - p_{\text{відм}} = 1 - 0.112 = 0.888,$$

$$A = \lambda \cdot Q = 0.25 \cdot 0.888 = 0.222,$$

$$\overline{t_{\text{обсл}}} = \bar{t}_{\text{с}} = \frac{\rho(1 - p_n)}{\lambda} = \frac{Q}{\mu} = \frac{0.888}{0.083} = 10.656 \text{ год.}$$

Висновок: таким чином, пропускна здатність у СМО з «взаємодопомогою» гірше ніж у багатоканальної СМО з відмовами (0.156 проти 0.222), а середній час обслуговування і перебування заявки в системі – кращі в сім разів (1.504 год. проти 10.656 год.).

СМО з необмеженою чергою і «взаємодопомогою» між каналами типу «всі як один» працює як одноканальна СМО з необмеженою чергою та інтенсивністю обробки в n разів вище за інтенсивність роботи одного каналу. Черга не обмежена по довжині, система має безліч станів. Граф системи наведено на рис. 2.19.

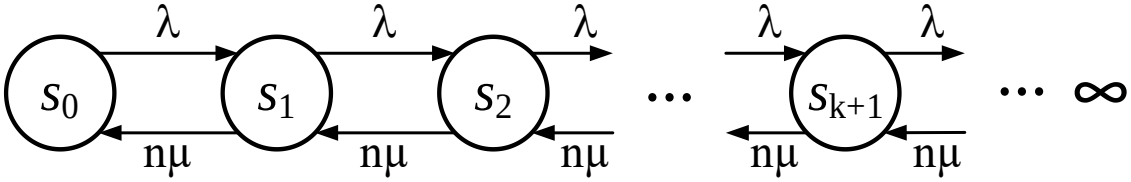


Рисунок 2.19 – Граф станів СМО з необмеженою чергою і «взаємодопомогою» між каналами типу «всі як один»

Стани системи:

s_0 – у системі немає заявок, усі n каналів вільні, черги немає;

s_1 – у системі одна заявка, усі n каналів зайняті її обслуговуванням, черги немає;

s_2 – у системі дві заявки, усі n каналів зайняті обслуговуванням однієї заявки, друга стоїть у черзі;

s_3 – у системі три заявки, усі n каналів зайняті обслуговуванням однієї заявки, дві інші стоять у черзі;

. . . . . ;

s_{k+1} – у системі $(k+1)$ заявок, усі n каналів зайняті обслуговуванням однієї заявки, інші k заявок стоять у черзі;

. . . . . .

Умова стаціонарного режиму (довжина черги не зростає до нескінченності):

$$\lambda < n\mu, \quad \text{або } \psi < 1.$$

Граничні ймовірності станів:

$$\begin{cases} p_0 = (1 + \psi + \psi^2 + \cdots + \psi^{k+1} + \cdots)^{-1} = \left(\dfrac{1}{1-\psi}\right)^{-1} = \dfrac{n-\rho}{n} \\ p_k = \psi^k p_0 = \left(\dfrac{\rho}{n}\right)^k \dfrac{n-\rho}{n}, (k = 1, 2, \dots) \end{cases}.$$

Як для всіх СМО з необмеженою чергою:

$$p_{\text{відм}} = 0, \qquad Q = p_{\text{обсл}} = 1 - p_{\text{відм}} = 1.$$

Абсолютна пропускна здатність:

$$A = \lambda \cdot Q = \lambda.$$

Інші показники:

$$\overline{l_{\text{обсл}}} = \psi, \qquad \overline{n_{\text{з}}} = n\psi = \rho, \qquad \overline{l_{\text{СМО}}} = \frac{\psi}{1-\psi}, \qquad \overline{l_{\text{оч}}} = \frac{\psi^2}{1-\psi},$$

$$p_{\text{ч}} = 1 - p_0 - p_1 = 1 - (1-\psi) - \psi(1-\psi) = \psi^2,$$

$$\overline{t_{\text{обсл}}} = \frac{\overline{l_{\text{обсл}}}}{\lambda} = \frac{\psi}{\lambda} = \frac{1}{n\mu}, \qquad \overline{t_{\text{оч}}} = \frac{\overline{l_{\text{оч}}}}{\lambda} = \frac{\psi^2}{\lambda(1-\psi)}, \qquad \overline{t_{\text{с}}} = \frac{\overline{l_{\text{СМО}}}}{\lambda} = \frac{\psi}{\lambda(1-\psi)}.$$

Приклад. У дата-центрі розгорнута бригадна система обробки великих пакетів архівованих даних, які надходять із зовнішніх систем резервного копіювання. Для цього використовується кластер із шести обчислювальних вузлів, які працюють спільно над кожним завданням, розподіляючи обсяг обробки між собою. Один пакет даних надходить у середньому раз на годину. Якщо кластер вільний, він одразу починає обробку. Якщо ж у момент надходження нового пакета всі вузли вже задіяні, пакет ставиться в чергу очікування. Середній час, який один вузол витрачає на обробку одного пакета – 4 години. Але оскільки всі шість вузлів працюють паралельно над

одним пакетом, повна обробка одного пакета займає в середньому $4/6 \approx 0.667$ годин. Необхідно оцінити ефективність такої організації обробки з точки зору часу обробки одного пакета та ефективність організації з точки зору очікування пакетів у черзі. Порівняти отримані характеристики з характеристиками СМО без «взаємодопомоги» каналів.

Розв'язання. Параметри системи:

- кількість каналів обслуговування $n = 6$;
- інтенсивність вхідного потоку заявок $\lambda = 1$ пакет/год.;
- середній час обслуговування заявки одним каналом 4 год.;
- інтенсивність потоку обслуговування одним каналом $\mu = \frac{1}{4} = 0.25$ пакетів/год.

$$\psi = \frac{\lambda}{n\mu} = \frac{1}{6 \cdot 0.25} \approx 0.667, \qquad \overline{l_{\text{обсл}}} = \psi = 0.667, \qquad \overline{t_{\text{обсл}}} = \frac{\psi}{\lambda} = 0.667,$$

$$\overline{l_{\text{оч}}} = \frac{\psi^2}{1-\psi} = \frac{0.667^2}{1-0.667} = 1.334, \qquad \overline{t_{\text{оч}}} = \frac{\overline{l_{\text{оч}}}}{\lambda} = 1.334.$$

Розрахунок аналогічних характеристик для багатоканальної СМО з необмеженою чергою без «взаємодопомоги» каналів:

$$\rho = \frac{\lambda}{\mu} = \frac{1}{0.25} = 4, \qquad p_0 = \left(\sum_{i=0}^{n} \frac{\rho^i}{i!} + \frac{\rho^{n+1}}{n! \cdot (n-\rho)} \right)^{-1} =$$

$$= \left(1 + 4 + \frac{4^2}{2!} + \frac{4^3}{3!} + \frac{4^4}{4!} + \frac{4^5}{5!} + \frac{4^6}{6!} + \frac{4^7}{6! \cdot (6-4)} \right)^{-1} \approx 0.017,$$

$$\overline{l_{\text{оч}}} = \frac{n\rho^{n+1}}{n! \cdot (n-\rho)^2} p_0 = \frac{6 \cdot 4^7}{6! \cdot (6-4)^2} p_0 \approx 0.58, \qquad \overline{t_{\text{оч}}} = \frac{\overline{l_{\text{оч}}}}{\lambda} = \frac{0.58}{1} = 0.58,$$

$$\overline{l_{\text{обсл}}} = \overline{n_{\text{з}}} = \rho = \frac{\lambda}{\mu} = \frac{1}{0.25} = 4, \qquad \overline{t_{\text{обсл}}} = \frac{\overline{l_{\text{обсл}}}}{\lambda} = \frac{4}{1} = 4.$$

Висновок: порівнюючи отримані характеристики можна побачити, що з точки зору часу очікування ефективнішою є система без «взаємодопомоги» каналів (швидше приблизно в $1.334/0.58 \approx 2.3$ рази), а з точки зору часу, який витрачається на обробку одного пакету, СМО з «взаємодопомогою»

більш ефективна: середній час обслуговування заявки менше в $4/0.667 \approx 6$ разів по порівнянню з СМО без «взаємодопомоги» каналів.

СМО з обмеженою чергою і «взаємодопомогою» між каналами типу «всі як один» має $n \geq 2$ каналів обслуговування та допускає наявність обмеженої черги, довжиною не більше $m \geq 1$ заявок. Граф системи наведено на рис. 2.20.

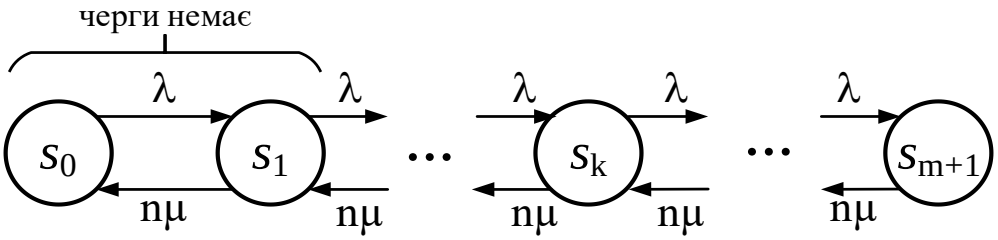


Рисунок 2.20 – Граф станів СМО з обмеженою чергою і «взаємодопомогою» між каналами типу «всі як один»

Стани нумеруються за кількістю заявок у системі:

s_0 – у системі немає заявок, всі n каналів вільні, черги немає;

s_1 – у системі одна заявка, всі n каналів зайняті її обслуговуванням, черги немає;

s_2 – у системі дві заявки, всі n каналів зайняті обслуговуванням однієї заявки, друга стоїть у черзі;

s_3 – у системі три заявки, всі n каналів зайняті обслуговуванням однієї заявки, дві інші стоять у черзі;

. . . . . ;

s_{m+1} – у системі $(m+1)$ заявок, всі n каналів зайняті обслуговуванням однієї заявки, інші m заявок стоять у черзі, заявка, що надходить у цей момент, отримує відмову.

Показники ефективності роботи розраховуються для двох випадків: $\psi \neq 1$ та $\psi = 1$.

Загальний випадок ($\psi \neq 1$).

Граничні ймовірності станів системи:

$$\begin{cases} p_0 = \dfrac{1-\psi}{1-\psi^{m+2}} \\ p_k = \psi^k p_0 = \psi^k \dfrac{1-\psi}{1-\psi^{m+2}}, (k = 1, 2, \dots, m+1) \end{cases}.$$

Ймовірність відмови заявці в обслуговуванні:

$$p_{\text{відм}} = \psi^{m+1} \frac{1-\psi}{1-\psi^{m+2}}.$$

Ймовірність наявності черги:

$$p_{\text{ч}} = p_2 + \cdots + p_{m+1} = (\psi^2 + \cdots + \psi^{m+1}) p_0 = \psi^2 \frac{1-\psi^m}{1-\psi} p_0 = \psi^2 \frac{1-\psi^m}{1-\psi^{m+2}}.$$

Відносна Q та абсолютна A пропускна здатність:

$$Q = \frac{1-\psi^{m+1}}{1-\psi^{m+2}}, \qquad A = \lambda Q = \lambda \frac{1-\psi^{m+1}}{1-\psi^{m+2}}.$$

Середня кількість заявок під обслуговуванням:

$$\overline{l_{\text{обсл}}} = \psi Q = \psi \frac{1-\psi^{m+1}}{1-\psi^{m+2}}.$$

Середня кількість заявок у черзі:

$$\overline{l_{\text{оч}}} = \psi^2 \frac{1-\psi^m (m - m\psi + 1)}{(1-\psi^{m+2})(1-\psi)}.$$

Середня кількість заявок у системі:

$$\overline{l_{\text{СМО}}} = \overline{l_{\text{оч}}} + \overline{l_{\text{обсл}}}.$$

Окремий випадок ($\psi = 1$).

Ймовірність простоювання системи:

$$p_0 = \frac{1}{m+2},$$

а ймовірності всіх інших без винятку станів системи рівні одна одній та рівні ймовірності вільного стану системи p_0:

$$p_0 = p_1 = p_2 = \cdots = p_{m+1} = \frac{1}{m+2}.$$

Інші показники:

$$p_{\text{відм}} = \frac{1}{m+2}, \qquad p_{\text{ч}} = p_2 + \cdots + p_{m+1} = \frac{m}{m+2},$$

$$Q = \frac{m+1}{m+2}, \qquad A = \lambda Q = \lambda \frac{m+1}{m+2},$$

$$\overline{l_{\text{обсл}}} = \psi Q = \frac{m+1}{m+2}, \qquad \overline{l_{\text{оч}}} = \sum_{i=1}^{m} i p_{i+1} = \frac{m(m+1)}{2(m+2)},$$

$$\overline{l_{\text{СМО}}} = \frac{m+1}{2}.$$

Кількість зайнятих каналів як і для всіх систем зі «взаємодопомогою» в n разів більше, ніж кількість заявок під обслуговуванням:

$$\overline{n_{\text{з}}} = n\overline{l_{\text{обсл}}} = \rho Q.$$

Показники середнього часу:

$$\overline{t_{\text{оч}}} = \frac{\overline{l_{\text{оч}}}}{\lambda}, \qquad \overline{t_{\text{обсл}}} = \frac{\overline{l_{\text{обсл}}}}{\lambda}, \qquad \overline{t_{\text{с}}} = \frac{\overline{l_{\text{СМО}}}}{\lambda}.$$

Приклад. У центрі обслуговування користувацьких IT-запитів, що працює в режимі онлайн, реалізовано систему для обробки звернень від клієнтів. Загальна кількість активних сесій, які система може одночасно обробляти або поставити в очікування, обмежена – не більше 11 користувачів одночасно. Один спеціаліст самостійно міг би обробити один запит за 20 хвилин. У середньому надходить 6 нових запитів на годину. Кожен запит потребує технічної обробки: діагностики проблеми, запуску скриптів, взаємодії з базою даних тощо. Для цього використовується система з трьома фахівцями технічної підтримки, які працюють кооперативно – над кожним запитом спільно. Якщо надходить новий запит і всі 11 сесій (обслуговування і черга) зайняті, клієнт отримує відмову. Необхідно визначити пропускну здатність системи і оцінити ефективність системи з точки зору середнього часу очікування користувача в черзі до початку обслуговування.

Розв'язання. Параметри системи:

- кількість каналів обслуговування $n = 3$;
- максимальна кількість користувачів 11 (1 обслуговується, m =10 – у черзі);
- інтенсивність вхідного потоку заявок $\lambda = 6$ користувачів/год.;

- середній час обслуговування заявки одним каналом 20 хв $= 1/3$ год;
- інтенсивність потоку обслуговування одним каналом $\mu = \frac{1}{1/3} = 3$ заявки/год.

$$\psi = \frac{\lambda}{n\mu} = \frac{6}{3 \cdot 3} \approx 0.667 \neq 1,$$

тобто розрахунки ведуться за формулами загального випадку:

$$p_0 = \frac{1-\psi}{1-\psi^{m+2}} = \frac{1-0.667}{1-0.667^{12}} \approx 0.336,$$

$$p_{\text{відм}} = \psi^{m+1}\frac{1-\psi}{1-\psi^{m+2}} = 0.667^{11}\frac{1-0.667}{1-0.667^{12}} \approx 0{,}004,$$

$$Q = \frac{1-\psi^{m+1}}{1-\psi^{m+2}} = \frac{1-0.667^{11}}{1-0.667^{12}} \approx 0.996, \qquad A = \lambda Q = 6 \cdot 0.996 \approx 5.976.$$

Отримане значення відносної пропускної здатності говорить про те, що з 1000 користувачів лише 4 отримують відмову.

$$\overline{l_{\text{оч}}} = \psi^2\frac{1-\psi^m(m-m\psi+1)}{(1-\psi^{m+2})(1-\psi)} = 0.667^2\frac{1-0.667^{10}(10-10\cdot 0.667+1)}{(1-0.667^{12})(1-0.667)} \approx$$

$$\approx 1.245,$$

$$\overline{t_{\text{оч}}} = \frac{\overline{l_{\text{оч}}}}{\lambda} = \frac{1.245}{6} = 0.2075 \text{ год.} = 12.45 \text{ хв},$$

$$p_{\text{ч}} = \psi^2\frac{1-\psi^m}{1-\psi^{m+2}} = 0.667^2\frac{1-0.667^{10}}{1-0.667^{12}} \approx 0.441,$$

$$\overline{l_{\text{обсл}}} = \psi Q = 0.667 \cdot 0.996 \approx 0.664,$$

$$\overline{t_{\text{обсл}}} = \frac{\overline{l_{\text{обсл}}}}{\lambda} = \frac{0.664}{6} \approx 0.1107 \text{ год.} \approx 6.64 \text{ хв},$$

$$\overline{n_{\text{з}}} = n\overline{l_{\text{обсл}}} = 3 \cdot 0.664 = 1.992,$$

$$\overline{l_{\text{СМО}}} = \overline{l_{\text{оч}}} + \overline{l_{\text{обсл}}} = 1.245 + 0.664 = 1.909,$$

$$\overline{t_{\text{с}}} = \frac{\overline{l_{\text{СМО}}}}{\lambda} = \frac{1.909}{6} \approx 0.3182 \text{ год.} \approx 19.09 \text{ хв}.$$

Висновок: пропускна здатність центра обслуговування – 99.6 % користувачів будуть обслуговані, час очікування початку обслуговування

приблизно 12 хвилин, а на саме обслуговування витрачається приблизно 7 хвилин.

СМО з відмовами і рівномірною «взаємодопомогою» між каналами працює таким чином: якщо заявка надходить у СМО, і всі канали вільні, то всі n каналів приступають до її обслуговування; якщо наступна заявка приходить у момент, коли всі канали зайняті обслуговуванням попередньої заявки, то частина каналів перемикається на її обслуговування; якщо надходить третя заявка, то знову частина каналів переключається на її обслуговування, і так доти, доки кожна заявка не опиниться під обслуговуванням лише одного каналу (кількість заявок = кількості каналів). Заявка, яка надійшла в момент, коли кожен канал зайнятий обслуговуванням однієї заявки, отримує відмову та залишає СМО.

Граф системи наведено на рис. 2.21.

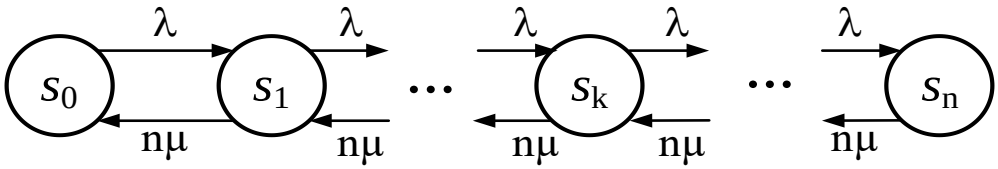


Рисунок 2.21 – Граф станів СМО з відмовами з рівномірною «взаємодопомогою» між каналами

Стани системи:

s_0 – у системі немає заявок, всі n каналів вільні, черги немає;

s_1 – у системі одна заявка, всі n каналів зайняті її обслуговуванням, черги немає;

s_2 – у системі дві заявки, всі n каналів зайняті обслуговуванням двох заявок, черги немає;

. . . . . ;

s_k – у системі k заявок, всі n каналів зайняті їх обслуговуванням, черги немає;

. . . . . ;

s_n – n каналів обслуговують n заявок, кожен канал обслуговує одну заявку, заявка, що надходить у цей момент, отримує відмову.

Система працює як одноканальна СМО з обмеженою чергою зі збільшеною інтенсивністю обслуговування $\varphi(n) = n\mu$ та числом місць у черзі $m = n - 1$.

Так само, як і для попередніх випадків, вводиться позначення:

$$\psi = \frac{\rho}{n}.$$

Загальний випадок ($\psi \neq 1$).

Граничні ймовірностей станів:

$$\begin{cases} p_0 = \dfrac{1-\psi}{1-\psi^{n+1}}, \psi \neq 1 \\ p_k = \psi^k p_0, (k = 1, 2, \dots, n) \end{cases}.$$

Ймовірність відмови:

$$p_{\text{відм}} = \psi^n \cdot p_0 = \psi^n \frac{1-\psi}{1-\psi^{n+1}},$$

відносна пропускна здатність:

$$Q = 1 - p_{\text{відм}} = \frac{1-\psi^n}{1-\psi^{n+1}},$$

абсолютна пропускна здатність:

$$A = \lambda \cdot Q = \lambda \frac{1-\psi^n}{1-\psi^{n+1}}.$$

Середнє число заявок, які знаходяться на обслуговуванні:

$$\overline{l_{\text{обсл}}} = \sum_{k=1}^{n} k\psi^k p_0 = \psi p_0 \left(\frac{\psi(1-\psi^n)}{1-\psi} \right)',$$

і, після знаходження похідної:

$$\overline{l_{\text{обсл}}} = \psi \frac{1-\psi^n(n-n\psi+1)}{(1-\psi)(1-\psi^{n+1})}.$$

Черги немає, тому

$$\overline{l_{\text{СМО}}} = \overline{l_{\text{обсл}}}.$$

Середня кількість зайнятих каналів:

$$\overline{n_з} = 0 \cdot p_0 + n \cdot (1 - p_0) = n \cdot (1 - p_0).$$

Середній час обслуговування і середній час перебування заявки в системі однакові:

$$\overline{t_{обсл}} = \frac{\overline{l_{обсл}}}{\lambda} = \overline{t_с} = \frac{\overline{l_{СМО}}}{\lambda}.$$

Окремий випадок ($\psi = 1$).

Граничні ймовірності:

$$p_0 = p_1 = p_2 = \cdots = p_n = \frac{1}{n+1}.$$

Ймовірність відмови:

$$p_{відм} = p_n = p_0 = \frac{1}{n+1}.$$

Відносна і абсолютна пропускна здатність:

$$Q = 1 - p_{відм} = \frac{n}{n+1}, \qquad A = \lambda \cdot Q = \lambda \frac{n}{n+1}.$$

Середня кількість зайнятих каналів:

$$\overline{n_з} = 0 \cdot p_0 + n \cdot (1 - p_0) = n \cdot (1 - p_0) = \frac{n^2}{n+1}.$$

Середнє число заявок, які знаходяться на обслуговуванні, та середнє число заявок у системі:

$$\overline{l_{СМО}} = \overline{l_{обсл}} = \frac{n}{2}.$$

Середній час обслуговування заявки співпадає з середнім часом перебування заявки в системі:

$$\overline{t_{обсл}} = \frac{\overline{l_{обсл}}}{\lambda} = \frac{n}{2\lambda} = \overline{t_с}.$$

Приклад. У центр оперативної обробки запитів до бази даних надходять запити від користувачів зі швидкістю $\lambda = 4$ запити/хв. Центр має три незалежних сервери, які можуть обслуговувати ці запити. Середній час обробки одного запиту на одному сервері – 0.5 хв, тобто $\mu = 2$ запити/хв на один сервер. Необхідно порівняти основні характеристики СМО для трьох варіантів:

а) без взаємодопомоги (звичайна 3-канальна СМО з відмовами);

б) «взаємодопомога» типу «всі як один»;

в) рівномірна «взаємодопомога».

Розв'язання:

а) Багатоканальна СМО з відмовами.

$$\rho = \frac{\lambda}{\mu} = \frac{4}{2} = 2,$$

$$p_0 = \left(\sum_{i=0}^{3} \frac{\rho^i}{i!}\right)^{-1} = \left(1 + \rho + \frac{\rho^2}{2!} + \frac{\rho^3}{3!}\right)^{-1} = \left(1 + 2 + \frac{4}{2} + \frac{8}{6}\right)^{-1} \approx 0.158,$$

$$p_{\text{відм}} = p_n = \frac{\rho^n}{n!} p_0 = \frac{\rho^3}{3!} p_0 = \frac{2^3}{6} 0.158 \approx 0.211,$$

$$Q = p_{\text{обсл}} = 1 - 0.211 = 0.789,$$

$$A = \lambda \cdot Q = \lambda \cdot p_{\text{обсл}} = 4 \cdot 0.789 = 3.156,$$

$$\overline{n_з} = \rho \cdot Q = 2 \cdot 0.789 = 1.578,$$

$$\overline{l_{\text{обсл}}} = \overline{l_{\text{СМО}}} = \overline{n_з} = 1.578,$$

$$\overline{t_{\text{обсл}}} = \overline{t_с} = \frac{Q}{\mu} = \frac{0.789}{2} = 0.3945 \text{ хв.}$$

б) СМО з відмовами і «взаємодопомогою» між каналами типу «всі як один».

$$p_0 = p_{\text{обсл}} = \frac{n}{\rho + n} = \frac{3}{2 + 3} = 0.6,$$

$$p_{\text{відм}} = \frac{\rho}{\rho + n} = \frac{2}{2 + 3} = 0.4,$$

$$Q = p_{\text{обсл}} = 0.6,$$

$$A = \lambda \cdot Q = 4 \cdot 0.6 = 2.4,$$

$$\overline{n_з} = \rho \cdot Q = 2 \cdot 0.6 = 1.2,$$

$$\overline{l_{\text{обсл}}} = \overline{l_{\text{СМО}}} = \frac{\rho}{\rho + n} = \frac{2}{2 + 3} = 0.4,$$

$$\overline{t_{\text{обсл}}} = \overline{t_с} = \frac{\overline{l_{\text{СМО}}}}{\lambda} = \frac{0.4}{4} = 0.1 \text{ хв.}$$

в) СМО з відмовами з рівномірною «взаємодопомогою» між каналами.

$$\psi = \frac{\rho}{n} = \frac{2}{3},$$

$$p_0 = \frac{1-\psi}{1-\psi^{n+1}} = \frac{1-\left({}^{2}/_{3}\right)}{1-\left({}^{2}/_{3}\right)^{4}} \approx 0.415,$$

$$p_{\text{відм}} = \psi^n \cdot p_0 = \left({}^{2}/_{3}\right)^{3} \cdot 0.415 \approx 0.123,$$

$$Q = 1 - p_{\text{відм}} = 1 - 0.123 = 0.877,$$

$$A = \lambda \cdot Q = 4 \cdot 0.877 = 3.508,$$

$$\overline{n_з} = n \cdot (1 - p_0) = 3 \cdot (1 - 0.415) \approx 1.755,$$

$$\overline{l_{\text{обсл}}} = \overline{l_{\text{СМО}}} = \psi \frac{1-\psi^n(n-n\psi+1)}{(1-\psi)(1-\psi^{n+1})} = \frac{2}{3} \cdot \frac{1-\left({}^{2}/_{3}\right)^{3}\left(3-3\cdot\left({}^{2}/_{3}\right)+1\right)}{\left(1-\left({}^{2}/_{3}\right)\right)\left(1-\left({}^{2}/_{3}\right)^{4}\right)} \approx$$

$$\approx 1.015,$$

$$\overline{t_{\text{обсл}}} = \overline{t_{\text{с}}} = \frac{\overline{l_{\text{обсл}}}}{\lambda} = \frac{1.015}{4} = 0.254 \text{ хв.}$$

Висновок: порівняння з багатоканальною СМО з відмовами показує, що при «взаємодопомозі» типу «всі як один» зменшуються пропускна здатність та кількість зайнятих каналів, при чому середній час обслуговування заявки зменшується майже в 4 рази. Водночас зростає ймовірність відмови та простоювання всіх каналів. Через надмірну інтенсивність обслуговування частина заявок втрачається, що знижує загальну ефективність системи. Натомість, при рівномірній «взаємодопомозі» знижується ймовірність відмови, зростає пропускна здатність і завантаження каналів. Час обслуговування зменшується незначним чином, що забезпечує більш збалансовану продуктивність системи.

СМО з обмеженою чергою і рівномірною «взаємодопомогою» між каналами має розмічений граф, наведений на рис. 2.22.

СМО працює як одноканальна система з обмеженою чергою зі збільшеною інтенсивністю обслуговування $\varphi(n) = n\mu$ і кількістю місць у черзі $n + m - 1$.

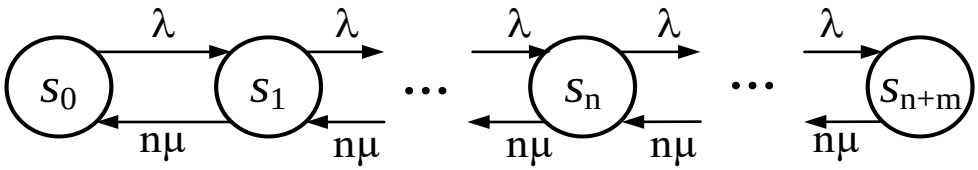


Рисунок 2.22 – Граф станів СМО з обмеженою чергою і рівномірною «взаємодопомогою» між каналами

Стани системи:

s_0 – у системі немає заявок, усі n каналів вільні, черги немає;

s_1 – у системі одна заявка, усі n каналів зайняті її обслуговуванням, черги немає;

s_2 – у системі дві заявки, усі n каналів зайняті обслуговуванням двох заявок, черги немає;

. . . . . ;

s_n – n каналів обслуговують n заявок, кожен канал обслуговує одну заявку, черги немає;

s_{n+1} – n каналів обслуговують n заявок, кожен канал обслуговує одну заявку, у черзі одна заявка;

. . . . . ;

s_{n+m} – n каналів обслуговують n заявок, кожен канал обслуговує одну заявку, у черзі m заявок, заявка, що надходить у цей момент, отримує відмову.

Загальний випадок ($\psi \neq 1$).

Граничні ймовірності:

$$\begin{cases} p_0 = \dfrac{1-\psi}{1-\psi^{n+m+1}}, \psi \neq 1 \\ p_k = \psi^k p_0, (k = 1, 2, \dots, n+m) \end{cases}.$$

Ймовірність відмови:

$$p_{\text{відм}} = \psi^{n+m} p_0 = \psi^{n+m} \frac{1-\psi}{1-\psi^{n+m+1}}.$$

Ймовірність утворення черги:

$$p_{\text{ч}} = \psi^{n+1}\frac{1-\psi^m}{1-\psi}p_0 = \psi^{n+1}\frac{1-\psi^m}{1-\psi^{n+m+1}}.$$

Відносна пропускна здатність:

$$Q = 1 - \psi^{n+m}\cdot p_0 = \frac{1-\psi^{n+m}}{1-\psi^{n+m+1}}.$$

Абсолютна пропускна здатність:

$$A = \lambda \cdot Q = \lambda\frac{1-\psi^{n+m}}{1-\psi^{n+m+1}}.$$

Середнє число заявок, які знаходяться на обслуговуванні:

$$\overline{l_{\text{обсл}}} = \psi\frac{1-\psi^n(n-n\psi+1)}{(1-\psi)(1-\psi^{n+m+1})}.$$

Середня кількість заявок $\overline{l_{\text{оч}}}$, що знаходяться у черзі:

$$\overline{l_{\text{оч}}} = \psi^{n+1}\frac{1-\psi^m(m-m\psi+1)}{(1-\psi)(1-\psi^{n+m+1})}.$$

Середнє число заявок у системі:

$$\overline{l_{\text{СМО}}} = \overline{l_{\text{обсл}}} + \overline{l_{\text{оч}}}.$$

Середня кількість зайнятих каналів:

$$\overline{n_{\text{з}}} = 0\cdot p_0 + n\cdot(1-p_0) = n\cdot(1-p_0).$$

Показники середнього часу:

$$\overline{t_{\text{обсл}}} = \frac{\overline{l_{\text{обсл}}}}{\lambda}, \qquad \overline{t_{\text{оч}}} = \frac{\overline{l_{\text{оч}}}}{\lambda}, \qquad \overline{t_{\text{с}}} = \frac{\overline{l_{\text{СМО}}}}{\lambda}.$$

Окремий випадок ($\psi = 1$).

Граничні ймовірності:

$$\begin{cases} p_0 = \dfrac{1}{n+m+1}, \psi = 1 \\ p_k = p_0, (k = 1, 2, \dots, n+m) \end{cases}.$$

Ймовірність відмови:

$$p_{\text{відм}} = p_{n+m} = p_0 = \frac{1}{n+m+1}.$$

Ймовірність утворення черги:

$$p_{\text{ч}} = mp_0 = \psi^{n+1}\frac{m}{n+m+1}.$$

Відносна пропускна здатність:

$$Q = 1 - p_{\text{відм}} = \frac{n+m}{n+m+1}.$$

Абсолютна пропускна здатність:

$$A = \lambda \cdot Q = \lambda \frac{n+m}{n+m+1}.$$

Середня кількість зайнятих каналів:

$$\overline{n_з} = n \cdot (1 - p_0) = \frac{n(n+m)}{n+m+1}.$$

Середнє число заявок, які знаходяться на обслуговуванні:

$$\overline{l_{\text{обсл}}} = \sum_{k=1}^{n} k p_k = p_0(1 + 2 + \cdots + n) = \frac{n(n+1)}{2} p_0 = \frac{n(n+1)}{2(n+m+1)}.$$

Середня кількість заявок $\overline{l_{\text{оч}}}$, що знаходяться в черзі:

$$\overline{l_{\text{оч}}} = \sum_{k=1}^{m} k p_{n+k} = p_0(1 + 2 + \cdots + m) = \frac{m(m+1)}{2} p_0 = \frac{m(m+1)}{2(n+m+1)}.$$

Середнє число заявок у системі:

$$\overline{l_{\text{СМО}}} = \overline{l_{\text{обсл}}} + \overline{l_{\text{оч}}} = \frac{n(n+1) + m(m+1)}{2(n+m+1)}.$$

Показники середнього часу:

$$\overline{t_{\text{обсл}}} = \frac{\overline{l_{\text{обсл}}}}{\lambda} = \frac{n(n+1)}{2\lambda(n+m+1)}, \qquad \overline{t_{\text{оч}}} = \frac{\overline{l_{\text{оч}}}}{\lambda} = \frac{m(m+1)}{2\lambda(n+m+1)},$$

$$\overline{t_{\text{с}}} = \frac{\overline{l_{\text{СМО}}}}{\lambda} = \frac{n(n+1) + m(m+1)}{2\lambda(n+m+1)}.$$

Приклад. У хмарному дата-центрі працює три віртуальні сервери, які обробляють запити клієнтів у реальному часі. Система налаштована на прийом максимум до 5 одночасних запитів: три з них можуть оброблятися паралельно, ще дві заявки можуть перебувати в черзі очікування. Якщо в системі вже 5 заявок (обслуговуються або очікують) – нові заявки відхиляються. Запити надходять у середньому з інтенсивністю $\lambda = 4$ запити/хв, а середній час обробки одного запиту на одному сервері – 0.5 хв ($\mu = 2$ запити/хв на один сервер). Визначити основні характеристики СМО.

Розв'язання.

$$n = 3, \qquad m = 2, \qquad \lambda = 4, \qquad \mu = 2, \qquad \rho = \frac{\lambda}{\mu} = 2, \qquad \psi = \frac{\rho}{n} = \frac{2}{3},$$

Загальний випадок ($\psi \neq 1$).

$$p_0 = \frac{1-\psi}{1-\psi^{n+m+1}} = \frac{1-{}^2/_3}{1-\left({}^2/_3\right)^6} \approx 0.365,$$

$$p_{\text{відм}} = \psi^{n+m} p_0 = \left({}^2/_3\right)^5 \cdot 0.36 \approx 0.048,$$

$$p_{\text{ч}} = \psi^{n+1}\frac{1-\psi^m}{1-\psi} p_0 = \left(\frac{2}{3}\right)^4 \cdot \frac{1-\left({}^2/_3\right)^2}{1-{}^2/_3} 0.365 \approx 0.12,$$

$$Q = \frac{1-\psi^{n+m}}{1-\psi^{n+m+1}} = \frac{1-\left({}^2/_3\right)^5}{1-\left({}^2/_3\right)^6} \approx 0.952,$$

$$A = \lambda \cdot Q = 4 \cdot 0.952 \approx 3.808,$$

$$\overline{l_{\text{обсл}}} = \psi\frac{1-\psi^n(n-n\psi+1)}{(1-\psi)(1-\psi^{n+m+1})} = \frac{2}{3} \cdot \frac{1-\left({}^2/_3\right)^3\left(3-3\left({}^2/_3\right)+1\right)}{\left(1-{}^2/_3\right)\left(1-\left({}^2/_3\right)^6\right)} \approx 0.893,$$

$$\overline{l_{\text{оч}}} = \psi^{n+1}\frac{1-\psi^m(m-m\psi+1)}{(1-\psi)(1-\psi^{n+m+1})} = \left(\frac{2}{3}\right)^4 \cdot \frac{1-\left({}^2/_3\right)^2\left(2-2\left({}^2/_3\right)+1\right)}{\left(1-{}^2/_3\right)\left(1-\left({}^2/_3\right)^6\right)} \approx$$

$$\approx 0.168,$$

$$\overline{l_{\text{СМО}}} = \overline{l_{\text{обсл}}} + \overline{l_{\text{оч}}} = 0.893 + 0.168 = 1.061,$$

$$\bar{n}_{\text{з}} = n \cdot (1 - p_0) = 3 \cdot (1 - 0.365) \approx 1.905,$$

$$\overline{t_{\text{обсл}}} = \frac{\overline{l_{\text{обсл}}}}{\lambda} = \frac{0.893}{\lambda} \approx 0.223 \text{ хв},$$

$$\overline{t_{\text{оч}}} = \frac{\overline{l_{\text{оч}}}}{\lambda} = \frac{0.168}{\lambda} \approx 0.042 \text{ хв},$$

$$\bar{t}_{\text{с}} = \frac{\overline{l_{\text{СМО}}}}{\lambda} = \frac{1.061}{\lambda} \approx 0.265 \text{ хв}.$$

Висновок: режим рівномірної «взаємодопомоги» між каналами забезпечує високу швидкість обробки заявки і високу продуктивність системи. Частка обслужених заявок дорівнює 95.2 %.

СМО з необмеженою чергою і рівномірною «взаємодопомогою» між каналами працює як одноканальна система з необмеженою чергою та збільшеною в n разів інтенсивністю обробки заявки. Для того щоб довжина черги не збільшувалася до нескінченності і система залишалася працездатною (щоб існували граничні ймовірності станів системи), необхідно виконання умови: $\psi < 1$.

Розмічений граф системи наведено на рис. 2.23.

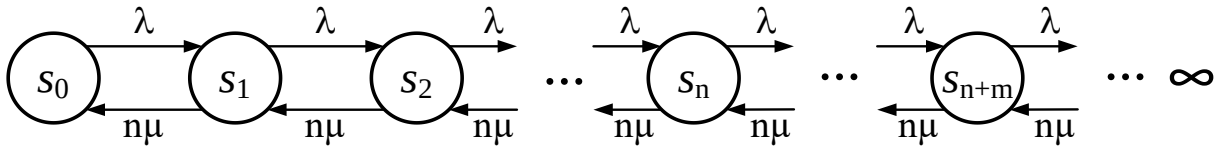


Рисунок 2.23 – Граф станів СМО з необмеженою чергою і рівномірною «взаємодопомогою» між каналами

Стани системи: особливістю системи є те, що черга починає в ній утворюватися не від стану s_2 як в одноканальній СМО, а починаючи від стану s_{n+1}; для станів $s_1, \dots, s_n$ черги немає, всі заявки перебувають під обслуговуванням.

Граничні ймовірності:

$$\begin{cases} p_0 = (1 + \psi + \psi^2 + \cdots)^{-1} = \left(\dfrac{1}{1-\psi}\right)^{-1} = 1 - \psi \\ p_k = \psi^k p_0 = \psi^k (1-\psi), (k = 1, 2, \dots) \end{cases}.$$

Оскільки відсутнє обмеження на довжину черги, то слушні вирази:

$$p_{\text{відм}} = 0, \qquad Q = p_{\text{обсл}} = 1, \qquad A = \lambda \cdot Q = \lambda.$$

Середнє число заявок, які знаходяться на обслуговуванні:

$$\overline{l_{\text{обсл}}} = \psi \frac{1 - \psi^n (n - n\psi + 1)}{(1-\psi)^2} p_0 = \psi \frac{1 - \psi^n (n - n\psi + 1)}{1 - \psi}.$$

Ймовірність наявності черги (стани СМО, починаючи зі стану s_{n+1}):

$$p_{\text{ч}} = \sum_{k=1}^{\infty} \psi^{n+k} p_0 = \psi^{n+1} p_0 (1 + \psi + \psi^2 + \psi^3 \dots) = \frac{\psi^{n+1}}{1-\psi} p_0 = \psi^{n+1}.$$

Середня кількість заявок $\overline{l_{\text{оч}}}$, що очікують у черзі:

$$\overline{l_{\text{оч}}} = \frac{\psi^{n+1}}{(1-\psi)^2} p_0 = \frac{\psi^{n+1}}{1-\psi}.$$

Середня кількість зайнятих каналів:

$$\overline{n_{\text{з}}} = n \cdot (1 - p_0) = n\psi = \rho.$$

Середня кількість заявок у системі:

$$\overline{l_{\text{СМО}}} = \overline{l_{\text{оч}}} + \overline{l_{\text{обсл}}} = \psi \frac{1 - n\psi^n(1-\psi)}{1-\psi}.$$

Середній час обслуговування заявки:

$$\overline{t_{\text{обсл}}} = \frac{\overline{l_{\text{обсл}}}}{\lambda} = \frac{\psi}{\lambda} \cdot \frac{1 - \psi^n(n - n\psi + 1)}{1-\psi}.$$

Середній час очікування обслуговування в черзі:

$$\overline{t_{\text{оч}}} = \frac{\overline{l_{\text{оч}}}}{\lambda} = \frac{\psi^{n+1}}{\lambda(1-\psi)}.$$

Середній час перебування заявки в системі:

$$\overline{t_{\text{с}}} = \overline{t_{\text{оч}}} + \overline{t_{\text{обсл}}} = \frac{\overline{l_{\text{СМО}}}}{\lambda} = \frac{\psi}{\lambda} \cdot \frac{1 - n\psi^n(1-\psi)}{1-\psi}.$$

Приклад. У центрі обробки даних працює три сервери, які обслуговують запити користувачів у режимі динамічного балансування навантаження. Кожен сервер може самостійно обслуговувати запит зі швидкістю $\mu = 2$ запити/хв. Усього запити надходять зі швидкістю $\lambda = 4$ запити/хв. Система допускає необмежену чергу очікування – усі заявки зберігаються до обробки, відмов немає. Визначити характеристики СМО.

Розв'язання.

$n = 3, \quad m = \infty, \quad \lambda = 4, \quad \mu = 2, \quad \rho = \frac{\lambda}{\mu} = 2, \quad \psi = \frac{\rho}{n} = \frac{2}{3},$

$p_0 = 1 - \psi = 1 - \frac{2}{3} = \frac{1}{3},$

$p_{\text{відм}} = 0,$

$Q = p_{\text{обсл}} = 1,$

$A = \lambda \cdot Q = 4,$

$$p_{\text{ч}} = \psi^{n+1} = \left(\frac{2}{3}\right)^4 \approx 0.198,$$

$$\overline{n_{\text{з}}} = \rho = 2,$$

$$\overline{l_{\text{обсл}}} = \psi \frac{1 - \psi^n(n - n\psi + 1)}{1 - \psi} = \frac{2}{3} \cdot \frac{1 - \left({}^2/_3\right)^3 \left(3 - 3\left({}^2/_3\right) + 1\right)}{1 - {}^2/_3} \approx 0.815,$$

$$\overline{l_{\text{оч}}} = \frac{\psi^{n+1}}{1 - \psi} = \frac{\left({}^2/_3\right)^4}{1 - {}^2/_3} \approx 0.593,$$

$$\overline{l_{\text{СМО}}} = \overline{l_{\text{оч}}} + \overline{l_{\text{обсл}}} = \overline{l_{\text{оч}}} + \overline{l_{\text{обсл}}} = 0.815 + 0.593 = 1.408,$$

$$\overline{t_{\text{обсл}}} = \frac{\overline{l_{\text{обсл}}}}{\lambda} = \frac{0.815}{4} \approx 0.204,$$

$$\overline{t_{\text{оч}}} = \frac{\overline{l_{\text{оч}}}}{\lambda} = \frac{0.593}{4} \approx 0.148,$$

$$\overline{t_{\text{с}}} = \overline{t_{\text{оч}}} + \overline{t_{\text{обсл}}} = 0.204 + 0.148 = 0.352.$$

2.3. Сутність і зміст методу динаміки середніх

Марковські випадкові процеси широко використовуються для математичного моделювання поведінки складних систем. У таких моделях роботу системи в часі розглядають як випадкову зміну станів. Якщо ймовірність переходу з одного стану в інший залежить лише від поточного стану і не залежить від того, що було раніше, можна застосовувати апарат марковських процесів.

Проте, зі збільшенням кількості елементів у системі стрімко зростає кількість можливих її станів, а отже, й обсяг обчислень – кількість рівнянь у моделі може досягати сотень тисяч або навіть більше (наприклад, 100^4). Навіть за наявності сучасних комп'ютерів, вирішення такого обсягу диференціальних чи алгебричних рівнянь є вкрай складним і практично нереальним. І навіть якщо обчислення вдасться провести, результати можуть виявитися надто складними для аналізу та ухвалення рішень.

Тому для систем із великою кількістю станів часто застосовують метод динаміки середніх. Він дозволяє визначати не самі ймовірності станів, а узагальнені характеристики процесу – наприклад, середню кількість елементів, які перебувають у певному стані. Це значно спрощує аналіз і робить моделі придатнішими для практичного використання.

У багатоелементних комп'ютерних системах (зокрема, у сфері інформаційної безпеки) ціллю моделювання часто є не відстеження стану кожного окремого елемента, а визначення середньої кількості компонентів у певному стані.

Наприклад, фахівця з безпеки цікавить загальна кількість сенсорів, які виявлені, заблоковані або нейтралізовані, а не конкретна ймовірність того, що один, два чи три сенсори одночасно перебувають у тому чи іншому стані. Такий підхід дозволяє оцінити загальний рівень стійкості системи, приймати стратегічні рішення щодо масштабування чи резервування, і визначати ймовірну ефективність протидії атакам без необхідності моделювати кожен сенсор окремо.

Однак, щоб визначити середню кількість елементів у кожному з можливих станів, необхідно знати ймовірності p_i всіх можливих конфігурацій системи, тому що $\overline{n} = \sum_{i=0}^{n} p_i n_i$. Проте, у реальних багатокомпонентних системах кількість можливих станів n стрімко зростає зі збільшенням числа елементів, а разом із нею – і кількість рівнянь Колмогорова, які потрібно розв'язати. Це може зробити моделювання на основі класичних марковських процесів практично неможливим, навіть за умови використання потужних обчислювальних засобів.

Наприклад, у розподіленій системі виявлення шкідливих програм компанія використовує 10 автономних сенсорів безпеки, розгорнутих у різних вузлах мережі. Кожен із сенсорів може перебувати в одному з чотирьох станів:

s_1 – сенсор справний, здійснює моніторинг, не виявлений зловмисником;

s_2 – сенсор справний і активний, але виявлений атакуючою стороною;

s_3 – сенсор не пошкоджений, але зазнає зовнішнього впливу, наприклад, DDoS-атаки або перевантаження, що заважає виконанню його функцій;

s_4 – сенсор виведений із ладу або дезактивований (нейтралізований) внаслідок атаки чи критичної помилки.

Для того щоб визначити середню кількість елементів у кожному зі станів, у випадку 10 компонентів, кожен з яких може перебувати в одному з 4 станів, потрібно було б скласти 4^{10} рівнянь Колмогорова. Очевидно, що такий обсяг обчислень є надмірним, і подібне моделювання стає практично непридатним для аналізу систем.

Метод динаміки середніх дає змогу безпосередньо оцінити очікувану кількість елементів (математичне сподівання), що перебувають у певному стані в складній системі. Це наближений метод, однак він має важливу властивість: що більша система – тим точнішими є результати моделювання.

Щоб вивести формули розрахунку, вважають, що система відповідає таким умовам:

- у ній відбувається випадковий марковський процес;
- усі елементи однорідні – мають однакові стани, їх кількість та ймовірності;
- кожен елемент змінює стан незалежно від інших.

Мета моделювання полягає в тому, щоб визначити:

- середню кількість (математичне сподівання) $m_i(t)$ елементів системи, які перебувають в однакових станах s_i;
- дисперсії $D_i(t)$, що характеризують коливання кількості елементів у цих станах із часом.

Тобто, моделювання дозволяє оцінити не лише типовий (середній) розподіл елементів по станах, але й ступінь їх розсіювання – наскільки сильно система може відхилятися від середнього в процесі функціонування.

Постановка задачі моделювання динаміки функціонування системи методом динаміки середніх: нехай система S складається з N елементів, кожен з яких у будь-який момент часу може перебувати в одному з можливих станів $s_1, s_2, \dots, s_n$. Вводиться позначення $x_j = x_j(t)$ – кількість елементів системи S, що перебувають у стані s_j у момент часу t, необхідно визначити основні невипадкові характеристики випадкової функції $x_j(t)$: математичне сподівання $m_j = m_j(t)$ і дисперсію $D_j = D_j(t)$. Схематично така система може бути представлена так, як наведено на рис. 2.24.

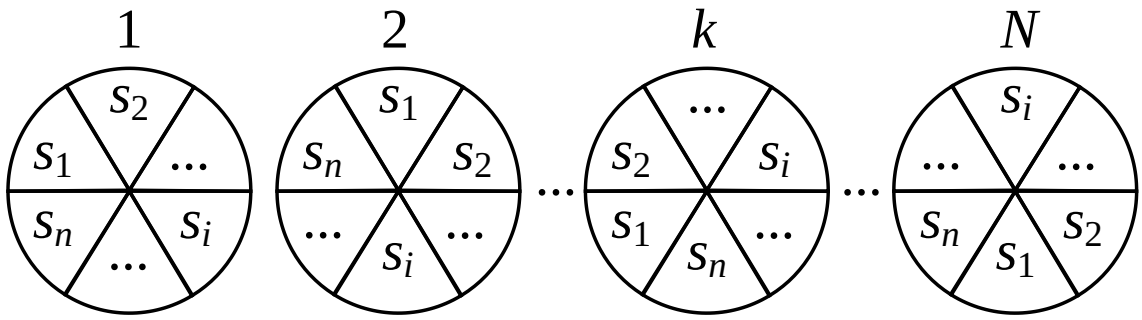


Рисунок 2.24 – Схематичне зображення системи

Щоб визначити необхідні характеристики випадкової функції $x_j(t)$, слід встановити залежність кількості елементів у стані s_j від стану кожного l-го елемента:

$$x_j = \sum_{l=1}^{N} x_j^l,$$

де $x_j^l = 1$, якщо l-й елемент перебуває в стані s_j, і $x_j^l = 0$ – в іншому випадку, тобто $P\{x_j^l = 1\} = p_j^l$. Оскільки всі елементи однорідні, ймовірність перебування будь-якого з них у стані s_j у будь-який момент часу t однакова:

$$p_j^l = p_j.$$

Очевидно, що: $M[x_j^l] = p_j$ і $D[x_j^l] = p_j(1 - p_j)$. Припускаючи незалежність випадкових величин x_j^l, можна отримати:

$$m_j = \sum_{l=1}^{N} M\left[x_j^l\right] = Np_j \quad \text{і} \quad D_j = \sum_{l=1}^{N} D\left[x_j^l\right] = Np_j\left(1 - p_j\right).$$

Звідси, зокрема, випливає, що для отримання рівнянь, за якими можна обчислити математичні сподівання $m_j(t)$, необхідно спочатку записати систему рівнянь Колмогорова для ймовірностей станів $p_j(t)$. Після цього слід помножити обидві частини кожного з рівнянь на N і виконати заміну виразів Np_j на m_j.

Рівність $m_j = Np_j$ пов'язує ймовірність j-го стану елемента в довільний момент часу з математичним сподіванням чисельності цих станів по всім елементам.

Система рівнянь Колмогорова для одного елемента містить n рівнянь, а для всіх N елементів – N^n, а метод динамики средних у n^{N-1} разів менше.

Порядок моделювання з використанням методу динаміки середніх полягає в такому.

1. Описати стани одного елемента системи.
2. Скласти розмічений граф станів для одного елемента, вказавши поруч із кожним станом $s_1, s_2, \dots, s_j, \dots, s_n$ середні чисельності $m_1, m_2, \dots, m_j, \dots, m_n$, що визначаються як $m_j = Np_j$.
3. Скласти диференціальні рівняння:
    - похідна середньої чисельності стану дорівнює сумі стількох членів, скільки стрілок пов'язано з цим станом;
    - якщо стрілка спрямована зі стану, член має знак мінус, якщо в стан – знак плюс;
    - кожен член дорівнює добутку інтенсивності потоку подій, що переводить елемент за цією стрілкою, на середню чисельність того стану, з якого виходить стрілка.
4. Розв'язати систему диференціальних рівнянь відносно m_j.

5. Обчислити значення дисперсій D_j і середніх квадратичних відхилень $\sigma_j = \sqrt{D_j}$.

Отримані рівняння для m_j називають рівняннями динаміки середніх. Система рівнянь динаміки середніх відносно m_j вирішується при таких початкових умовах:

$$m_1(0) = N, m_2(0) = m_3(0) = \cdots m_n(0) = 0.$$

Приклад. Побудувати математичну модель для визначення середньої кількості робочих станцій у різних технічних станах у межах корпоративного IT-підрозділу, якщо загальна кількість комп'ютерів $N = 100$, а кожен із них може перебувати в одному з таких станів:

s_1 – комп'ютер справний;

s_2 – виявлено збої, проводиться автоматична діагностика;

s_3 – збій локалізовано, комп'ютер передано на ручну перевірку технічним фахівцям;

s_4 – виконується ремонт на рівні внутрішнього IT-відділу;

s_5 – комп'ютер передано до виробника або сервісного центру на ремонт;

s_6 – комп'ютер списано, замінюється на новий пристрій.

При знаходженні комп'ютера в несправному стані (s_3) проводиться його діагностика, внаслідок чого він може бути відправлений на ремонт на рівні внутрішнього IT-відділу (s_4), до сервісного центру (s_5), або списаний (s_6). У випадку списання виконується заміна на новий комп'ютер.

Інтенсивності переходів між станами:

$\lambda_{12} = 0.08$; $\lambda_{23} = \lambda_{34} = 2$; $\lambda_{35} = \lambda_{36} = \lambda_{41} = 1$; $\lambda_{51} = 0.5$; $\lambda_{61} = 4$.

Розв'язання.

Розмічений граф станів системи наведено на рис. 2.25.

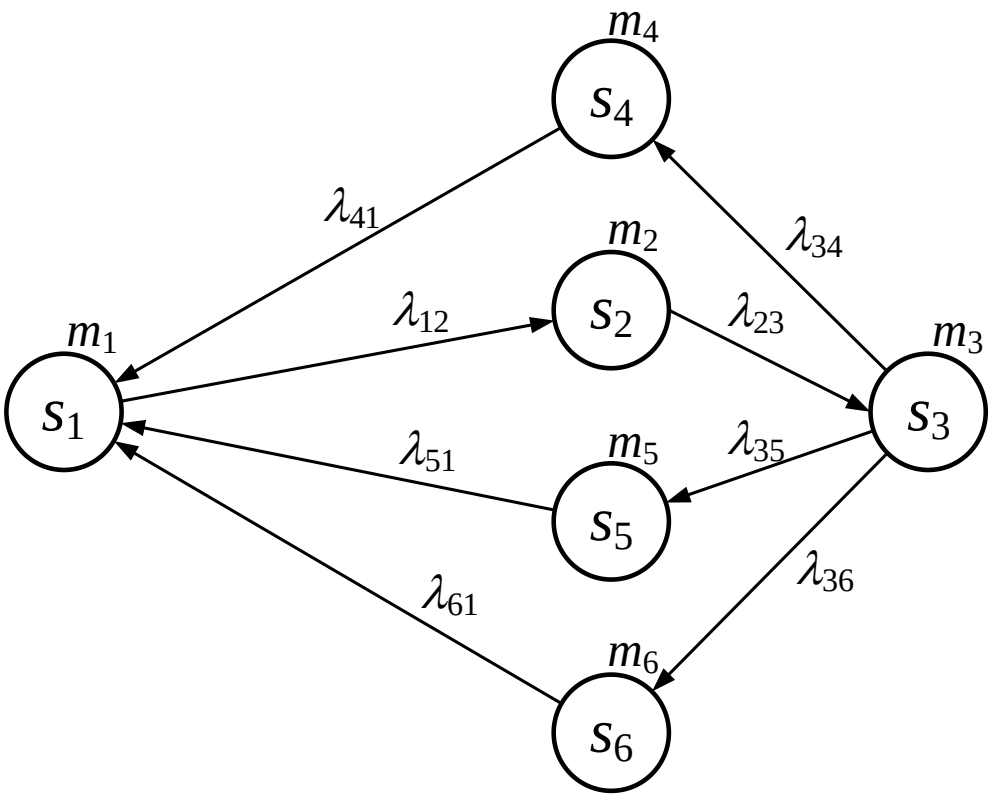


Рисунок 2.25 – Розмічений граф станів системи

Система рівнянь динаміки:

$$\begin{cases} \frac{dm_1}{dt} = -\lambda_{12}m_1 + \lambda_{41}m_4 + \lambda_{51}m_5 + \lambda_{61}m_6 \\ \frac{dm_2}{dt} = -\lambda_{23}m_2 + \lambda_{12}m_1 \\ \frac{dm_3}{dt} = -\lambda_{34}m_3 - \lambda_{35}m_3 - \lambda_{36}m_3 + \lambda_{23}m_2 \\ \frac{dm_4}{dt} = -\lambda_{41}m_4 + \lambda_{34}m_3 \\ \frac{dm_5}{dt} = -\lambda_{51}m_5 + \lambda_{35}m_3 \\ \frac{dm_6}{dt} = -\lambda_{61}m_6 + \lambda_{36}m_3 \end{cases}.$$

У наведеній системі чисельності m_j не є функціями часу. Тому можна скористуватись нормувальною умовою.

$$m_1 + m_2 + m_3 + m_4 + m_5 + m_6 = N = 100$$

для будь-якого значення t.

Система лінійних алгебричних рівнянь для знаходження граничних ймовірностей:

$$\begin{cases} 0 = -\lambda_{12}m_1 + \lambda_{41}m_4 + \lambda_{51}m_5 + \lambda_{61}m_6 \\ 0 = -\lambda_{23}m_2 + \lambda_{12}m_1 \\ 0 = -\lambda_{34}m_3 - \lambda_{35}m_3 - \lambda_{36}m_3 + \lambda_{23}m_2 \\ 0 = -\lambda_{41}m_4 + \lambda_{34}m_3 \\ 0 = -\lambda_{51}m_5 + \lambda_{35}m_3 \\ 0 = -\lambda_{61}m_6 + \lambda_{36}m_3 \end{cases},$$

звідки, з урахуванням нормувальної умови:

$$\begin{cases} 0 = -0.08m_1 + m_4 + 0.5m_5 + 4m_6 \\ 0 = -2m_2 + 0.08m_1 \\ 0 = -4m_3 + 2m_2 \\ 0 = -m_4 + 2m_3 \\ 0 = -0.5m_5 + m_3 \\ 0 = -4m_6 + m_3 \\ m_1 + m_2 + m_3 + m_4 + m_5 + m_6 = 100 \end{cases}.$$

У результаті розв'язання даної системи рівнянь будуть отримані такі значення:

$$m_1 = 87.34, m_2 = 3.49, m_3 = 1.75, m_4 = 3.49, m_5 = 3.49, m_6 = 0.44.$$

Далі визначаються дисперсії чисельностей m_j:

$$D_j = m_j\left(1 - \frac{m_j}{\sum_{i=1}^{N} m_i}\right).$$

Для даного прикладу:

$$D_1 = m_1\left(1 - \frac{m_1}{N}\right) = 87.34\left(1 - \frac{87.34}{100}\right) \approx 11.057,$$

$$D_2 = D_4 = D_5 \approx 3.368, \qquad D_3 \approx 1.719, \qquad D_6 \approx 0.438,$$

і середні квадратичні відхилення:

$$\sigma_1 = \sqrt{D_1} \approx 3.3252, \qquad \sigma_2 = \sigma_3 = \sigma_5 = 1.8352,$$

$$\sigma_3 = 1.3111, \qquad \sigma_6 = 0.6618.$$

По середнім квадратичним відхиленням знаходиться діапазон можливих значень чисельності для кожного стану s_j як $(m_j \pm 3 \cdot \sigma_j)$.

Слід зауважити, що знаючи математичні сподівання та середні квадратичні відхилення чисельностей станів, є можливість оцінювати також і

ймовірності різних станів системи загалом, тобто, наприклад, ймовірність того, що чисельність якогось стану буде в певних заданих межах.

Так, якщо число елементів N велике, то чисельність стану s_k можна наближено вважати розподіленою за нормальним законом. Звідси виходить, що ймовірність того, що випадкова величина x_k (чисельність стану s_k) у межах від α до β буде виражатися формулою:

$$P\{x_k \in (\alpha, \beta)\} = \Phi_0\left(\frac{\beta - m_k}{\sigma_k}\right) - \Phi_0\left(\frac{\alpha - m_k}{\sigma_k}\right),$$

де m_k і σ_k – математичне сподівання і середнє квадратичне відхилення чисельності стану s_k, $\Phi_0(x)$ – функція Лапласа.

Контрольні запитання

1. Що таке марковська модель?
2. Які типи марковських моделей існують?
3. Що таке дискретний марковський процес?
4. Шо таке ланцюг Маркова?
5. Як визначаються ймовірності переходів у дискретному марковському процесі?
6. Що таке матриця ймовірностей переходів?
7. Які властивості має матриця ймовірностей переходів?
8. Що таке поглинаючий стан?
9. Опишіть задачу «розорення азартного гравця».
10. Як обчислити ймовірність переходу за k кроків?
11. Що таке схема «загибелі і розмноження»?
12. Яка роль інтенсивностей λ («розмноження») та μ («загибелі»)?
13. Чим схема «загибелі і розмноження» відрізняється від загального марковського ланцюга?
14. Що таке система масового обслуговування (СМО)?
15. Які основні елементи включає будь-яка СМО?
16. Що таке черга в СМО?

17. Що таке канал обслуговування?

18. Як інтерпретується коефіцієнт завантаження $\rho = \lambda/\mu$?

19. Чим відрізняється СМО з відмовами від СМО з очікуванням?

20. Що таке СМО з необмеженою чергою? Наведіть приклади.

21. Що таке СМО з обмеженою чергою? Наведіть приклади.

22. Які показники ефективності СМО розглядаються найчастіше?

23. Як обчислити ймовірність простоювання каналу?

24. Що таке багатоканальна СМО? Наведіть приклад одно- та багатоканальної СМО.

25. Які характеристики визначають пропускну здатність СМО?

26. Що таке метод динаміки середніх?

27. Яка основна ідея методу динаміки середніх?

28. Чим відрізняється метод динаміки середніх від точних методів?

29. Яке значення має математичне сподівання в методі динаміки середніх?

30. Як будується система рівнянь у методі динаміки середніх?

31. Як метод динаміки середніх пов’язаний з марковськими процесами?

32. Які існують переваги методу динаміки середніх порівняно з імітаційним моделюванням?

33. Наведіть приклади практичних задач, що вирішуються методом динаміки середніх.

3. СТАТИСТИЧНЕ МОДЕЛЮВАННЯ

3.1. Сутність імітаційного моделювання

Імітаційне моделювання – процес конструювання моделі реальної системи та постановки експериментів на цій моделі з метою або зрозуміти поведінку системи, або оцінити (у рамках обмежень, що накладаються деяким критерієм чи сукупністю критеріїв) різні стратегії, що забезпечують функціонування цієї системи [4]. Імітаційне моделювання – це хід обчислення характеристик динамічного процесу шляхом відтворення його перебігу на комп'ютері на основі математичної моделі. Такий підхід застосовується тоді, коли отримати потрібні результати іншими методами складно або неможливо. Саме це комп'ютерне відтворення процесу за допомогою моделі називають імітаційним експериментом.

Широке впровадження імітаційного моделювання стало можливим лише на певному етапі розвитку інформаційних технологій – засобів збору, передавання, обробки та зберігання інформації. До таких засобів належать не лише комп'ютери, але й засоби обчислювальної математики, багаторівневі програмні інструменти, системи керування базами даних, інформаційні банки та інші цифрові технології.

Імітаційні моделі належать до класу моделей, що представляють собою систему співвідношень між характеристиками певного процесу. Ці характеристики умовно поділяються на внутрішні (фазові змінні) та зовнішні (параметри).

Внутрішні характеристики – це ті, значення яких визначаються в результаті імітаційного моделювання. Зовнішні характеристики – це такі, що суттєво впливають на внутрішні, проте зворотного впливу з боку внутрішніх змінних на них не передбачається.

Особливістю імітаційних моделей є те, що внутрішні змінні розглядаються як функції часу, а модельні співвідношення повинні

однозначно описувати їхній розвиток, якщо задані всі необхідні зовнішні параметри. Для здійснення такого моделювання обов'язковою є наявність початкових значень внутрішніх змінних, тобто значень, які відповідають певному стартовому моменту часу. Це базове припущення є основою для прогнозування динаміки процесу.

Модель, яка здатна здійснювати прогноз внутрішніх характеристик, повинна бути замкнутою. Це означає, що її математичні співвідношення мають забезпечувати можливість обчислення внутрішніх змінних на основі заданих зовнішніх параметрів. Визначення зовнішніх характеристик такої моделі здійснюється в процесі ідентифікації або калібрування.

Імітаційне моделювання вирізняється тим, що передбачає не просто побудову математичної моделі, але й організацію обчислювального експерименту засобами обчислювальної техніки та аналіз отриманих результатів. Воно не виключає використання аналітичних моделей, особливо у випадках, коли інформація про початкові умови чи вхідні впливи є неповною. Навпаки – імітація доповнює аналітичні підходи, дозволяючи розширити можливості їх застосування до складніших умов дослідження.

При цьому важливу роль відіграє саме технологічний аспект моделювання – тобто не лише побудова моделей, але й організація їх реалізації, обчислень, зміни параметрів, обробки результатів. У цьому сенсі технологія імітаційного моделювання – це спосіб здобуття нових знань: значень внутрішніх характеристик системи, що розраховуються на основі вже наявних зовнішніх параметрів.

Частина практично значущих задач – зокрема й оптимізаційних – або не піддається вирішенню за допомогою класичних моделей, або рішення, отримані таким шляхом, виявляються недостатньо ефективними. Основні причини цього – обмежена здатність моделей деталізувати складні системи та відсутність ефективних аналітичних методів для їх обчислення.

Типові моделі зазвичай мають високий рівень абстракції, що підвищує їхню універсальність, але знижує практичну точність. Зі зростанням ролі

автоматизованих систем управління та розвитком обчислювальної техніки зростає складність завдань, які потребують вирішення. Сучасна практика висуває нові вимоги – моделі мають враховувати невизначеність, динаміку, взаємозалежність подій та рішень.

У таких умовах створення адекватних моделей і пошук ефективних методів їх розв’язання стають критично важливими – особливо для задач, де рішення впливають на майбутні події і мають системний ефект. Як правило, такі практичні завдання мають велику розмірність, мають велику кількість внутрішніх взаємозв’язків, тому їх не вдається звести до відомих моделей типу математичного програмування або застосувати для їх вирішення інші традиційні методи математичного моделювання.

Експериментування з реальними об’єктами має такі особливості:

- експеримент може порушити звичний порядок функціонування системи;
- важко забезпечити ідентичні умови під час кожного повторення експерименту або підтримувати їх стабільними протягом тривалого часу;
- для досягнення статистично значущих результатів може знадобитися надто багато часу й ресурсів;
- в експериментах із реальними об’єктами іноді неможливо перевірити всі можливі варіанти або сценарії через обмеження ресурсів чи ризики;
- якщо до системи залучено людей, результат може бути спотворений ефектом Хоторна – коли учасники змінюють свою поведінку через усвідомлення, що за ними спостерігають.

Імітаційне моделювання доцільне у випадках:

- якщо не існує чітко сформульованої математичної моделі або відсутні аналітичні методи її розв’язання;

- якщо математичні методи існують, але вони надто складні та трудомісткі, тоді імітація є ефективнішою альтернативою;
- якщо реалізація аналітичного розв'язання ускладнена через нестачу фахівців, а залучення зовнішніх експертів є дорожчим, ніж розробка імітаційної моделі;
- якщо крім числових результатів важливо також візуально спостерігати за розвитком процесу в часі;
- якщо моделювання в реальних умовах є складним або небезпечним (наприклад, аналіз поведінки космічних апаратів у міжпланетних польотах);
- якщо дослідження стосується довготривалих процесів, і виникає потреба змінювати масштаб часу – уповільнювати або прискорювати події. Імітація дозволяє повністю контролювати часову шкалу.

На відміну від аналітичної моделі імітаційна модель при ускладненні модельованого об'єкта зростає не з прискоренням, а з певного рівня – незначно, як наведено на рис. 3.1.

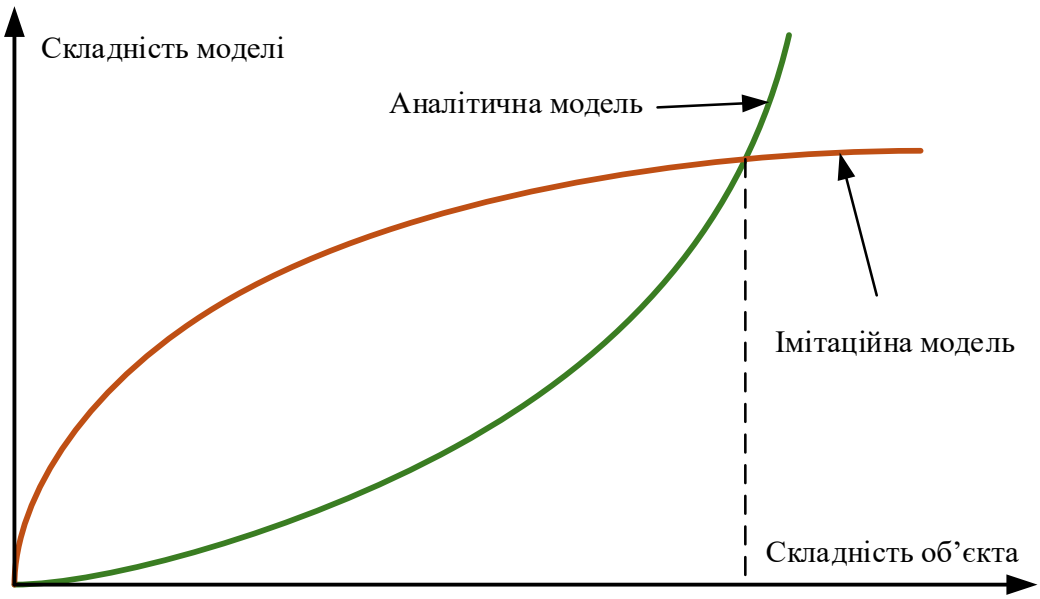


Рисунок 3.1 – Зростання складності моделей

Для вирішення завдань моделювання розробляється, а останніми роками отримав особливо широкий розвиток, метод імітаційного моделювання засобами обчислювальної техніки.

За одним із визначень, імітація – це чисельний метод проведення комп'ютерних експериментів із математичними моделями, що описують поведінку складних систем протягом тривалих періодів часу. Математичні моделі, які використовуються в імітації, можуть суттєво відрізнятися від традиційних.

Під час імітаційного моделювання структура досліджуваної системи відображається в моделі максимально точно, а самі процеси її функціонування відтворюються у формі обчислювального експерименту, тобто імітуються в рамках побудованої моделі. Тому процес створення імітаційної моделі полягає у всебічному описі як структури системи, так і характеру (механізму) її функціонування. Опис імітаційної моделі має дві основні складові:

- статичний опис системи – це опис її внутрішньої структури, який ґрунтується на проведенні структурного аналізу об'єкта моделювання. У цьому описі визначаються основні елементи системи, їхні властивості, зв'язки між ними, типи взаємодій, функції кожного елемента, а також обмеження, які накладаються на систему в цілому;
- динамічний опис системи – це опис процесів, які відбуваються в моделі в часі, тобто опис механізмів динамічної взаємодії елементів системи. Такий опис передбачає побудову функціональної моделі, яка імітує поведінку реальної системи в часі, з урахуванням змін її стану, впливу зовнішніх факторів, обробки подій, що виникають, і реакції системи на них.

3.2. Загальна характеристика методу імітаційного моделювання

У найзагальнішому вигляді структуру моделі можна подати математично у вигляді:

$$E = f(x_i, y_j)$$

де E – результат функціонування системи; x_i – змінні та параметри, на які можна впливати (керовані); y_j – змінні та параметри, на які не можна впливати (некеровані); f – функціональна залежність між x_i та y_j, яка визначає значення E.

Таке спрощення є корисним тим, що воно дозволяє чітко показати, як залежить функціонування системи як від керованих, так і від некерованих змінних.

Кожна модель, як правило, являє собою певну комбінацію таких складових:

- компоненти,
- змінні,
- параметри,
- функціональні залежності,
- обмеження,
- цільові функції.

Компонентами математичної моделі системи називають складові частини, які при відповідному об’єднанні утворюють систему. Іноді під компонентами маються на увазі елементи системи або її підсистеми. Система визначається як група або сукупність об’єктів, об’єднаних певною формою регулярної взаємодії чи взаємозалежності з метою виконання заданої функції. Компоненти – це об’єкти, що утворюють досліджувану систему.

Параметри – це величини, які оператор моделі може обирати довільно, на відміну від змінних, значення яких визначаються видом заданої функції. Можна сказати, що після встановлення параметри є фіксованими (незмінними) величинами.

У моделі системи розрізняють два типи змінних: екзогенні та ендогенні.

- екзогенні змінні (або вхідні) – породжуються поза системою або є результатом впливу зовнішніх чинників;
- ендогенні змінні – виникають у межах системи або через її внутрішні чинники. Їх також називають змінними стану (якщо вони описують поточний стан системи) або вихідними змінними (якщо відображають результат роботи системи).

У статистиці екзогенні змінні часто називають незалежними, а ендогенні – залежними.

Функціональні залежності описують поведінку змінних і параметрів у межах одного компонента або визначають зв’язки між компонентами системи. Ці залежності можуть бути:

- детермінованими – коли результат однозначно визначається заданими вхідними даними (тотожності, рівняння, правила);
- стохастичними – коли вихід залежить від випадкових факторів навіть за фіксованих вхідних значень.

Обидва типи зазвичай задаються у вигляді математичних рівнянь, які встановлюють зв’язок між ендогенними (вихідними) та екзогенними (вхідними) змінними. Такі рівняння зазвичай формулюються як гіпотези або отримуються на основі статистичного чи математичного аналізу.

Обмеження – це умови, які встановлюють допустимі межі для змінних або ресурсів (час, енергія, запаси тощо). Вони можуть бути:

- штучними – заданими розробником моделі;
- природними – зумовленими властивостями самої системи.

Цільова функція (або функція критерію) точно відображає мету або задачу, які ставляться перед системою, і містить правила оцінки ефективності їх досягнення. Цільова функція має бути чітким і недвозначним математичним представленням задач, які повинні узгоджуватись із

рішеннями, що приймаються. Правильне визначення цільової функції надзвичайно важливе, оскільки:

- воно впливає на структуру моделі та її використання;
- некоректно сформульована функція може призвести до помилкових висновків.

У загальному випадку цільова функція є невід'ємною частиною моделі, а весь процес роботи з моделлю спрямований на оптимізацію або досягнення заданого критерію.

Якісна модель повинна відповідати таким критеріям:

- бути простою і зрозумілою для користувача;
- мати чітко визначену мету;
- бути надійною (не давати абсурдних результатів);
- бути зручною в експлуатації;
- забезпечувати вирішення основних задач;
- бути адаптивною (легко оновлюваною або змінюваною);
- мати властивість розширюваності: тобто розвиватись від простої до більш складної моделі в процесі взаємодії з користувачем.

З погляду програмної реалізації, сутність методу імітаційного моделювання полягає в тому, що елементам системи відповідають певні програмні компоненти, а їхні стани відображаються змінними. Функціонування моделі забезпечується моделюючим алгоритмом, який реалізує поведінку окремих елементів системи, їх взаємодію та обмін інформацією. Окремо описується алгоритм модифікації змінних, що характеризують стан системи в певний момент часу. Часова динаміка системи моделюється через механізм просування модельного часу, що дозволяє відтворити послідовність подій і змін станів у часовому масштабі.

Метод імітаційного моделювання дозволяє описувати та відтворювати взаємодію між елементами системи. Для побудови імітаційної моделі потрібно:

- представити реальну систему або процес як сукупність взаємодіючих елементів;
- алгоритмічно описати функціонування кожного елемента;
- змоделювати взаємодію між елементами, а також їхній зв’язок із зовнішнім середовищем.

Ключовим етапом є визначення станів системи через набір змінних, значення яких відображають конкретний стан. Зміна комбінацій цих змінних імітує перехід системи між станами. Таким чином, імітаційне моделювання – це опис динамічної поведінки системи як послідовного переходу з одного стану в інший згідно з операційними правилами. Перехід між станами може бути як неперервним, так і дискретним у часі.

Імітаційне моделювання – це спосіб відображення динаміки змін стану системи в часі. Щоб змоделювати такі зміни, керуюча програма має містити механізм управління модельним часом. Для реалізації паралельності подій, характерної для реальних систем, у моделі вводиться глобальна змінна, яку називають модельним (або системним) часом. Вона слугує для синхронізації всіх подій, що відбуваються в системі. Існують такі основні способи просування модельного часу:

- покроковий підхід – модельний час змінюється на фіксовані інтервали;
- подійний підхід – час змінюється на величину до моменту настання наступної події, тобто крок часу є змінним.

Покрокове просування модельного часу здійснюється з мінімально можливою константною довжиною кроку. Такий підхід доцільно застосовувати в таких випадках:

- коли динаміка моделі описується інтегро-диференціальними рівняннями, що розв’язуються чисельними методами. У цьому випадку модель є дискретним наближенням до безперервного реального процесу;

- якщо події в системі розподілені рівномірно, і можливо підібрати оптимальне значення константного кроку часу;
- у ситуаціях, коли поява подій є складнопередбачуваною, або коли події виникають масово, у вигляді груп.

Недоліком покрокового методу є низька ефективність використання обчислювальних ресурсів, оскільки модельний час змінюється навіть у ті моменти, коли система залишається незмінною.

Подійний метод просування часу є більш ефективним у випадках, коли події розподілені нерівномірно, а між ними існують значні часові інтервали. У цьому підході модельний час змінюється лише в моменти зміни стану системи, тобто тільки при виникненні подій. Крок часу при цьому може бути максимально можливим – він дорівнює часовому інтервалу до настання наступної події. Такий підхід дозволяє значно прискорити хід симуляції у випадках рідкісного виникнення подій. На практиці подійний метод є найбільш поширеним. Він забезпечує всевдопаралельне моделювання, тобто перетворення паралельних процесів у послідовні для зручності реалізації.

Виходячи зі способу просування модельного часу, розрізняють три типи імітаційних моделей:

- безперервні моделі – зміна стану системи відбувається як безперервна функція часу, що зазвичай описується системами диференціальних рівнянь. Просування часу реалізується через чисельні методи;
- дискретні моделі – зміни в системі відбуваються лише у визначені моменти часу, відповідні подіям. Динаміка зводиться до переходів від однієї події до іншої;
- безперервно-дискретні моделі – поєднують механізми обох типів, ураховуючи як безперервні зміни, так і дискретні події.

Логіко-математичні моделі, що використовуються в імітаційному моделюванні складних систем, можуть мати як алгоритмічну, так і

неалгоритмічну природу. Для того щоб така модель могла бути реалізована комп’ютерно, створюється моделюючий алгоритм, який описує структуру системи та логіку взаємодії її елементів. Програмна реалізація цього алгоритму утворює імітаційну модель, що створюється за допомогою спеціальних засобів автоматизації моделювання.

Агентне моделювання – відносно новий напрямок в імітаційному моделюванні, який використовується для дослідження децентралізованих систем, динаміка функціонування яких визначається результатом індивідуальної активності членів групи. Агент – якась сутність, що володіє активністю, автономною поведінкою, може приймати рішення відповідно з деяким набором правил, взаємодіяти з оточенням, а також самостійно змінюватися. Використовується для дослідження децентралізованих систем, динаміка функціонування яких визначається результатом індивідуальної активності членів групи.

Системна динаміка – парадигма моделювання, де для досліджуваної системи будуються графічні діаграми причинних зв’язків і глобальних впливів одних параметрів на інші в часі, а потім створена на основі цих діаграм модель імітується на комп’ютері. За допомогою системної динаміки будують моделі бізнес-процесів, розвитку міста, моделі виробництва, динаміки популяції, екології та розвитку епідемії тощо.

Виходячи з того, що імітаційне моделювання має застосовуватись для дослідження реальних систем, можна виділити такі основні етапи цього процесу:

1. Визначення системи – встановлення меж досліджуваної системи, обмежень її функціонування та критеріїв ефективності, за якими буде оцінюватись її робота.
2. Формалізація моделі – абстрагування від реального об’єкта до логічної схеми, що описує систему у вигляді моделі.

3. Підготовка даних – збір та опрацювання інформації, необхідної для побудови моделі, її представлення в придатному для подальшого використання форматі.
4. Трансляція моделі – опис моделі мовою, що відповідає можливостям обраної обчислювальної платформи.
5. Оцінювання адекватності – перевірка відповідності моделі реальній системі та підвищення рівня довіри до висновків, зроблених на основі результатів моделювання.
6. Стратегічне планування – розробка загального плану імітаційного експерименту, спрямованого на отримання необхідної інформації. При стратегічному плануванні ставиться завдання побудови оптимального плану експерименту для досягнення мети поставленої перед моделюванням (оптимізація структури алгоритмів і параметрів системи).
7. Тактичне планування – деталізація способу проведення кожної серії експериментів відповідно до загального плану. Тактичне планування має частинні цілі оптимальної реалізації кожного конкретного експерименту з безлічі необхідних заданих при стратегічному плануванні.
8. Проведення експерименту – безпосереднє здійснення імітації з метою отримання потрібних даних, у тому числі аналіз чутливості моделі до зміни параметрів.
9. Інтерпретація результатів – формування висновків і узагальнень на основі даних, отриманих у процесі моделювання.
10. Реалізація – практичне використання моделі або результатів моделювання для прийняття рішень чи вдосконалення системи.
11. Документування – фіксація перебігу реалізації проєкту, результатів експериментів, а також опис процесу створення, тестування та застосування моделі.

Загальна схема проведення імітаційного моделювання наведена на рис. 3.2.

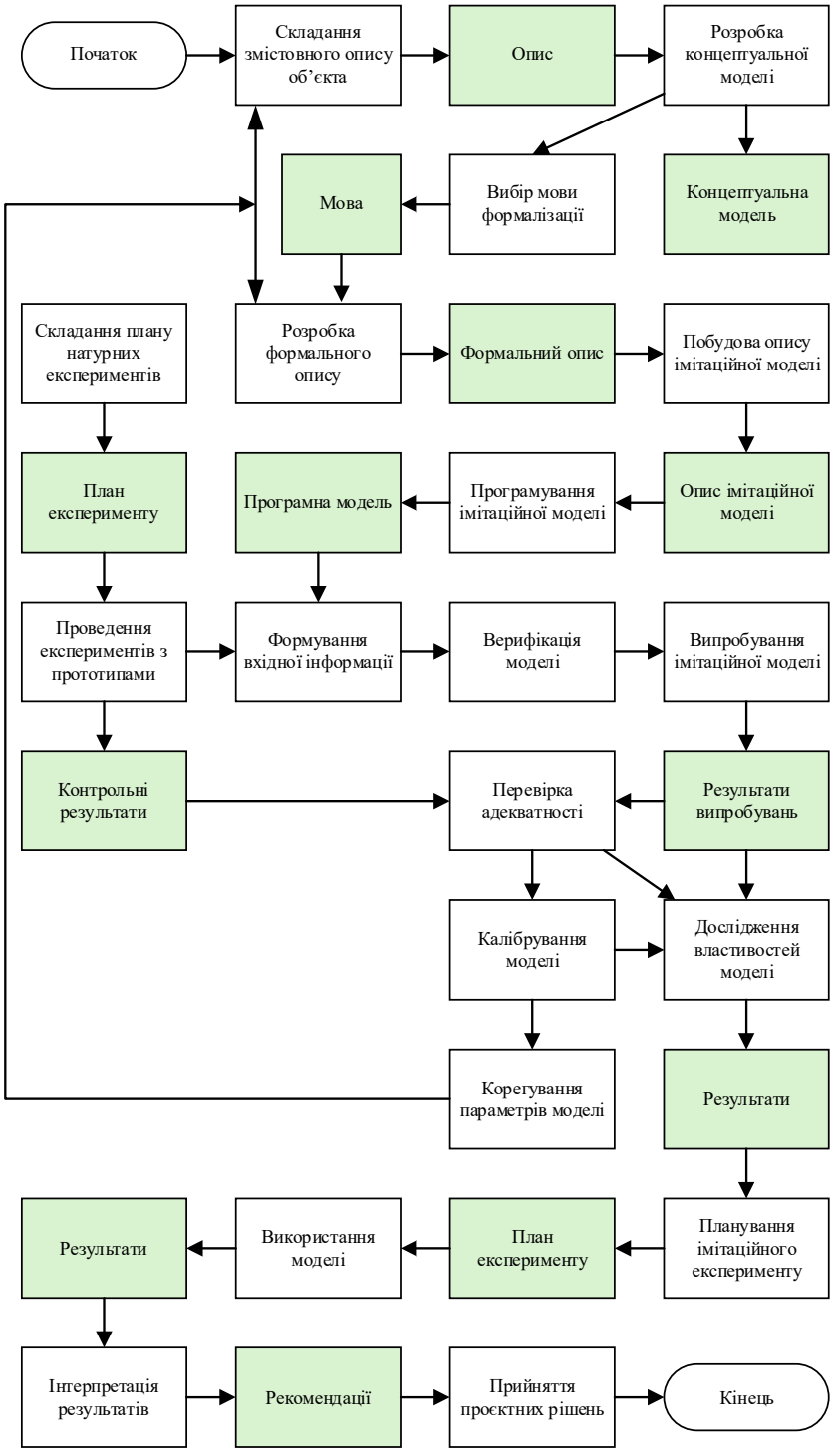


Рисунок 3.2 – Загальна схема імітаційного моделювання

Під час проведення імітаційного моделювання найчастіше допускають такі три основні класи помилок:

1. Помилки формалізації – виникають, коли модель недостатньо точно або детально описує реальну систему; це може призвести до втрати суттєвих характеристик або взаємозв’язків.
2. Помилки розв’язання – пов’язані з використанням некоректного або надмірно спрощеного методу побудови чи реалізації моделі; внаслідок цього модель може неадекватно відтворювати динаміку системи.
3. Помилки задання параметрів – виникають, коли в модель вводять невірні або неточні значення параметрів системи, що зумовлює викривлення результатів моделювання.

Основні види перевірок, які необхідно виконувати в процесі розробки імітаційної моделі:

- перевірка моделей елементів – оцінка правильності опису окремих компонентів системи в моделі (чи правильно вони реалізовані згідно з реальними функціями);
- перевірка моделей зовнішніх впливів – перевіряється, наскільки достовірно змодельовано вплив зовнішніх чинників на систему;
- перевірка концептуальної моделі – аналіз логіки побудови моделі: чи охоплює вона всі необхідні елементи, зв’язки та поведінку;
- перевірка формалізованої та математичної моделі – контроль коректності математичних співвідношень, рівнянь і формальних описів процесів;
- перевірка способів вимірювання та обчислення вихідних характеристик – оцінка правильності обчислень та метрик, які використовуються для аналізу результатів моделювання;
- перевірка програмної моделі – тестування реалізації моделі у вигляді програмного коду: перевірка на помилки, баги, ефективність і точність обчислень.

Якщо за результатами перевірки виділяється неприпустиме розходження моделі та системи, виникає необхідність коригування або змін. Вирізняють такі типи змін:

- глобальні – виникають у випадку методичних помилок у концептуальній або математичній моделі, потребують повторення моделювання з самого початку;
- локальні – пов'язані з уточненням деяких параметрів та алгоритмів, потребують заміну компонентів моделі більш точними;
- параметричні (калібрувальні) – зміни деяких спеціальних характеристик.

3.3. Класифікація випадкових процесів

За допомогою випадкових процесів можна описати велику кількість різноманітних систем. Випадковий процес, який відбувається в системі, полягає в тому, що з плином часу система у випадковий спосіб змінює свій стан, який може бути охарактеризованим за допомогою певних числових змінних. Поняття випадкового процесу є узагальненням поняття випадкової величини.

Випадковий процес – це сімейство випадкових величин $X(t)$, які залежать від деякого параметра t, що належить множині значень параметра T. У більшості практичних застосувань параметр t інтерпретується як час.

Якщо система S у момент часу t описується однією скалярною випадковою величиною X, то розглядається скалярний випадковий процес $X(t)$. Якщо стан системи S у момент часу t описується кількома випадковими величинами $X_1, X_2, \dots, X_k$, то розглядається векторний випадковий процес.

Перетин випадкового процесу в момент часу t_0 – це звичайна випадкова величина $X(t_0)$, яка утворюється в результаті фіксації значення часу t_0 у випадковому процесі.

Аналогічно до того, як випадкова величина може бути подана як функція елементарної події (що виникає внаслідок стохастичного експерименту), випадковий процес $X(t)$ можна розглядати як функцію двох аргументів – часу t та елементарної події ω:

$$X(t) = \phi(t, \omega); t \in T;\ \omega \in \Omega; X(t) \in \mathrm{X},$$

де ω – елементарна подія; Ω – простір елементарних подій; X – множина можливих значень випадкового процесу (фазовий простір).

Припустимо, що стохастичний експеримент проведено, і відбулася певна подія $\omega_0 \in \Omega$, тобто другий аргумент ω у функції $\phi(t, \omega)$ зафіксовано, тоді:

$$x(t) = \phi(t, \omega_0); t \in T.$$

Цю функцію називають реалізацією або траєкторією випадкового процесу. Таким чином, реалізація випадкового процесу є невипадковою функцією, яка утворюється в результаті конкретного проведення експерименту.

Випадкові процеси прийнято класифікувати залежно від структури множини станів системи та структури множини значень аргументу t.

Випадковий процес $X(t)$ називається процесом з дискретним часом, якщо система, в якій він відбувається, може змінювати свої стани лише в моменти $t_1, t_2, \dots, t_n, \dots$, число яких є скінченним або зліченим (дискретна множина моментів часу $T = \{t_1, t_2, \dots, t_n, \dots\}$).

Перетини випадкового процесу з дискретним часом утворюють послідовність випадкових величин, тому випадкові процеси з дискретним часом називають також випадковими послідовностями або часовими рядами.

Випадковий процес $X(t)$ називається процесом з неперервним часом, якщо система, в якій він відбувається, може переходити зі стану в стан у будь-який момент часу t (при будь-якому значенні аргументу). Для процесу з неперервним часом множина моментів часу T є незліченою, тобто T – деякий проміжок дійсної осі часу t.

Випадковий процес $X(t)$ називається процесом з дискретними станами, якщо кожен його перетин у будь-який момент часу t є дискретною випадковою величиною. Множина станів системи скінченна або зчисленна. Перетини можуть бути одновимірними або багатовимірними.

Випадковий процес $X(t)$ називається процесом з неперервними станами, якщо всі його перетини при будь-якому значенні аргументу t є неперервними випадковими величинами.

Таким чином, випадкові процеси можна поділити на такі основні класи:

- процеси з дискретними станами і дискретним часом;
- процеси з дискретними станами і неперервним часом;
- процеси з неперервними станами і дискретним часом;
- процеси з неперервними станами і неперервним часом.

Більш інформативною та практично значущою є класифікація випадкових процесів за типом залежності між значеннями $X(t)$ у різні моменти часу t. Такий підхід дозволяє глибше аналізувати динаміку процесів, враховуючи характер кореляцій між їхніми станами на різних часових інтервалах:

- випадкові процеси з незалежними значеннями (значення процесу в різні моменти часу не залежать одне від одного);
- випадкові процеси з незалежними приростами (прирости процесу на неперетинних інтервалах часу є незалежними випадковими величинами);
- марковські випадкові процеси (майбутній стан процесу залежить тільки від його поточного стану, але не від попередньої історії);
- стаціонарні випадкові процеси (статистичні характеристики, такі як математичне сподівання, дисперсія, кореляція тощо, не змінюються з часом);
- мартингали (процеси, які в майбутньому залишаються в середньому постійними).

Випадковий процес $X(t)$ називається процесом **із незалежними значеннями**, якщо для будь-яких двох моментів часу перетини (перерізи) процесу в ці моменти – незалежні. Якщо час дискретний, то процес із незалежними значеннями – це те саме, що й послідовність незалежних випадкових величин.

Випадковий процес з незалежними приростами – це такий процес $X(t)$, якщо для довільних моментів часу $t_1, t_2, \dots, t_n \in T$, таких що задовольняють умові $t_1 \le t_2 \le \cdots \le t_n$, випадкові величини (прирости процесу $X(t)$, що не пересікаються) $X(t_n) - X(t_{n-1}), \dots, X(t_3) - X(t_2)$, $X(t_2) - X(t_1)$, $X(t_1)$ є незалежними.

Прикладом такого процесу може бути так званий пуассонівській процес, який використовується для опису реальних процесів, що змінюються стрибкоподібно (імпульсні процеси).

Також випадкові процеси з незалежними приростами використовують у фізиці, зокрема так званий вінеровський процес – процес, незалежні прирости якого мають нормальний розподіл із нульовим математичним сподіванням та дисперсією, що дорівнює величині прирощення інтервалу часу. Вінеровський процес є математичною моделлю броунівського руху.

Випадковий процес $X(t)$ називається **марковським**, якщо для будь-яких двох моментів часу t_1 та t_2 (причому $t_1 < t_2$) умовний розподіл ймовірностей перерізу $X(t_2)$ за умови, що задані всі значення $X(t)$ при $t < t_1$, залежить тільки від $X(t_1)$. Тобто, за відомого поточного стану процесу, його майбутнє не залежить від минулого. Через таке визначення марковські процеси називають процесами без післядії. Марковські процеси є природним узагальненням детермінованих процесів. У детермінованих процесах стан системи в час t_1 однозначно визначає хід процесу в майбутньому. У марковських процесах стан системи на момент t_1 однозначно визначає розподіл ймовірностей ходу процесу при $t < t_1$, причому жодні відомості про перебіг процесу до моменту t_1 не змінюють цей розподіл, тобто майбутня

поведінка системи, що описується марковським випадковим процесом, цілком визначається її станом в останній з відомих моментів часу.

Марковський процес є стохастичною математичною моделлю багатьох процесів у біології (розподіл епідемій, ріст популяцій), у фізиці (радіоактивний розпад), у теорії масового обслуговування тощо.

Стаціонарні випадкові процеси. Важливою особливістю випадкових функцій є залежність властивостей таких функцій від початку відліку часу. Відповідно до цього розглядають стаціонарні та нестаціонарні випадкові процеси.

Розрізняють два види стаціонарності: у вузькому та широкому значенні.

Випадковий процес $X(t)$ називається стаціонарним у вузькому значенні, якщо всі n-вимірні функції розподілу ймовірностей однакові при будь-якому t_0 і всіх $t_1, t_2, \dots, t_n$ з області його визначення, тобто справедлива рівність:

$$F_n(x_1, x_2, \dots, x_n;\ t_1, t_2, \dots, t_n) = F_n(x_1, x_2, \dots, x_n;\ t_1 + t_0, t_2 + t_0, \dots, t_n + t_0),$$

де $n = 1,2, \dots$.

Аналогічна рівність виконується і для функції щільності ймовірностей:

$$f(x_1, x_2, \dots, x_n;\ t_1, t_2, \dots, t_n) = f(x_1, x_2, \dots, x_n;\ t_1 + t_0, t_2 + t_0, \dots, t_n + t_0),$$

де $n = 1,2, \dots$, $-\infty < t_0 < +\infty$, а також для моментних і кореляційних функцій.

Стаціонарний випадковий процес можна аналізувати, починаючи з будь-якого моменту часу, оскільки він знаходиться в стані ймовірної рівноваги.

Випадкові процеси, що стаціонарні у вузькому значенні, завжди стаціонарні в широкому значенні. Зворотне твердження в загальному випадку невірне.

Стаціонарні процеси є частковим випадком більш широкого класу – нестаціонарних процесів. Якщо будь-яка зі щільностей розподілу змінюється при зміні початку відліку часу, то випадковий процес називається

нестаціонарним. У цьому випадку хоча б одна з моментних характеристик залежить від часу.

Стаціонарний випадковий процес називається ергодичним, якщо його характеристики, знайдені усередненням множини реалізацій, співпадають з відповідними характеристиками, знайденими усередненням за часом однієї реалізації $x(t)$, яка досліджувалася на інтервалі $(0;\ T)$, де $T \to \infty$.

Умова ергодичності виконується лише для стаціонарного процесу, тому ергодичні процеси є також стаціонарними. Усі нестаціонарні процеси є неергодичними, але неергодичними можуть також бути і стаціонарні випадкові процеси. Ергодичні випадкові процеси є стаціонарними у вузькому значенні, зворотне твердження взагалі невірне.

Схема стаціонарних випадкових процесів із хорошим наближенням визначає багато реальних явищ, що супроводжуються невпорядкованими флуктуаціями. Так, наприклад, пульсація швидкості в точці турбулентної течії являє собою стаціонарний випадковий процес, якщо течія постійна.

В економіці значну роль грає вивчення економічних часових рядів. У загальному випадку випадкова складова економічного часового ряду є стаціонарною випадковою послідовністю.

Мартингал – це стохастичний процес $X(t)$, для якого в кожний момент часу t математичне сподівання майбутнього значення, за умови всього попереднього розвитку процесу до моменту t, дорівнює його поточному значенню.

Мартингали знайшли своє застосування при моделюванні стохастичних процесів, пов’язаних з фінансами, азартними іграми та іншими різними галузями аналізу даних.

3.4. Способи просування модельного часу

Для опису динаміки модельованих процесів в імітаційному моделюванні реалізується механізм задання модельного часу. Такий механізм вбудовується в керуючі програми будь-якої системи моделювання.

Якщо б імітувалась поведінка однієї компоненти системи, то виконання дій в імітаційній моделі можна було б робити послідовно, за допомогою перерахунку часової координати. А для забезпечення імітації паралельних подій реальної системи вводиться спеціальний механізм, що забезпечує синхронізацію всіх подій у системі, який має назву – модельний (або системний) час t_m.

Механізм модельного часу виконує низку важливих функцій у процесі імітаційного моделювання, а саме:

- відображення переходу модельованої системи з одного стану в інший – забезпечує фіксацію змін у системі, що відбуваються в певні моменти часу;
- синхронізація компонент моделі – координує взаємодію між частинами системи, щоб події та процеси відбувалися узгоджено;
- зміна масштабу часу «життя» досліджуваної системи – дозволяє пришвидшити або сповільнити хід часу, що дає змогу спостерігати за поведінкою системи в зручному темпі;
- управління ходом модельного експерименту – визначає послідовність, умови зупинки, повтори та інші аспекти моделювання;
- моделювання квазіпаралельної реалізації подій – забезпечує обробку подій, що в реальності можуть відбуватись паралельно, у вигляді послідовного, але узгодженого виконання.

Залежно від реалізації, механізм просування модельного часу може бути заснований на:

- реальному часі системи, функціонування якої імітується;
- модельному часі, за яким організовується синхронізація подій у моделі;
- машинному часі імітації, що відображає витрати ресурсу обчислювальної системи.

Час у комп'ютерній моделі принципово не може протікати безперервно: кожен момент часу виконується одна команда. Але окремі події реального процесу, що протікають одночасно і миттєво, в імітаційній моделі реалізуються ланцюжком команд, для виконання яких витрачається машинний час. Отже, модельний час просувається дискретно, стрибками.

Просування часу в моделі може бути організовано двома способами:

- просування модельного часу до чергової події (за принципом «від події до події» Δx);
- просування модельного часу з фіксованим змінним кроком Δt.

Підхід, що використовує **просування часу в моделі від події до події**, застосовується всіма основними комп'ютерними програмами та більшістю розробників, які створюють свої моделі універсальними мовами програмування.

Під час аналізу процесів функціонування деяких систем можна виявити, що для них характерні два типи станів:

- особливі стани, які виникають у конкретні моменти часу;
- неособливі стани, у яких система перебуває весь інший час.

Тип події пов'язаний з видом елемента системи, що змінює свій стан. Зміна стану всієї системи розглядається як сукупність змін станів її окремих елементів.

При цьому приймається припущення, що подія відбувається миттєво – у конкретний момент часу. Насправді ж, зміна стану (тобто подія) триває певний проміжок часу. Проте в процесах, які моделюються, ця тривалість набагато менша, ніж час перебування системи в якомусь фіксованому стані. Тому цією тривалістю події можна знехтувати.

Особливими називають такі стани, в які система переходить у певні (ізольовані) моменти часу, коли її характеристики змінюються стрибкоподібно. Для зміни стану необхідна певна причина – наприклад, надходження нового вхідного сигналу. З погляду моделювання, основну

увагу слід приділяти саме моментам зміни характеристик, тобто переходам системи з одного особливого стану в інший.

В одному елементі системи можуть відбуватися активні та пасивні події, такі як:

- початок розв'язання задачі на комп'ютері – подія пасивна, оскільки вона походить від іншої події: надходження запиту на рішення;
- завершення рішення – подія активна, оскільки залежить від внутрішніх властивостей комп'ютера (продуктивності);
- надходження заявок на обслуговування в чергу та залишення черги – події пасивні, але відхід з черги «нетерплячих» заявок – події активні.

Зазвичай елемент, в якому відбувається активна подія, називають активним, пасивна подія – пасивним.

Основу імітаційної моделі системи з просуванням модельного часу за подіями становлять моделі активних подій. Пасивні події моделюються, якщо виявлено їх причинно-наслідкові відносини з активними подіями.

При використанні методу просування часу на основі подій, спочатку модельний час встановлюється на нуль, після чого визначається момент настання майбутніх найближчих подій. Потім годинник модельного часу пересувається до часу найближчої події, і в цей момент відбувається оновлення стану системи з урахуванням цієї події та коригування інформації про майбутні події. Далі модельний час знову просувається до наступної найближчої події, після чого повторюється оновлення стану системи та уточнення моментів майбутніх подій – і так далі.

Схему алгоритму моделювання з просуванням часу від події до події наведено на рис. 3.3.

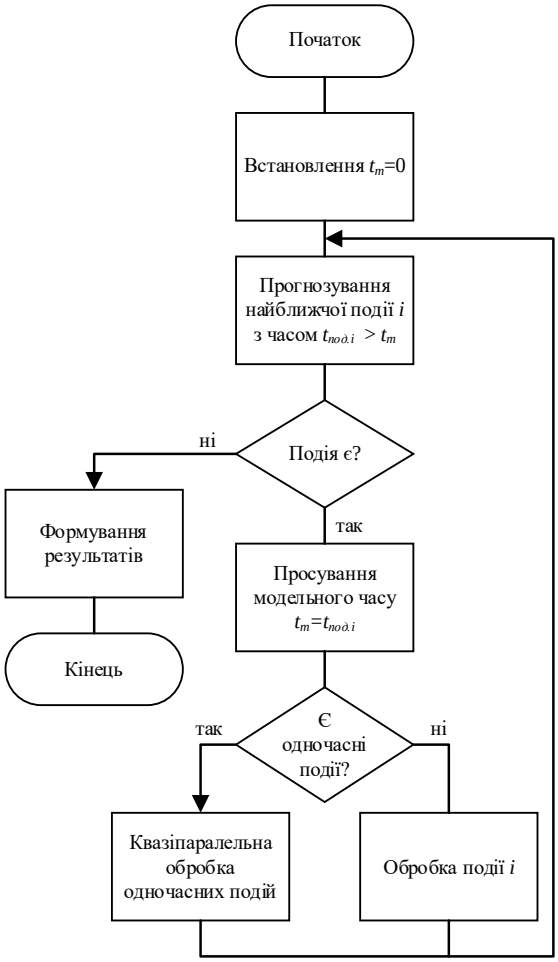


Рисунок 3.3 – Схема алгоритму моделювання з просуванням модельного часу до чергової події

Процес моделювання триває доти, доки не виконається певна умова завершення моделювання, задана заздалегідь. Оскільки в дискретно-подієвих імітаційних моделях усі зміни відбуваються лише в моменти настання подій, періоди бездіяльності системи просто пропускаються, а модельний час перескакує від однієї події до іншої. Слід зазначити, що тривалість інтервалів між подіями може змінюватися – вона не є сталою.

У системі може одночасно протікати кілька процесів. При моделюванні без використання реальної паралельності ця ситуація вирішується шляхом псевдопаралельної імітації кількох процесів, тобто послідовно в установленому порядку реалізуються реакції на всі одночасні події без просування модельного часу.

У моделюванні **з фіксованим кроком часу** модельний годинник просувається на сталу величину Δt, обрану заздалегідь. Слід зауважити, що в загальному випадку $t_m \neq t$ і $\Delta t_m \neq \Delta t$, а мають деякі пропорційні значення. Після кожного такого кроку перевіряється, чи відбулися якісь події протягом останнього інтервалу часу. Якщо події були заплановані в цьому проміжку, вважається, що вони відбулися наприкінці інтервалу, і відповідним чином оновлюється стан системи та накопичується статистика.

Схему алгоритму моделювання з просуванням часу від події до події наведено на рис. 3.4.

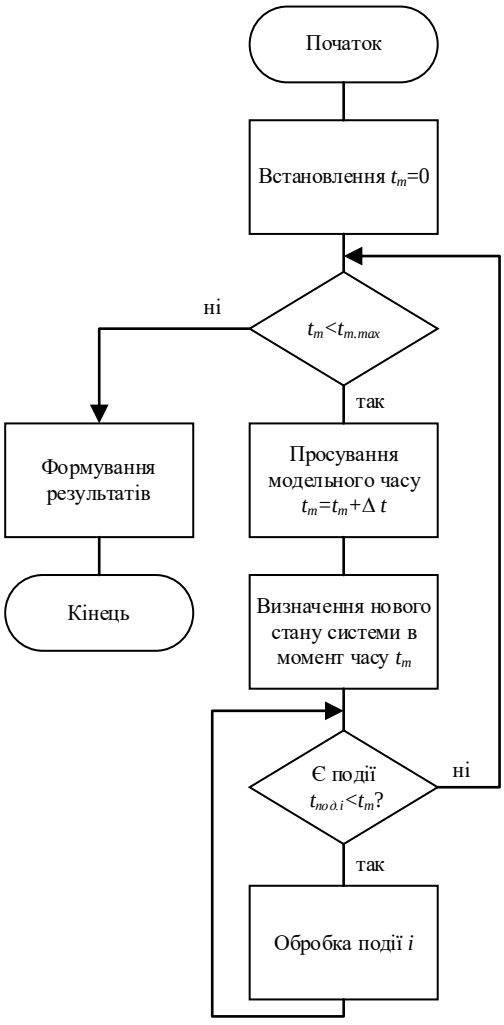


Рисунок 3.4 – Схема алгоритму моделювання з просуванням фіксованими кроками

У випадках, коли передбачається, що дві або більше подій відбуваються в один і той самий момент часу, необхідно застосовувати набір

правил, що дозволяють визначити порядок обробки цих подій. Таким чином, просування часу за допомогою постійного кроку має два основні недоліки:

- виникнення похибок, пов'язаних з обробкою подій наприкінці інтервалу, протягом якого вони насправді сталися;
- необхідність визначення, яку подію обробляти першою, у випадку коли події, що в дійсності мали різний час настання, вважаються такими, що відбулися одночасно.

Такого роду проблеми можна частково вирішити, зменшивши тривалість інтервалу Δt, однак це призведе до збільшення кількості перевірок на настання подій, що, у свою чергу, викличе зростання часу виконання обчислень. При просуванні часу фіксованими кроками навіть «порожні» періоди обробляються, що потребує більше комп'ютерних ресурсів. Враховуючи це, просування часу за постійним кроком не застосовується в дискретно-подієвих імітаційних моделях, коли інтервали часу між послідовними подіями можуть значно відрізнятися за тривалістю.

Загалом такий підхід застосовується до систем, в яких припускається, що всі події фактично відбуваються в один із моментів часу: t_1, $t_1 + \Delta t$, $t_1 + 2\Delta t$, $t_1 + 3\Delta t$ … Наприклад, в економічних системах дані часто подаються за річні проміжки часу, тому логічно встановити крок просування часу в моделі рівним одному року. Варто також зазначити, що просування часу за допомогою постійного кроку можна реалізувати і через механізм просування часу від події до події, якщо заздалегідь планувати виникнення подій через Δt одиниць часу – цей підхід є різновидом механізму просування часу «від події до події».

Однозначних вказівок щодо вибору того чи іншого способу просування модельного часу не існує. Загалом можна сказати, що якщо зміни стану в модельованій системі відбуваються часто й регулярно, а обмежень на використання обчислювальних ресурсів немає, то цілком прийнятним є метод просування модельного часу фіксованими кроками Δt.

Однак, якщо стани змінюються зрідка та нерегулярно, а також висуваються високі вимоги до точності моделювання, доцільніше використовувати підхід на основі подій – просування часу стрибками до моменту настання найближчої події.

Підходи для вибору кроку моделювання:

- приймати величину кроку рівну середній інтенсивності виникнення подій різних типів;
- вибирати величину кроку, що дорівнює інтервалу між найбільш частими (або найбільш важливими) подіями.

На рис. 3.5 наведено часові діаграми просування часу для описаних підходів (для наочності $t_m = t$).

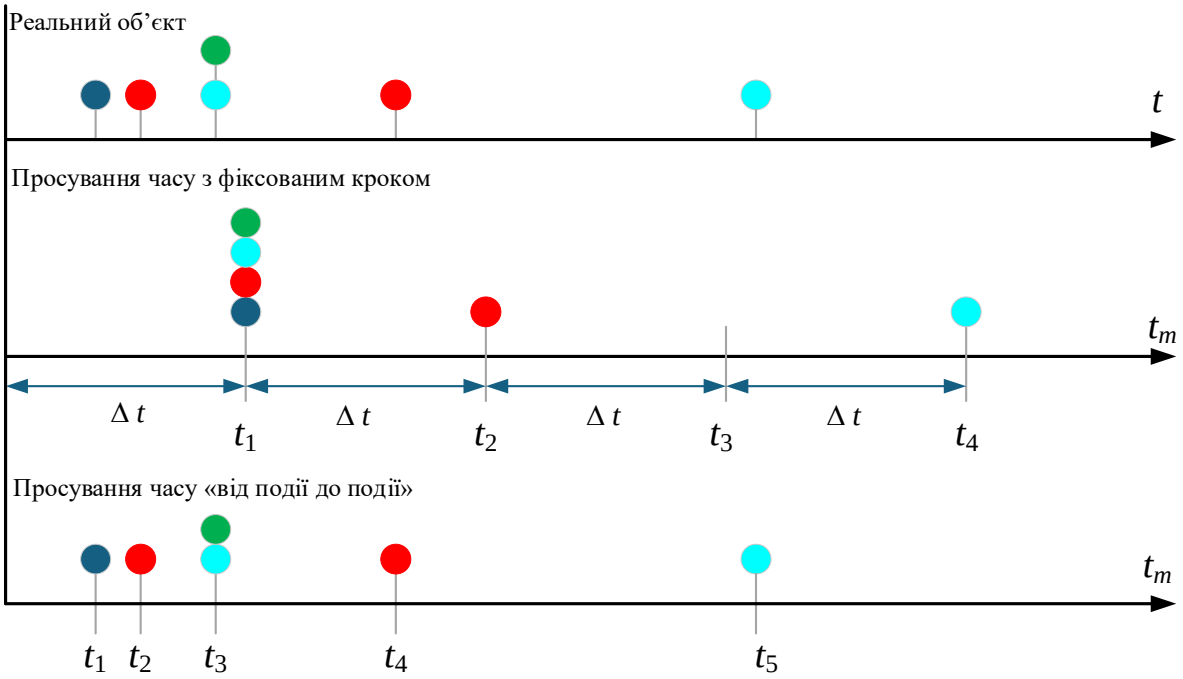


Рисунок 3.5 – Часові діаграми просування часу в моделі способом «від події до події» та з фіксованим кроком часу

Слід зауважити, що події, які виникають одночасно в реальному об'єкті, під час імітаційного моделювання обробляються послідовно (момент t_1 для просування часу з фіксованим кроком і t_3 – «від події до події» на рис. 3.5), тобто необхідно визначити послідовність їх обробки. Також можна побачити, що в момент часу t_3 для просування часу з фіксованим кроком не

обробляється жодна подія – «порожній» період, на обробку якого витрачаються ресурси обчислювальної техніки, а в момент часу t_4 (також для просування часу з фіксованим кроком) виконується обробка події зі значною затримкою від її настання в реальному об'єкті.

Існують імітаційні моделі, які призначені для роботи в реальному (природному) масштабі часу. Але і в цьому випадку модельний час просувається стрибками – за подіями або комбінованим методом. Це, своєю чергою, породжує проблему синхронізації модельного та реального часу.

Також існує поняття розподіленого імітаційного моделювання, яке з'явилось як відповідь на потребу у великій обчислювальній потужності для складних моделей. Під розподіленим імітаційним моделюванням розуміють виконання єдиної імітаційної програми на мультипроцесорній або мультикомп'ютерній системі. Послідовну модель можна адаптувати до паралельної архітектури, що дозволяє прискорити обчислення завдяки паралельній обробці подій, які заплановані на один і той самий момент модельного часу. У розподіленому підході паралельно опрацьовуються події, заплановані на різні відрізки модельного часу.

Приклад розподіленого моделювання – моделювання інфраструктури, де об'єднуються економічні, екологічні, транспортні і інші підмоделі. Такі підсистеми можуть виконуватись на географічно розподілених гетерогенних обчислювальних комплексах.

Існує також комбінований метод просування модельного часу Delft [1], який поєднує два підходи до просування модельного часу – на основі подій та з фіксованим кроком. Він дозволяє просувати модельний час як за розкладом подій, так і за фіксованими часовими інтервалами (кроками). За допомогою алгоритму Delft можна достатньо точно обробляти події, не втрачаючи регулярності часових кроків, що робить його надзвичайно ефективним для моделювання складних систем із динамічними і нерівномірними процесами.

Алгоритм Delft працює за такими принципами:

- просування за подіями – якщо запланована подія має настати раніше, ніж наступний фіксований момент часу, модельний час пересувається до цієї події, і вона обробляється;
- дискретне просування – якщо подій найближчим часом немає, модельний час збільшується на фіксований крок Δt.

Це дозволяє уникнути втрати точності, коли події трапляються між фіксованими часовими кроками.

Алгоритм Delft ефективний у випадках, коли необхідно:

- поєднати обидва підходи (наприклад, у фізичних моделях, де важливі події – це спрацювання датчиків, а стан системи оновлюється з певною частотою);
- забезпечити гнучке управління часом моделі, що актуально при моделюванні складних процесів з нерівномірною динамікою.

На відміну від просування за подіями, Delft дозволяє зберігати рівномірність часових кроків, що зручно для статистичного аналізу вихідних даних. А порівняно з традиційним фіксованим кроком Δt, Delft забезпечує вищу точність, адже дозволяє не пропустити події, що відбуваються між моментами часу.

Основні етапи роботи алгоритму Delft:

- Ініціалізація модельного часу та встановлення кроку Δt.
- Запуск циклу моделювання:
  - Перевірка, чи є події, що повинні трапитись до наступного кроку часу $t + \Delta t$.
    - Якщо події є – час просувається до найближчої події, вона обробляється.
    - Якщо подій немає – час збільшується на Δt.
- Процес триває до досягнення заданого кінцевого моменту моделювання.

Контрольні запитання

1. У чому полягає сутність процесу імітаційного моделювання?
2. Які є особливості експериментування з реальними об’єктами?
3. Коли доцільно проводити імітаційне моделювання?
4. Чим відрізняється статичний опис системи від динамічного опису?
5. З чого складається модель?
6. Що таке екзогенні та ендогенні змінні?
7. Яким критеріям повинна відповідати модель?
8. Які існують методи просування модельного часу?
9. Опишіть основні ідеї покрокового просування модельного часу.
10. Опишіть основні ідеї подійного просування модельного часу.
11. Що таке агентне моделювання?
12. Назвіть основні етапи імітаційного моделювання.
13. Які помилки виникають під час імітаційного моделювання?
14. Що перевіряється під час проведення імітаційного моделювання?
15. Що таке випадковий процес?
16. Що таке перетин випадкового процесу?
17. Чим відрізняються випадкові процеси з дискретним часом від процесів з неперервним часом?

18 Які стани можуть бути у випадкового процесу?

19. Що таке марковський випадковий процес?
20. Які процеси відносяться до стаціонарних випадкових процесів?
21. Опишіть послідовність дій при моделюванні з просуванням модельного часу до чергової події.
22. Опишіть послідовність дій при моделюванні з просуванням фіксованими кроками.

4. ПЛАНУВАННЯ ЕКСПЕРИМЕНТІВ

4.1. Сутність і цілі планування експериментів

Цілі моделювання досягаються шляхом дослідження створеної моделі за допомогою експериментів, під час яких визначаються вихідні характеристики при різних значеннях параметрів.

Експеримент – це особливий вид наукової діяльності, спрямований на пізнання властивостей об’єктів або систем, а також виявлення закономірностей, що визначають взаємозв’язки між ними. У ході експерименту дослідник активно впливає на об’єкт за допомогою спеціальних методів, засобів або приладів і може цілеспрямовано змінювати умови з метою отримання результатів. Якщо під час спостереження дослідник лише фіксує явища ззовні, то експеримент дозволяє глибше зануритися в сутність процесу, що вивчається.

Експерименти бувають натурними та модельними, активними і пасивними.

Натурний експеримент передбачає вивчення реального явища чи процесу в природному або виробничому середовищі. Такий підхід дозволяє уточнити параметри об’єкта, перевірити його поведінку в нестандартних або аварійних умовах, оцінити надійність. При цьому враховується широкий спектр зовнішніх впливів на об’єкт. Однак натурні експерименти потребують значних ресурсів і часто супроводжуються підвищеним ризиком, тому використовуються лише у виняткових випадках.

Модельний експеримент проводиться на спеціально створених моделях, що відтворюють об’єкт або процес. Такий підхід дозволяє детально дослідити об’єкт і супутні процеси, багаторазово повторювати експеримент у контрольованих умовах, що майже неможливо при натурному дослідженні.

Активний експеримент передбачає алгоритмізований вибір значень керованих параметрів об’єкта, що досліджується, серед усіх їх можливих

комбінацій. Такий підхід дозволяє істотно підвищити точність побудованої моделі та зменшити кількість необхідних експериментів у 5-10 разів порівняно з традиційними методами.

Пасивний експеримент ґрунтується на інтуїтивному виборі значень керованих параметрів. При цьому експериментальні значення обираються без чіткої системи. Це ускладнює як математичне моделювання, так і подальше практичне використання отриманих моделей.

Проведення експериментальних досліджень включає низку пізнавальних дій, серед яких:

- формулювання мети експерименту на основі наявних наукових теорій, з урахуванням практичних потреб і напрямів розвитку науки;
- теоретичне обґрунтування умов, за яких буде проводитись експеримент;
- розроблення принципів організації експерименту та підбір необхідних технічних засобів;
- здійснення спостережень, вимірювань і фіксація властивостей, взаємозв’язків та динаміки змін об’єкта, що досліджується;
- обробка отриманих результатів із застосуванням статистичних методів.

Методологія проведення традиційного експерименту включає такі основні етапи:

1) розробка програми експерименту;

2) вибір засобів для проведення експерименту;

3) проведення експерименту;

4) обробка та аналіз експериментальних даних.

Під час проведення експерименту необхідно дотримуватись таких загальних вимог:

- об’єкт дослідження повинен бути описаним у вигляді системи змінних параметрів, що відображають його функціонування;
- мають бути доступні якісні та кількісні вимірювання параметрів, які впливають на об’єкт;
- необхідно описати умови існування та використання об’єкта (наприклад, галузь, тип виробництва, умови праці);
- слід мати чітко сформульовану експериментальну гіпотезу, яка лежить в основі дослідження;
- усі поняття, що використовуються в межах гіпотези, мають бути точно визначені;
- потрібно надати детальний опис специфічних умов проведення експерименту: місця, часу, обладнання тощо.

Експериментальні дослідження зазвичай проводяться в спеціально обладнаних приміщеннях – лабораторіях. Частину лабораторії або виробничого приміщення, де встановлено експериментальне обладнання, називають робочим простором. Під час виконання експерименту необхідно суворо дотримуватися норм промислової санітарії, правил техніки та пожежної безпеки.

Необхідною вимогою при проведенні досліджень є ведення журналу, форма якого може бути довільною, однак бажано, щоб вона була адаптована до умов експерименту та дозволяла зручно фіксувати значення всіх вимірюваних параметрів. У журналі обов’язково зазначаються дата проведення досліду, опис впливу окремих факторів, значення вимірюваних параметрів, а також марка і тип приладу, яким проводились вимірювання.

Результати лабораторних та більшості виробничих експериментів оформлюються у вигляді протоколу. Його підписують експериментатор і керівник лабораторії, виробництва або дослідницької групи. У навчальному процесі протокол лабораторної роботи підписують студент і викладач.

Після завершення експерименту отримані результати обробляються за допомогою методів математичної статистики. Результати мають бути представлені у вигляді таблиць, графіків, діаграм тощо – відповідно до цілей і завдань дослідження.

Проведення експериментів повинно бути чітко спланованим. При цьому кожен комп'ютерний експеримент проводиться для конкретного набору значень параметрів. Особливо важливим це стає в умовах числового або статистичного імітаційного моделювання на універсальних обчислювальних системах, адже кількість можливих комбінацій керованих параметрів може бути дуже великою. Так, кількість всіх можливих комбінацій визначається як $N_з = q^n$, де q – кількість можливих значень кожного параметра, n – кількість керованих параметрів. Зрозуміло, що за умови обмежених часових чи обчислювальних ресурсів провести всі можливі експерименти неможливо. Тому виникає потреба у виборі певних комбінацій параметрів і правильному порядку проведення експериментів. Цей процес називають стратегічним плануванням.

Планування розпочинається ще на початкових етапах створення моделі, коли визначаються ключові параметри, що впливають на ефективність функціонування системи. Такі параметри називають **факторами**. **Відгук** – змінна, залежність якої від інших змінних досліджується. Відгук є функцією факторів. Фактори – це аргументи відгуку. Далі встановлюються можливі значення кількісних параметрів і варіанти якісних – так звані **рівні** q. Кількість рівнів варіювання – це кількість різних значень, які приймає фактор в експерименті. Для випадку неоднакової кількісті рівнів кожного фактору $N_з = q_1 \cdot q_2 \cdot \ldots \cdot q_k$, де k – число факторів.

Особливо ретельно потрібно планувати експерименти під час дослідження нестаціонарних систем, оскільки в таких випадках необхідне вибіркове збільшення кількості запусків моделі.

Методи скорочення тривалості моделювання за умови збереження достовірності результатів імітаційного моделювання називають тактичним

плануванням. На тривалість одного експерименту впливають ступінь стаціонарності системи, взаємозв'язки характеристик та початкові умови моделювання.

Дані, зібрані під час експерименту, можна розглядати як часові ряди, які складаються з вимірювань певних характеристик. Наприклад, ряд значень деякої величини y може розглядатись як вибірка з випадкової послідовності. Якщо ця послідовність є стаціонарною, її математичне сподівання $\overline{y}$ не залежить від часу. Оцінкою $\overline{y}_N$ є середнє значення по часовому ряду $y_1, \ldots, y_N$.

Якщо задано максимально допустиму похибку оцінки довірчого інтервалу та мінімальну ймовірність того, що істинне середнє $\overline{y}$ потрапляє в цей інтервал, тоді існує мінімальний обсяг вибірки, що відповідає мінімальній тривалості експерименту. При оцінюванні кількох характеристик період моделювання обирають за найбільшим значенням. Необхідна кількість спостережень залежить від дисперсії оцінюваної характеристики: що вона більша – то більша має бути вибірка. Для корельованих випадкових характеристик потрібно додатково оцінювати дисперсію. Існують спеціальні методи обробки результатів моделювання, що враховують апріорну інформацію про систему і дозволяють зменшити обсяг вибірки без втрати точності. До них належать методи корельованих, стратифікованих вибірок тощо.

Більшість імітаційних моделей використовують для вивчення усталених рівноважних режимів роботи систем. Однак у початковий момент система проходить через перехідний режим, тривалість якого може бути значною. Значення вихідних характеристик, отримані в цей період, можуть зсувати загальну оцінку.

Є три основні способи зменшення похибок, пов'язаних з початковими умовами:

- збільшення періоду моделювання – зі зростанням кількості вимірювань вплив початкового зміщення зменшується;
- зсув початку збору статистики далі від нульового моменту часу;
- старт моделювання не з «нульового», а з попередньо визначеного стану, наближеного до усталеного режиму.

У випадку моделювання нестаціонарних процесів усталений режим може взагалі бути відсутнім. Найбільш природним способом оцінки характеристик таких моделей є метод повторних експериментів, що потребує спеціального підходу до планування.

При обґрунтуванні вибору виду плану статистичні його характеристики є вторинними для прийняття рішення, основним є умови проведення експерименту і наявність апріорної інформації про досліджуваний процес.

Багатофакторні статистичні експерименти є основним методом дослідження поведінки складноорганізованих систем. Їх особливість полягає в тому, що всі досліджувані фактори варіюються одночасно в межах одного експерименту, а вплив сторонніх (заважаючих) факторів зводиться до випадкових впливів. Такий підхід кардинально відрізняється від класичних однофакторних експериментів, для яких характерна необхідність проведення великої кількості спостережень та складність стабілізації сторонніх факторів. Багатофакторні експерименти також відзначаються тим, що зі збільшенням кількості досліджуваних факторів підвищується точність отриманих результатів.

На сьогодні не існує єдиної чітко визначеної класифікації статистичних експериментів, однак численні наукові публікації дозволяють певною мірою їх систематизувати. Як і в інших сферах наукових досліджень, класифікація експериментів доцільно проводиться за різними ознаками. До основних критеріїв класифікації можна віднести такі:

1. За типом математичних моделей, що використовуються для планування експерименту та статистичної обробки його результатів:

- плани дисперсійного аналізу;
- плани регресійного аналізу;
- плани коваріаційного (кореляційного) аналізу.

2. За кількістю рівнів факторів, що досліджуються:
   - двохрівневі плани, побудовані на основі лінійних моделей;
   - трьох- та багато-рівневі плани, побудовані на основі моделей другого та вищих порядків.
3. Залежно від цілей експерименту:
   - плани для виявлення суттєвих факторів та відсіювання незначущих;
   - плани для побудови моделі, що описує поведінку системи (процесу);
   - плани для пошуку оптимальних умов функціонування системи (дослідження поверхні відгуку);
   - спеціалізовані плани для вирішення конкретних задач.
4. За характером охоплення факторного простору:
   - повнофакторні плани;
   - неповнофакторні плани;
   - композиційні плани;
   - інші спеціальні типи планів.
5. За критеріями оптимальності, що застосовуються при плануванні:
   - ортогональні плани;
   - рототабельні плани;
   - D-оптимальні плани.
6. Залежно від обмежень на послідовність проведення дослідів:
   - плани без обмежень щодо рандомізації;
   - плани з обмеженнями щодо рандомізації.

7. Залежно від кількості ступенів свободи f (обчислюється як різниця між кількістю експериментальних точок у плані та кількістю параметрів

моделі, що визначаються незалежно), які враховуються під час статистичної обробки:

- ненасичені плани ($f > 0$);
- насичені плани ($f = 0$);
- перенасичені плани ($f < 0$).

Плани дисперсійного аналізу призначені для вивчення впливу як якісних, так і кількісних факторів на досліджувану ознаку. Крім визначення значущості та характеру впливу різних факторів, такі плани використовуються також як неповнофакторні, тобто для скорочення обсягу експериментальних досліджень. Наприклад, для цього застосовують латинські або греко-латинські квадрати. Ці плани можуть бути основою так званих «відсіювальних експериментів».

Плани регресійного аналізу набули найбільшого поширення в практиці експериментальних досліджень. Їх застосовують для вивчення кількісних факторів, коли рівні факторів є фіксованими значеннями.

Плани коваріаційного (кореляційного) аналізу використовуються для дослідження як кількісних, так і якісних факторів, причому рівні кількісних факторів можуть бути як випадковими, так і фіксованими.

Серед основних задач, які вирішують за допомогою статистичних методів експериментів, важливе місце займає визначення значущих факторів та відсіювання несуттєвих. Для цього використовують як дисперсійні, так і регресійні плани. Зазвичай, такі дослідження проводяться з обмеженою кількістю експериментів, тому застосовуються неповнофакторні плани та спеціальні плани для відсіювання (наприклад, плани випадкового балансу). Плани для відсіювання несуттєвих факторів зазвичай є насиченими або перенасиченими.

Існує також велика група планів, призначених для вивчення механізмів явищ. Сюди відносять, наприклад, плани для побудови інтерполяційної моделі системи – як у випадку, коли структура моделі відома, але потрібно

визначити коефіцієнти, так і у випадку, коли сам вигляд моделі невідомий. Також сюди належать плани для уточнення параметрів відомої моделі або для вибору найкращої моделі серед кількох можливих варіантів. Найчастіше для таких завдань застосовують регресійні плани.

Ще одним важливим напрямом є дослідження поверхні відгуку, тобто пошук оптимальних умов роботи системи чи протікання процесів. Тут зазвичай використовують регресійні плани другого, рідше третього порядку. Найпоширенішими є рототабельні композиційні плани другого порядку та плани Гертлі та Вестлейка.

Повнофакторні плани передбачають вивчення всіх можливих комбінацій рівнів факторів. Вони дозволяють досліджувати як основні ефекти, так і їхню взаємодію. Проте часто взаємодії факторів вищих порядків виявляються несуттєвими, тому такі плани стають ненасиченими і, відповідно, неекономічними за обсягом необхідних спостережень.

Саме через ці причини часто віддають перевагу неповнофакторним планам, до яких, зокрема, належать і часткові репліки (фракційні плани). Ступінь скорочення кількості експериментів визначається як специфікою дослідницької задачі, так і наявною інформацією про систему чи процес.

Для об’єктів дослідження, результати експериментів яких характеризуються великою дисперсією відтворюваності (наприклад, у виробничих умовах), застосовують робастні плани експериментів, відомі також як плани Тагучі [18]. Такі плани розроблені для забезпечення стійких результатів навіть при впливі неконтрольованих факторів (зовнішніх збурень або шумів). Вони дають змогу мінімізувати вплив некерованих змінних і підвищити надійність процесів чи систем без значного збільшення обсягу експериментів. Для робастних планів характерним є рівномірне розміщення точок у багатовимірному факторному просторі, а також мінімальна корельованість між усіма ефектами – як основними, так і ефектами взаємодії. Основною метою побудови таких планів є створення оптимальних умов для виявлення структури рівняння регресії. Виконання цієї вимоги забезпечує

стійкість оцінок коефіцієнтів регресії та дотримання ключових передумов, необхідних для коректного застосування методів регресійного аналізу.

4.2. Елементи стратегічного планування експериментів

Терміни «стратегічний» і «тактичний» часто вживають як синоніми, хоча насправді між ними існує суттєва різниця [10]. Стратегія – це загальний план досягнення довгострокових цілей експерименту, що визначає основні напрямки, підходи та логіку дослідження. Натомість, тактика стосується конкретних дій, необхідних для реалізації стратегічного плану. Вона включає вибір конкретних параметрів, методик, послідовності проведення дослідів, засобів контролю тощо.

Стратегічне планування дозволяє відповісти на питання, при якому поєднанні рівнів зовнішніх і внутрішніх факторів може бути отримана найбільш повна і достовірна інформація про поведінку системи. Тактичне планування ж забезпечує досягнення заданих точності і достовірності результатів.

Формування стратегічного плану експерименту відбувається в так званому **факторному просторі**. Під факторним простором розуміють сукупність усіх зовнішніх і внутрішніх параметрів, значення яких дослідник може змінювати або контролювати під час підготовки та проведення експерименту.

Схематично базовий експеримент наведено на рис. 4.1.

Математичні методи планування експериментів базуються на кібернетичному підході до опису процесу проведення експерименту. Під час проведення експерименту об’єкт дослідження піддається впливу одного чи декількох керованих факторів, щоб вивчити його реакцію або відгук. Однак важливо також враховувати некеровані чинники, які можуть прямо або опосередковано впливати на результати експерименту. Такі неконтрольовані впливи ще називають збурюючими факторами або фоновими діями. Їхній

вплив може викликати спотворення результатів, тому їх слід по можливості виявляти, оцінювати або мінімізувати.

Рисунок 4.1 – Схематичне подання експерименту

До основних об'єктів стратегічного планування належать:

- вхідні змінні (фактори, екзогенні параметри) $x_1, x_2, x_3, \ldots, x_n$;
- вихідні змінні (відгуки, реакції, ендогенні параметри) $y = f(x_1, x_2, x_3, \ldots, x_n) + \xi = f(x) + \xi$, де $f(x)$ – оператор, що моделює дію реальної системи, визначає залежність вихідної змінної y від чинників x_i (модель процесу, що протікає в системі); ξ – помилка, перешкода, спричинена наявністю випадкових факторів;
- рівні факторів (тобто конкретні значення параметрів, які використовують у ході експерименту).

Успіх та достовірність дослідження значною мірою залежать від стратегічного планування експерименту, яке базується на кількох ключових принципах, дотримання яких дозволяє зменшити кількість помилок і забезпечити достовірність та відтворюваність результатів експерименту:

- чітке визначення цілей експерименту та конкретних показників, які необхідно виміряти або покращити. Формулювання конкретних, вимірюваних і перевірюваних гіпотез для подальшої перевірки в ході дослідження;

- вибір типу експериментального дизайну, що найкраще відповідає цілям (випадковий розподіл, квазіексперимент, перехресний дизайн тощо). Також вирішується питання про необхідність створення контрольних груп для порівняння з експериментальними результатами;
- визначення екзогенних та ендогенних змінних, значення яких буде вимірюватися. Виділення контрольованих змінних, які потрібно тримати незмінними, щоб уникнути їх впливу на результати;
- розрахунок оптимального розміру вибірки для досягнення статистичної значущості результатів. Вибір методу відбору вхідних значень (випадкова, стратифікована вибірка тощо);
- чіткий опис процедур проведення експерименту для забезпечення його повторюваності. Підготовка інструментів, обладнання та програмного забезпечення, що будуть використовуватись під час дослідження;
- визначення методів збору даних. Встановлення відповідних методів статистичного аналізу для подальшого трактування результатів;
- ретельна оцінка потенційних ризиків, пов’язаних із проведенням експерименту, та розробка плану управління ними для їх мінімізації;
- документування кожного етапу експерименту. Після завершення дослідження складається детальний звіт, що містить опис методів, результати та їх інтерпретацію.

Додатково, якщо експеримент стосується людей або тварин, обов’язково отримується інформована згода всіх учасників та етичне схвалення.

Під час стратегічного планування комп'ютерних експериментів з моделями виникає чимало взаємопов'язаних питань. Вони стосуються як особливостей роботи самої модельованої системи, так і нюансів реалізації моделі в обчислювальному середовищі та обробки отриманих результатів. Під час стратегічного планування доводиться вирішувати основні три проблеми.

Перша проблема. Під час стратегічного планування експериментів із використанням моделей систем важливим є визначення вихідних параметрів (відгуків), тобто тих характеристик, які потрібно вимірювати в ході експерименту для отримання необхідних результатів. Вибір таких параметрів цілковито залежить від цілей дослідження. Наприклад, при моделюванні інформаційно-пошукової системи може цікавити або час відповіді системи на запит, або максимальна кількість оброблених запитів за певний період часу, або ж і те, і інше одночасно. Таких параметрів може бути кілька.

Друга проблема стратегічного планування – вибір ключових факторів та їх комбінацій, що впливають на роботу модельованого об'єкта. Фактори можуть бути різними: напруга живлення, температура, вологість, стабільність постачання комплектуючих тощо. Зазвичай кількість факторів велика, особливо коли система ще мало вивчена. За правилом Парето при системному аналізі приблизно 20 % факторів визначають 80 % властивостей системи і, відповідно, решта 80 % факторів впливають лише на 20 % її характеристик. Тому важливо вміти виділити найбільш значущі фактори, що потребує глибокого аналізу об'єкта та процесів у ньому.

Фактори можуть бути:

- кількісними (наприклад, інтенсивність потоків, об'єм буферів, кількість каналів);
- якісними (наприклад, дисципліни обслуговування черг, кваліфікація персоналу).

Ключовою вимогою до факторів є можливість їх регулювання, тобто можливість встановити та підтримувати значення згідно з планом

експерименту. Проте існують і некеровані фактори – наприклад, вплив навколишнього середовища.

Для сукупності факторів важливо забезпечити:

- сумісність – усі можливі комбінації значень мають бути реальними;
- незалежність – значення одного фактора можна змінювати незалежно від інших.

Третьою проблемою є вибір кількості рівнів для кожного фактора. Для економії часу і ресурсів варто обирати мінімально необхідну кількість рівнів, але достатню для досягнення необхідної точності експерименту. Мінімальна кількість рівнів – два.

Зручно, коли всі фактори мають однакову кількість рівнів – такий план називають симетричним. Для полегшення обробки даних рівні факторів варто розташовувати рівновіддалено один від одного – такі плани називають ортогональними.

Якщо експеримент містить всі можливі комбінації рівнів усіх факторів, він називається повним факторним експериментом (ПФЕ). ПФЕ дає максимально повну інформацію, але потребує значних ресурсів. Число вимірювань моделі при ПФЕ розраховується як добуток кількостей рівнів по всіх факторах.

При комп'ютерному моделюванні в кожній точці плану потрібно виконати кілька прогонів моделі для досягнення необхідної точності результатів. Кількість прогонів визначається на етапі тактичного планування. Якщо позначити кількість прогонів у кожній точці плану через p, то для симетричного ПФЕ загальна кількість запусків моделі дорівнюватиме добутку всіх рівнів факторів і p: $N = p \cdot q_1 \cdot q_2 \cdot \ldots \cdot q_k = p \cdot q^k$.

4.3. Елементи тактичного планування експериментів

Тактичне планування експерименту при імітаційному моделюванні містить специфічні аспекти, пов'язані з використанням комп'ютерних систем

і програмного забезпечення. Воно передбачає створення детального документа, який описує всі операції в цифровому середовищі, включаючи використання конкретних програмних засобів, налаштування систем і порядок виконання задач.

Стандартизація процедур необхідна для мінімізації впливу зовнішніх факторів і підвищення відтворюваності експерименту. Основні етапи виглядають так:

- важливо визначити та підготувати всі необхідні програмні інструменти та комп'ютерне обладнання, перевіривши їх працездатність, сумісність і актуальність версій. Також потрібно призначити відповідальних осіб за різні аспекти експерименту, забезпечивши їх необхідною підготовкою та інструктажем щодо роботи з відповідним програмним забезпеченням та обладнанням;
- планування збору даних включає вибір методів і засобів, таких як бази даних і лог-файли. Важливим є створення детального графіка проведення експерименту із зазначенням термінів і відповідальних осіб, а також передбачення автоматизації процесів, де це можливо;
- для забезпечення надійності і точності даних необхідно провести пілотне тестування, щоб перевірити всі цифрові процедури збору інформації, включаючи тестування програмних модулів і систем. Введення заходів контролю якості, таких як перевірка даних на помилки, повноту і коректність, а також використання методів верифікації та валідації, гарантує достовірність результатів;
- управління ризиками включає оцінку потенційних загроз, пов'язаних із технічними аспектами, наприклад, збоями в роботі програмного забезпечення, поломками обладнання, кіберзагрозами тощо. Розробка стратегій з мінімізації та

управління виявленими ризиками, включаючи резервні плани, створення резервних копій даних та підготовку додаткових ресурсів, дає змогу знизити потенційні ризики для експерименту;

- документування і звітність передбачають ведення детальної документації всіх цифрових процедур та етапів експерименту, включаючи лог-файли, вихідний код, конфігураційні файли та журнали дій. Підготовка проміжних та фінальних звітів, що містять опис методів, хід експерименту, використані цифрові інструменти, результати та їх інтерпретацію, є завершальним етапом цього процесу.

Нарешті, етичні аспекти включають отримання інформованої згоди від усіх учасників експерименту, якщо це необхідно, із детальним поясненням цілей, процедур та можливих ризиків. Гарантування конфіденційності даних учасників, включаючи їх анонімізацію, шифрування та дотримання всіх чинних норм і стандартів захисту даних, також є обов’язковою частиною тактичного планування експерименту.

Тактичне планування насамперед спрямоване на вирішення проблеми точного оцінювання імітаційних моделей і пов’язане з тим, що у випадку стохастичної моделі для досягнення заданої точності результатів експериментів зазвичай необхідно виконувати багаторазові прогони (реалізації). Однак час, необхідний для серії прогонів складного модельного експерименту, може бути значним, тоді як доступний машинний час обмежується наявними ресурсами та часовими рамками експерименту. Основне протиріччя полягає в суперечності між бажаною точністю результатів та обмеженими ресурсами (витратами машинного часу і ресурсами, необхідними для проведення серії експериментів). На практиці завжди шукають компроміс між цими вимогами.

Експериментатор має не лише отримати дані, але й оцінити їх точність, тобто ступінь довіри до висновків, які буде зроблено на основі цих результатів. Тому є очевидним прагнення збільшити тривалість окремого

прогону або кількість прогонів (обсяг вибірки), адже саме від цього безпосередньо залежить точність отриманих результатів.

Кількість задач, які вирішується при тактичному експерименті велика:

- визначення тривалості імітаційного прогону або необхідної кількості повторень кожного прогону (обсягу вибірки), що забезпечує задану точність результатів моделювання;
- визначення тривалості перехідного режиму (аналіз встановленого стану) та задання початкових умов (початкового стану системи);
- вибір коректного кроку моделювання, оскільки саме від нього залежить точність відтворення в моделі тих подій, які реально відбуваються в системі;
- вибір послідовностей подій;
- контроль повторюваності результатів для перевірки стабільності моделі;
- встановлення правил зупинки моделювання;
- зменшення дисперсії вихідних параметрів за допомогою спеціальних методів;
- зменшення похибок моделювання, пов’язаних з використанням генераторів псевдовипадкових чисел у моделі;
- інші супутні задачі, що забезпечують точність, достовірність та ефективність імітаційного експерименту.

Всі задачі можуть бути поєднані у вирішення чотирьох ключових проблем.

Перша проблема – проблема визначення початкових умов і їхнього впливу на досягнення сталого результату під час моделювання.

Усталеним (стаціонарним) називають такий стан моделі, коли послідовні спостереження за відгуком у цьому стані мають певний граничний стаціонарний розподіл ймовірностей і не залежать від часу.

Стаціонарність режиму моделювання свідчить про те, що в моделі системи встановилась рівновага процесів. Часто говорять, що динамічна система знаходиться в рівноважному (стаціонарному) стані, якщо її функціонування відбувається відповідно до параметрів граничного стаціонарного розподілу, який не змінюється з часом. Тобто, якщо продовжувати моделювання, нової інформації вже не отримати, а подальший експеримент стане марною витратою ресурсів.

Проблема визначення початкових умов пов'язана зі штучною природою роботи моделі. Імітаційні моделі зазвичай застосовують для вивчення систем у типових умовах. Саме для таких умов характерне досягнення стаціонарного стану (наприклад, для СМО не є типовим початковий стан s_0). У деяких стохастичних моделях потрібен певний час для того, щоб модель досягла стаціонарного режиму роботи. Необхідно визначити момент досягнення цього режиму та мінімізувати або виключити вплив початкового періоду моделювання на результати. Якщо досліджувати тільки встановлений режим, точність статистичних оцінок підвищується. На відміну від реальної системи, яка функціонує безперервно, модель працює лише епізодично – кожного разу, коли експериментатор запускає її для проведення чергового прогону. Тому щоразу після запуску моделі потрібен певний час, щоб система увійшла в стабільний (усталений) режим, подібний до режиму роботи реального об'єкта.

Внаслідок цього на початковому етапі роботи моделі її поведінка спотворюється впливом початкових умов запуску. Щоб зменшити вплив цих відхилень, застосовують такі основні підходи:

1. Виконання довгих прогонів моделі, щоб система самостійно вийшла в стаціонарний режим. Основний недолік такого підходу – велика витрата машинного часу.

2. Відсікання початкового періоду прогону. У цьому випадку аналізують, коли модель переходить у стаціонарний стан, і тільки після цього починають накопичувати статистику. Недоліком є необхідний додатковий

контроль за станом моделі і складність визначення моменту стабілізації в стохастичних процесах. Частина машинного часу витрачається на цей аналіз, а не на моделювання.

3. Вибір наближених до типових початкових умов, що скорочує тривалість перехідного періоду. Для цього зазвичай проводять пробний прогін, щоб визначити приблизний час виходу системи в стаціонарний режим.

Проте повністю усунути перехідний період під час машинного експерименту практично неможливо – ці заходи лише зменшують його тривалість, але не виключають повністю. Крім того, ці підходи слід застосовувати обережно, оскільки для деяких систем і процесів саме перехідний режим може бути предметом дослідження.

Розв’язання другої проблеми тактичного планування машинного експерименту пов’язане з оцінюванням точності та достовірності результатів моделювання (наприклад, при використанні методу статистичного моделювання засобами обчислювальної техніки) за певної кількості прогонів (об’єму вибірки). А також – з необхідністю визначення необхідної кількості реалізацій для досягнення заданого рівня точності та достовірності результатів моделювання системи.

Основні способи організації прогонів моделі включають повторні запуски моделювання та метод поділу на підінтервали (коли весь прогін розбивають на окремі частини і для кожної з них обчислюють середнє значення результатів). Щоб провести статистичний аналіз усієї послідовності змодельованого випадкового процесу, або кілька разів запускають модель із самого початку (метод повторних прогонів), або здійснюють моделювання протягом тривалішого часу (метод довгих прогонів).

Основні способи визначення тривалості імітаційного прогону такі:

- найчастіше задають конкретний момент завершення моделювання (наприклад, досягнення певного часу);

- використовують метод, що контролює розмір вибірки (передбачає автоматичне завершення моделювання за певними критеріями) після надходження до моделі певної кількості елементів або після обробки в системі певної кількості елементів.

З проблемою вибору кількості реалізацій для забезпечення необхідної точності та достовірності результатів машинного експерименту тісно пов’язана і третя проблема – проблема зменшення дисперсії оцінок характеристик процесу функціонування модельованих систем. Існують методи, які дозволяють:

- або при заданій кількості реалізацій підвищити точність оцінок, отриманих у процесі моделювання;
- або при заданій точності зменшити необхідну кількість реалізацій.

Ці методи ґрунтуються на апріорній інформації про структуру та поведінку модельованої системи і називаються методами зменшення дисперсії. Основна задача – спеціально побудувати алгоритм моделювання так, щоб забезпечити позитивну кореляцію між результатами. Наприклад, це може досягатися шляхом керування генерацією випадкових величин.

Оцінюючи ефективність методів зменшення дисперсії, необхідно враховувати додаткові витрати комп’ютерних ресурсів (часу та пам’яті) на реалізацію таких підходів. Теоретичне скорочення загального машинного часу для моделювання має бути перевірене з урахуванням складності самої реалізації алгоритмів.

Четвертою проблемою тактичного планування імітаційних експериментів є вибір правил автоматичної зупинки імітації. Найпростіший підхід – заздалегідь задати необхідну кількість реалізацій або тривалість моделювання. Проте цей детермінований підхід не є ефективним, оскільки ґрунтується на досить грубих припущеннях про розподіл вихідних параметрів, які на етапі тактичного планування ще невідомі.

Більш гнучкий спосіб – це задання довірчих інтервалів для вихідних параметрів і автоматична зупинка моделювання, коли ці інтервали досягнуті. Такий підхід теоретично дозволяє наблизити час моделювання до оптимального. Проте на практиці додавання до моделі процедур розрахунку довірчих інтервалів та правил зупинки призводить до збільшення часу на обробку однієї вибіркової точки при статистичному моделюванні.

4.4. Основні плани, що використовуються в імітаційному експерименті

Багато експериментальних планів сьогодні є стандартизованими. Вони описані в довідниках, реалізовані в математичних пакетах програм і в системах моделювання [16]. Однак дослідник має бути готовим до модифікації існуючих планів і їхнього пристосування до специфічних умов конкретних завдань.

За методом аналізу та типом математичної моделі розрізняють такі види планів: плани дисперсійного аналізу (однофакторні, багатофакторні), плани регресійного аналізу, плани коваріаційного аналізу.

Серед багатофакторних планів виділяють: двохрівневі та багаторівневі, а також симетричні та несиметричні.

У практиці машинного експерименту корисними є такі види планів:

1. Плани багатофакторного аналізу.

Плани типу 2^{k-p} передбачають, що всі k факторів мають два рівні, а для експерименту використовується лише частина всіх можливих комбінацій (дробна реплікація). Це дозволяє оцінити основні ефекти факторів та взаємодії низького порядку.

Виділяють такі типи планів:

- плани роздільної здатності III: жоден основний ефект не змішується з іншим основним ефектом, але основні ефекти можуть змішуватися з двофакторними взаємодіями, які також змішуються між собою;

- плани роздільної здатності IV: жоден основний ефект не змішується ні з іншим основним ефектом, ні з двофакторними взаємодіями, але самі двофакторні взаємодії можуть бути змішані між собою;
- плани роздільної здатності V: жоден основний ефект і жодна двофакторна взаємодія не змішуються з іншими основними ефектами або двофакторними взаємодіями, але ці взаємодії можуть змішуватися з трифакторними взаємодіями.

2. Плани відсіювального експерименту.

Якщо кількість факторів k велика, то навіть при використанні неповного факторного плану (2^{k-p}) загальна кількість комбінацій залишається значною. У такому випадку застосовують плани відсіювальних експериментів. Відсіювальний експеримент передбачає попередній відбір – виявлення найбільш важливих (суттєвих) факторів, які впливають на результат. Такі плани використовують на етапі попереднього вивчення, щоб скоротити кількість подальших досліджень, зосередившись тільки на тих факторах, які мають істотний вплив на систему.

Серед таких планів найчастіше використовуються:

- випадкові плани – комбінації рівнів факторів обираються випадковим чином (рандомізація) серед усіх можливих варіантів. Кількість комбінацій N може визначатися незалежно від кількості факторів та рівнів (N може бути, наприклад, меншим за k). Такий підхід дозволяє отримати достатньо точні оцінки індивідуальних ефектів факторів, при цьому не потребуються спеціальні методи аналізу – достатньо використання традиційного дисперсійного аналізу або регресійного аналізу;
- перенасичені плани – випадки, коли кількість комбінацій N менша, ніж кількість факторів k, і комбінації підбираються так, щоб для заданих N і k якість оцінок ефектів залишалася

прийнятною. Такі плани застосовують, коли кількість доступних експериментів обмежена, але потрібно оцінити велику кількість факторів, хоча і із певною втратою точності оцінок;

- плани групового відсіву (послідовного відсіву) – всі k факторів розбивають на q груп, де $q << k$. Далі ці q груп факторів досліджують за допомогою плану типу 2^{k-r} або за допомогою перенасиченого плану. Після першого етапу визначають, яка з груп факторів має найбільший вплив на результат, і цю групу далі ділять на кілька підгруп меншого розміру. Процес розбиття триває до тих пір, поки фактори не будуть проаналізовані окремо (в результаті отримують групи розміром в один фактор). Такий підхід дозволяє поетапно звузити коло досліджуваних факторів, скоротивши загальний обсяг експериментальних випробувань.

3. Плани для дослідження поверхні відгуку.

В імітаційному моделюванні при аналізі поверхні відгуку використовується технологія послідовного планування експерименту, яка спрямована на пошук оптимальної комбінації рівнів k кількісних факторів.

У майже стаціонарній області бажано апроксимувати поверхню відгуку щонайменше поліномом другого порядку. Для цього проводиться експеримент з квадратичним наближенням. Приклад квадратичного полінома (другого порядку) для випадку двох незалежних змінних:

$$\hat{y} = a_0 + a_1 x_1 + a_2 x_2 + a_{11} x_1^2 + a_{22} x_2^2 + a_{12} x_1 x_2.$$

Для оцінки коефіцієнтів регресії в такій моделі потрібно дослідити кожний фактор щонайменше на трьох рівнях, що викликає необхідність проведення повного трирівневого факторного експерименту 3^n або використання неповного факторного експерименту. При рості числа змінних більше трьох, кількість необхідних прогонів стає занадто великою, навіть при неповному експерименті. Крім того, експеримент з 3^n комбінаціями дає досить низьку точність оцінок коефіцієнтів регресії.

Для квадратичних поліномів застосовують спеціальні методи побудови експериментів. Серед них найбільш корисними є центральні композиційні, які створюються шляхом додавання додаткових точок до даних, отриманих у 2^n-факторному експерименті. Для таких планів стандартна похибка однакова для всіх точок, що знаходяться на однаковій відстані від центру області. Такі плани існують для будь-якої кількості факторів і являють собою регулярні або напіврегулярні геометричні фігури з центральними точками.

Відомі й інші типи планів, які застосовуються в плануванні експериментів: латинські квадрати, квадрати Юдена, збалансовані блоки та інші.

Латинський план (латинський квадрат) застосовують у випадках, коли експеримент проводиться з одним основним і двома другорядними факторами (наприклад, основний фактор – швидкість, другорядні – кут і крок). Якщо основний фактор має q рівнів, то для кожного другорядного фактора також беруть q рівнів. Вибір комбінацій цих рівнів здійснюється за допомогою спеціальної процедури. Нехай A – основний фактор, а B і C – другорядні (кількість факторів $n = 3$), кількість рівнів $q = 4$ для кожного фактору (A1, A2, A3, A4, B1, ..., B4, C1, ..., C4). Тоді план можна подати у вигляді матриці за рівнями фактора A. При цьому матрицю будують так, щоб у кожному рядку та кожному стовпці кожний рівень фактора A зустрічався лише один раз (табл. 4.1). Згідно наведеного плану досліди проводяться для таких комбінацій рівнів факторів: (A1, B1, C1), (A2, B1, C2), (A3, B1, C3), ..., (A3, B4, C4).

Таблиця 4.1 – Приклад латинського квадрата

Значення фактору B	Значення фактору C			
	C1	C2	C3	C4
B1	A1	A2	A3	A4
B2	A2	A3	A4	A1
B3	A3	A4	A1	A2
B4	A4	A1	A2	A3

Порівняно з ПФЕ використання латинського квадрату в даному випадку дозволяє скоротити кількість дослідів у чотири рази ($q^2 = 16$ проти $q^n = 4^3 = 64$).

Аналогічно латинським квадратам, може бути збудований латинський куб (греко-латинський квадрат) для проведення експериментів для чотирьох факторів ($n = 4$), кількість дослідних точок дорівнює при цьому також дорівнює q^2 замість q^n для ПФЕ.

Слід зауважити, що не для всіх випадків можна побудувати греко-латинський план – для кількості рівнів $q = 2$ та $q = 6$ його побудова неможлива.

Збалансований неповний блочний план – це такий неповний блочний план, в якому всі можливі комбінації рівнів зустрічаються однакову кількість разів. Якщо порівняння результатів дослідів між собою є однаково важливими, то комбінації рівнів у кожному блоці повинні підбиратися збалансовано, тобто так, щоб кожна пара комбінацій рівнів зустрічалася стільки ж разів, скільки й будь-яка інша.

Квадрат Юдена є латинським квадратом, в якому бракує хоча б одного стовпця, рядка або діагоналі (якщо випадковим чином вилучити більше одного стовпця з латинського квадрата, може бути порушена його збалансованість) – кількість стовпців не збігається з кількістю рядків та комбінацій рівнів. У таблиці 4.2 наведено приклад квадрата Юдена, отриманого з матриці латинського квадрата шляхом видалення другого рядка.

Таблиця 4.2 – Приклад квадрата Юдена

Значення фактору B	Значення фактору C			
	C1	C2	C3	C4
B1	A1	A2	A3	A4
B3	A3	A4	A1	A2
B4	A4	A1	A2	A3

У загальному випадку квадрат Юдена відповідає симетричному збалансованому неповному блочному плану, причому рядки відповідають блокам, а кожна комбінація рівнів факторів трапляється в кожному стовпці або на кожній «позиції» блока лише один раз.

Контрольні запитання

1. Що називається експериментом?
2. Якими бувають експерименти?
3. Які основні етапи проведення експерименту?
4. Що таке фактор?
5. Яким чином можна зменшити похибки, пов’язані з початковими умовами?
6. За якими ознаками класифікуються експерименти?
7. Що таке «відсіювальні» експерименти?
8. Чим відрізняються плани регресійного та кореляційного аналізу?
9. Що таке робастний план?
10. Яка основна ідея стратегічного планування експериментів?
11. Назвіть основні об’єкти стратегічного планування.
12. Які основні три проблеми вирішуються під час стратегічного планування експериментів?
13. Яка основна ідея тактичного планування експериментів?
14. Які основні чотири проблеми вирішуються під час стратегічного планування експериментів?
15. Які використовують види планів під час проведення машинного експерименту?
16. Які плани застосовуються при проведенні відсіювального експерименту?
17. Які плани застосовуються для дослідження поверхні відгуку?
18. Що таке «латинські» плани?

5. ОБРОБКА РЕЗУЛЬТАТІВ ІМІТАЦІЙНОГО ЕКСПЕРИМЕНТУ

5.1. Характеристики випадкових величин і процесів

До основних характеристик випадкових процесів належать математичне сподівання, дисперсія, коваріація, початкові і центральні моменти різних порядків тощо [2].

Сучасні системи імітаційного моделювання забезпечують автоматизовану обробку результатів моделювання, зокрема дозволяють виконати:

- обчислення статистичних характеристик випадкових параметрів, передусім математичних сподівань та дисперсій;
- фіксацію мінімальних і максимальних значень модельованих величин;
- побудову частотних розподілів результатів (гістограм);
- визначення коефіцієнтів використання елементів моделі тощо.

Але в багатьох випадках досліднику необхідно виконувати більш складну аналітичну обробку, що містить:

- встановлення функціональних або статистичних залежностей між досліджуваними параметрами;
- ідентифікацію значущих і незначущих факторів, що впливають на результати експерименту;
- порівняння випадкових величин з метою оцінювання статистичної значущості відмінностей або подібностей в їхніх характеристиках тощо.

У найбільш розвинених середовищах моделювання передбачені інструменти для реалізації таких видів обробки. Проте незалежно від доступних засобів, дослідник повинен розуміти принципи статистичної обробки, вміти правильно підготувати вхідні дані, коректно інтерпретувати

отримані результати, а також – обґрунтовано обирати методи аналізу і, за потреби, розробляти власні алгоритми обробки даних.

Оскільки результати експериментів можуть змінюватися щоразу, кількісна характеристика, яка при цьому аналізується, за своєю природою є змінною, що має випадковий характер. Таким чином, випадкова величина – це така величина, яка внаслідок експерименту з невизначеним результатом набуває певного числового значення з множини можливих, вона може приймати одне з можливих значень, причому наперед невідомо, яке саме значення буде отримано.

У математичних записах випадкові величини зазвичай позначають великими літерами латинського алфавіту (X, Y, Z тощо), а їх конкретні числові значення – відповідними малими літерами з індексами: $x_1, x_2, \dots, x_n;\ y_1, y_1, \dots, y_m;\ z_1, z_1, \dots, z_k$.

Змінна величина X, яка внаслідок випробування набуває одного зі значень $x_1, x_2, \dots, x_n$ скінченної або нескінченної числової послідовності, називається дискретною (розривною) випадковою величиною, якщо кожному значенню x_i випадкової величини X поставлено у відповідність певну ймовірність p_i події $(X = x_i)\ i = 1, 2, \dots, n$: $p_i = P(X = x_i)$.

Для дискретної випадкової величини характерно те, що між будь-якими двома сусідніми можливими значеннями не існує інших значень. Інакше кажучи, всі допустимі значення такої величини можна впорядкувати у вигляді послідовності, тобто перенумерувати. Кількість можливих значень дискретної випадкової величини може бути як скінченною, так і нескінченною; у другому випадку множину таких значень називають зліченною.

Значення випадкової величини можна розглядати як події, що з’являються при випробуваннях. Якщо ймовірності повної групи несумісних подій $x_1, x_2, \dots, x_n$: $p_1 = P(X = x_1),\ p_2 = P(X = x_2),\ \dots,\ p_n = P(X = x_n),$ то

сума ймовірностей усіх імовірних значень випадкової величини X дорівнює одиниці:

$$\sum_{i=1}^{n} P(X = x_i) = \sum_{i=1}^{n} p_i = 1.$$

Неперервна випадкова величина – це така випадкова величина, яка здатна приймати будь-яке значення з певного інтервалу числової прямої – як обмеженого, так і необмеженого. Кількість можливих значень такої величини є нескінченна.

Часто виникають задачі, в яких результати експериментів описуються не однією, а кількома випадковими величинами, що формують певну систему. Такі величини класифікують як одновимірні, двовимірні, або в загальному вигляді – n-вимірні випадкові величини. Наприклад, похибки Δx і Δy при визначенні положення точки на місцевості становлять двовимірну випадкову величину. У цьому випадку величини X та Y, які аналізуються одночасно, утворюють систему з двох випадкових величин. Аналогічно можна розглядати систему з n випадкових величин.

Множину випадкових величин $(X_1, X_2, \dots, X_n)$, які розглядаються одночасно, називають системою випадкових величин. Таку систему можна інтерпретувати як випадкову точку в p-вимірному просторі з координатами $(X_1, X_2, \dots, X_n)$, або як випадковий вектор, що спрямований від початку координат до точки $M = (X_1, X_2, \dots, X_n)$.

Багатовимірні випадкові величини, залежно від характеру їхніх можливих значень, також поділяються на дискретні та неперервні.

Для повного опису випадкової величини X недостатньо лише вказати множину її можливих значень – необхідно також задати ймовірності всіх подій, пов’язаних із цією величиною. Зокрема, слід визначити ймовірність того, що випадкова величина набуде певного значення або потрапить у заданий інтервал. Такий повний опис випадкової величини виконується за законом її розподілу.

Законом розподілу випадкової величини називають будь-яку залежність, що описує зв'язок між можливими значеннями випадкової величини та відповідними їм ймовірностями.

Існують різні способи задання закону розподілу: таблична, графічна та аналітична форма. Для дискретних випадкових величин найзручнішою є таблична форма – ряд розподілу, який подається у вигляді таблиці, де вказано всі можливі значення випадкової величини X та відповідні їм ймовірності.

Загальний вид таблиці розподілу наведено в табл. 5.1.

Таблиця 5.1 – Загальний вид таблиці розподілу одновимірної випадкової величини

X	x_1	x_2	…	x_n
p	p_1	p_2	…	p_n

При упорядкуванні таблиці звичайно формують варіаційний ряд, тобто розташовують дані в порядку зростання. Таким чином, $x_1 \le x_2 \le \cdots \le x_n$. Таблична форма запису може бути використана при порівняно невеликій кількості вимірів дискретної величини. Така форма називається ще рядом розподілу.

Законом розподілу дискретної двовимірної випадкової величини називають перелік можливих значень цієї величини (x_i, x_j) та їх ймовірностей $p(x_i, x_j)$, $i = 1, 2, \dots, n$; $j = 1, 2, \dots, m$. Закон розподілу двовимірної випадкової величини задається у вигляді таблиці з двома входами (табл. 5.2).

Таблиця 5.2 – Загальний вид таблиці розподілу двовимірної випадкової величини

Y	X			
	x_1	x_2	…	x_n
y_1	$p(x_1, y_1)$	$p(x_2, y_1)$	…	$p(x_n, y_1)$
y_2	$p(x_1, y_2)$	$p(x_2, y_2)$	…	$p(x_n, y_2)$
…	…	…	…	…
y_m	$p(x_1, y_m)$	$p(x_2, y_m)$	…	$p(x_n, y_m)$

Події $\left(X = x_i, Y = y_j\right)$, $i = 1, 2, \dots, n$; $j = 1, 2, \dots, m$ утворюють повну групу, тому і для них сума ймовірностей дорівнює 1:

$$\sum_{i=1}^{n}\sum_{j=1}^{m} p(x_i, x_j) = 1.$$

Графічне подання закону розподілу дискретної випадкової величини здійснюється за допомогою многокутника розподілу. Під многокутником розподілу розуміють ламану лінію, вершини якої мають координати, де по осі абсцис відкладаються можливі значення x_i випадкової величини X, а по осі ординат – відповідні ймовірності p_i (рис. 5.1).

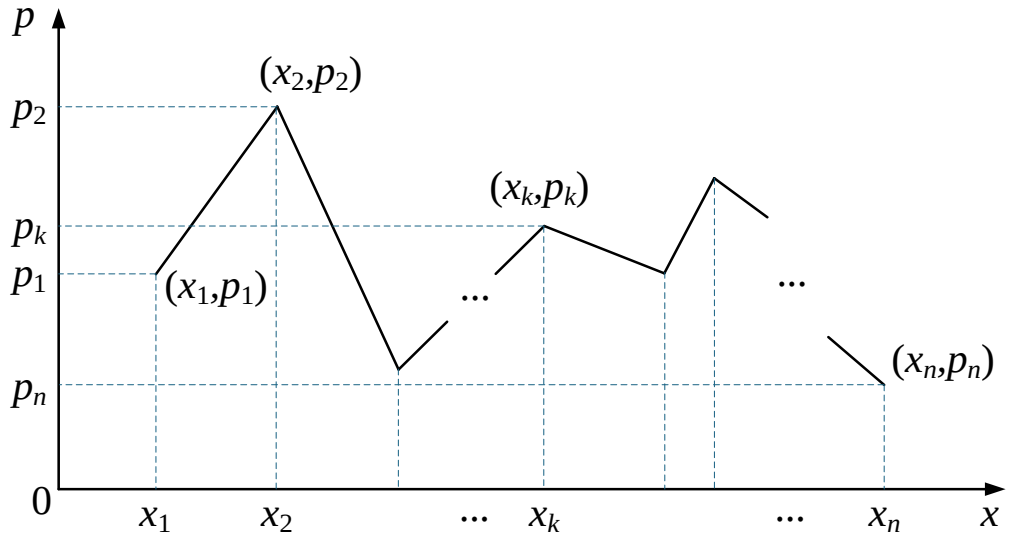


Рисунок 5.1 – Многокутник розподілу випадкової величини

Аналітична форма закону розподілу передбачає задання формули, за допомогою якої можна обчислити ймовірність p_i для будь-якого значення x_i випадкової величини X.

У практичній діяльності не завжди вдається точно визначити закон розподілу випадкової величини, або ж цей закон може виявитися надто складним для застосування в розрахунках. У зв’язку з цим виникла потреба використовувати числові характеристики, які дозволяють узагальнено описати основні властивості дискретних випадкових величин і які будуть достатніми для багатьох практичних цілей.

Серед числових характеристик випадкових величин виокремлюють характеристики положення, розсіювання, зв’язку та закону розподілу.

Характеристики положення описують, де на числовій осі зосереджені значення випадкової величини, тобто визначають її «середнє» або типове значення. До них належать мода, медіана, математичне сподівання, середнє геометричне та ін.

Характеристики розсіювання, своєю чергою, відображають ступінь відхилення можливих значень випадкової величини від центру розподілу. До таких характеристик належать дисперсія, середнє квадратичне відхилення, розмах.

Характеристика зв'язку між випадковими величинами (кореляція) визначає ступінь зв'язку між ними за допомогою коефіцієнта кореляції. У випадковому процесі зв'язок між значеннями випадкової функції в моменти часу визначає коефіцієнт автокореляції.

Характеристика закону розподілу ймовірностей випадкової величини задається у вигляді щільності або функції розподілу.

Мода неперервної випадкової величини Mo – це значення, при якому функція щільності розподілу набуває свого максимуму. Інакше кажучи, це найімовірніше значення випадкової величини. Залежно від кількості таких максимумів на графіку розподілу, розрізняють одномодальні, двомодальні та багатомодальні розподіли.

Медіана випадкової величини — це таке значення Me, відносно якого з рівною ймовірністю випадкова величина може набувати як більших, так і менших значень. Медіана розділяє розподіл імовірностей навпіл.

Математичне сподівання дискретної випадкової величини – це зважена сума всіх її можливих значень, де вагами виступають відповідні ймовірності. Інакше кажучи, це сума добутків кожного можливого значення випадкової величини на ймовірність його появи:

$$M(X) = \sum_{i=1}^{n} x_i \cdot p_i.$$

Математичне сподівання має низку важливих властивостей, серед яких основними є такі:

- математичне сподівання сталої величини дорівнює самій цій величині: $M(C) = C$;
- математичне сподівання суми двох випадкових величин дорівнює сумі їх математичних сподівань: $M(X + Y) = M(X) + M(Y)$. Наслідок: математичне сподівання суми сталої та випадкової величини дорівнює сумі цієї сталої і математичного сподівання випадкової величини: $M(C + X) = C + M(X)$;
- математичне сподівання добутку двох незалежних випадкових величин дорівнює добутку їх математичних сподівань: $M(X \cdot Y) = M(X) \cdot M(Y)$, якщо X і Y – незалежні. Наслідок: константу можна винести за знак математичного сподівання – добуток сталої величини та випадкової величини має математичне сподівання, яке дорівнює добутку сталої і математичного сподівання цієї випадкової величини: $M(C \cdot X) = C \cdot M(X)$.

Дисперсія характеризує середній ступінь розсіювання значень випадкової величини відносно її середнього (очікуваного) значення. Дисперсією випадкової величини називають математичне сподівання квадрата різниці між випадковою величиною та її математичним сподіванням:

$$D(X) = M((X - M(X))^2),$$

$$D(X) = \sum_{i=1}^{n} (x_i - M(X))^2 \cdot p_i.$$

Основні властивості дисперсії:

- дисперсія сталої величини дорівнює нулю: $D(C) = 0$;
- сталий множник можна винести за знак дисперсії в квадраті: $D(C \cdot X) = C^2 \cdot D(X)$;

- дисперсія суми двох незалежних випадкових величин X і Y дорівнює сумі їх дисперсій: $D(X+Y)=D(X)+D(Y)$. Наслідок: дисперсія суми сталої та випадкової величини дорівнює дисперсії самої випадкової величини: $D(X+C)=D(X)$;
- дисперсія різниці двох незалежних випадкових величин X і Y також дорівнює сумі їхніх дисперсій: $D(X-Y)=D(X)+D(Y)$;
- альтернативне визначення дисперсії: дисперсія випадкової величини дорівнює різниці між математичним сподіванням квадрата цієї величини та квадратом її математичного сподівання: $D(X)=M(X^2)-[M(X)]^2$ або

$$D(X)=\sum_{i=1}^{n} {x_i}^2 \cdot p_i - M(X)^2.$$

Дисперсія випадкової величини має розмірність, що відповідає квадрату розмірності самої величини. У практичних застосуваннях це може бути незручним, тому для оцінювання розсіювання значень випадкової величини частіше використовують середнє квадратичне відхилення $\sigma(X)=\sigma_x$, яке визначається як квадратний корінь з дисперсії: $\sigma_x=\sqrt{D(X)}$ і має ту саму розмірність, що й випадкова величина, і тому є більш інтерпретованою з прикладної точки зору.

Розмах – найпростіший показник варіації, який являє собою різницю між максимальним і мінімальним значеннями випадкової величини.

$$R=x_{max}-x_{min}.$$

Початковим моментом α_k k-го порядку випадкової величини X називають математичне сподівання її k-го ступеня. Ця характеристика визначається за формулою:

$$\alpha_k=M(X^k),$$

для дискретної випадкової величини:

$$\alpha_k=\sum_{i=1}^{n} x_i^k \cdot p_i.$$

Відповідно, математичне сподівання і дисперсія:

$$M(X) = \alpha_1, \qquad D(X) = M(X^2) - [M(X)]^2 = \alpha_2 - (\alpha_1)^2.$$

Центральним моментом k-го порядку називають математичне сподівання k-го ступеня відхилення випадкової величини від її математичного сподівання:

$$\mu_k = M((X - M(X))^k),$$

для дискретної випадкової величини:

$$\mu_k = \sum_{i=1}^{n} (x_i - M(X))^k \cdot p_i.$$

Центральні моменти першого та другого порядку:

$$\mu_1 = 0, \qquad \mu_2 = D(X).$$

Центральні моменти широко застосовуються для аналізу симетрії, асиметрії та «гостровершинності» розподілу.

Коефіцієнтом асиметрії називають відношення центрального моменту третього порядку до куба середнього квадратичного відхилення. Ця величина обчислюється за формулою:

$$C_S = \frac{\mu_3}{\sigma^3} = \frac{\mu_3}{(\mu_2)^{3/2}}.$$

Ще однією важливою числовою характеристикою випадкової величини є коефіцієнт ексцесу. Він використовується для оцінки «гостровершинності» або «плосковершинності» розподілу відносно нормального і визначається за формулою:

$$E_x = \frac{\mu_4}{\sigma^4} - 3 = \frac{\mu_4}{\mu_2^2} - 3.$$

Коефіцієнт кореляції – число, яке дає уявлення про напрям і силу зв’язку між випадковими величинами. Межі його коливань – від 0 до ±1.

Форма зв’язку може бути:

- пряма (лінійна) – рівномірній зміні однієї випадкової величини відповідає рівномірна зміна іншої випадкової величини (за незначних коливань);

- нелінійна (криволінійна) – рівномірній зміні однієї випадкової величини відповідає нерівномірна зміна іншої випадкової величини (причому нерівномірність має певну закономірність).

Напрямок зв’язку визначається за знаком коефіцієнта кореляції:

- прямий зв’язок (додатний) – динаміка є однонаправленою: збільшення однієї випадкової величини зумовлює збільшення іншої;
- зворотний зв’язок (від’ємний) – динаміка є різноспрямованою: збільшення однієї випадкової величини зумовлює зменшення іншої.

Сила зв’язку:

- значення коефіцієнту кореляції $|r| = 0$ – зв’язок відсутній;
- значення коефіцієнту кореляції $|r| = 0.01 – 0.29$ – зв’язок слабкий;
- значення коефіцієнту кореляції $|r| = 0.30 – 0.69$ – зв’язок середній;
- значення коефіцієнту кореляції $|r| = 0.70 – 0.99$ – зв’язок сильний;
- значення коефіцієнту кореляції $|r| = 1$ – зв’язок повний.

Види коефіцієнтів кореляції:

- коефіцієнт парної кореляції – дає уявлення про зв’язок між двома випадковими величинами без урахування впливу сторонніх чинників. Вирізняють ранговий (Спірмена) та лінійний (Пірсона) коефіцієнти. Формула розрахунку для останнього:

$$r_{x,y} = \frac{cov(x, y)}{\sigma_x \sigma_y};$$

- множинний коефіцієнт кореляції – відображає зв’язок між двома і більше випадковими величинами. Його розрахунок дозволяє

зрозуміти взаємодію багатьох чинників і їхній вплив на основну досліджувану випадкову величину;

- парціальний (частковий) коефіцієнт кореляції – оцінюється на основі парного та множинного коефіцієнтів і дозволяє визначити взаємозв'язок між двома випадковими величинами, виключаючи вплив інших.

У статистичному аналізі оцінка параметра генеральної сукупності завжди розглядається як випадкова величина, оскільки вона залежить від вибірки, яка є випадковою. У зв'язку з цим оцінювання параметрів потребує не лише знаходження конкретного числового значення, а й аналізу властивостей функції, що зіставляє кожній вибірці її оцінку.

Щоб оцінка вважалась статистично надійною та коректною, вона повинна задовольняти низці важливих вимог, серед яких основними є:

- незміщеність – оцінка $\overline{\Theta}$ називається незміщеною, якщо її математичне сподівання дорівнює істинному (невідомому) значенню параметра Θ, тобто $M[\overline{\Theta}] = \Theta$. Це означає, що в середньому, за багатьох реалізацій вибірки, оцінка не буде систематично завищеною або заниженою;
- обґрунтованість (консистентність) – оцінка $\overline{\Theta}$ є обґрунтованою, якщо зі збільшенням обсягу n вибірки вона з імовірністю, що прямує до одиниці, наближається до істинного значення параметра. Формально, для оцінки $\overline{\Theta}$ параметра Θ виконується умова збіжності за ймовірністю $\overline{\Theta} \xrightarrow{P} \Theta$ при $n \to \infty$. Це гарантує, що при достатньо великій вибірці похибка оцінювання буде доволі малою з високою ймовірністю;
- ефективність – серед усіх k можливих незміщених та обґрунтованих оцінок $\overline{\Theta}_k$ перевагу надають тій, яка має найменшу дисперсію $D[\overline{\Theta}] = min\overline{\Theta}_k$. Це означає, що така оцінка характеризується найменшою мінливістю навколо істинного

значення параметра, тобто забезпечує найточніше наближення при заданих умовах.

У загальному випадку задача статистичного оцінювання полягає в знаходженні такої оцінки параметра, яка поєднує в собі вказані властивості, забезпечуючи високу точність, стабільність і надійність результатів у процесі інтерпретації даних.

У табл. 5.3 наведені основні формули оцінок характеристик досліджуваної величини, що відповідають нормальному закону розподілу ймовірностей.

Таблиця 5.3 – Оцінки характеристик випадкової величини

Характеристика Θ	Оцінка $\overline{\Theta}$	Середньоквадратичне відхилення оцінки
Математичне сподівання $M[x] = \int_{-\infty}^{\infty} xf(x)dx$	$\overline{x} = \frac{\sum_{i=1}^{N} x_i}{N}$	$\sigma_{\overline{x}} = \frac{S}{\sqrt{N}}$
Дисперсія $D[x] = M[x^2] - (M[x])^2$	$S^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})^2$	$\sigma_{S^2} = \sqrt{\frac{2}{N}}S^2$
Середньоквадратичне відхилення $\sigma_x = \sqrt{D[x]}$	$S = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})^2}$	$\sigma_S = \frac{S}{\sqrt{2N}}$
Ймовірність події P	$\overline{P} = \frac{m}{N}$	$\sigma_{\overline{P}} = \sqrt{\frac{P(1-P)}{N}}$
Коефіцієнт кореляції $Q_{x,y} = \frac{cov(x,y)}{\sigma_x \sigma_y}$	$\overline{r}_{xy} = \frac{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})\,(y_i - \bar{y})}{S_x S_y}$	$\sigma_{\overline{r}} = \frac{1-\overline{r}_{xy}^2}{\sqrt{N}}$

Іноді формули для обчислення оцінок математичного сподівання та дисперсії випадкової величини використовують у рекурентній формі:

$$\overline{x}_n = \overline{x}_{n-1}\frac{n-1}{n} + \frac{x_n}{n},$$

$$S_n^2 = S_{n-1}^2\frac{n-1}{n} + \frac{(x_n - \overline{x}_n)^2}{n},$$

де $\overline{x}_n$, $\overline{x}_{n-1}$, S_n^2, S_{n-1}^2 – відповідно оцінки математичного сподівання та дисперсії, обчислені за результатами n і $(n-1)$ реалізацій імітаційної моделі; x_n – значення випадкової величини в реалізації n.

Ці рекурентні формули зручні для покрокового оновлення статистик без потреби зберігати всі попередні дані.

До характеристик випадкового процесу відносять математичне сподівання, дисперсію та коваріаційну функцію.

Математичним сподіванням випадкового процесу $X(t)$ називають траєкторію математичних сподівань випадкових величин, що входять до складу цього процесу:

$$m(t) = M[X(t)] = \int_{-\infty}^{\infty} xf(t,x)dx\,, t \in T.$$

Дисперсією випадкового процесу називають траєкторію дисперсій випадкових величин, що входять до складу цього процесу:

$$D(t) = D[X(t)] = M[X^2(t)] - m^2(t), t \in T.$$

Таким чином, математичне сподівання $m(t)$ випадкового процесу $X(t)$ є певною невипадковою «середньою функцією», навколо якої коливаються реалізації випадкового процесу.

Дисперсія $D(t)$ випадкового процесу $X(t)$ є невипадковою невід'ємною функцією, яка характеризує ступінь розсіяння реалізацій випадкового процесу $X(t)$ відносно його математичного сподівання $m(t)$.

Якщо перетин випадкового процесу являє собою дискретну випадкову величину з реалізацією розподілу $X(t)$, то дисперсія $D(t)$ випадкового процесу обчислюється за формулою:

$$D(t) = D[X(t)] = \sum (x_i - m(t))^2 p_i(t)\,, t \in T,$$

де i – номер можливого значення випадкової величини $X(t)$ при даному t, $p_i(t)$ – ймовірність цього значення.

Для безперервних випадкових процесів зі щільністю $f(t,x)$ дисперсія обчислюється за формулою:

$$D(t) = \int_{-\infty}^{\infty} [x - m(t)]^2 f(t,x) dx\,, t \in T.$$

Середнє квадратичне відхилення для випадкових процесів:

$$\sigma(t) = \sigma[X(t)] = \sqrt{D(t)}.$$

Математичне сподівання $m(t)$ та дисперсія $D(t)$ є важливими характеристиками випадкового процесу $X(t)$, однак вони не є вичерпними, оскільки визначаються лише одновимірним законом розподілу.

На рис. 5.2 наведено реалізації двох випадкових процесів з приблизно однаковими математичними сподіваннями $m_1(t)$ і $m_2(t)$ та дисперсіями.

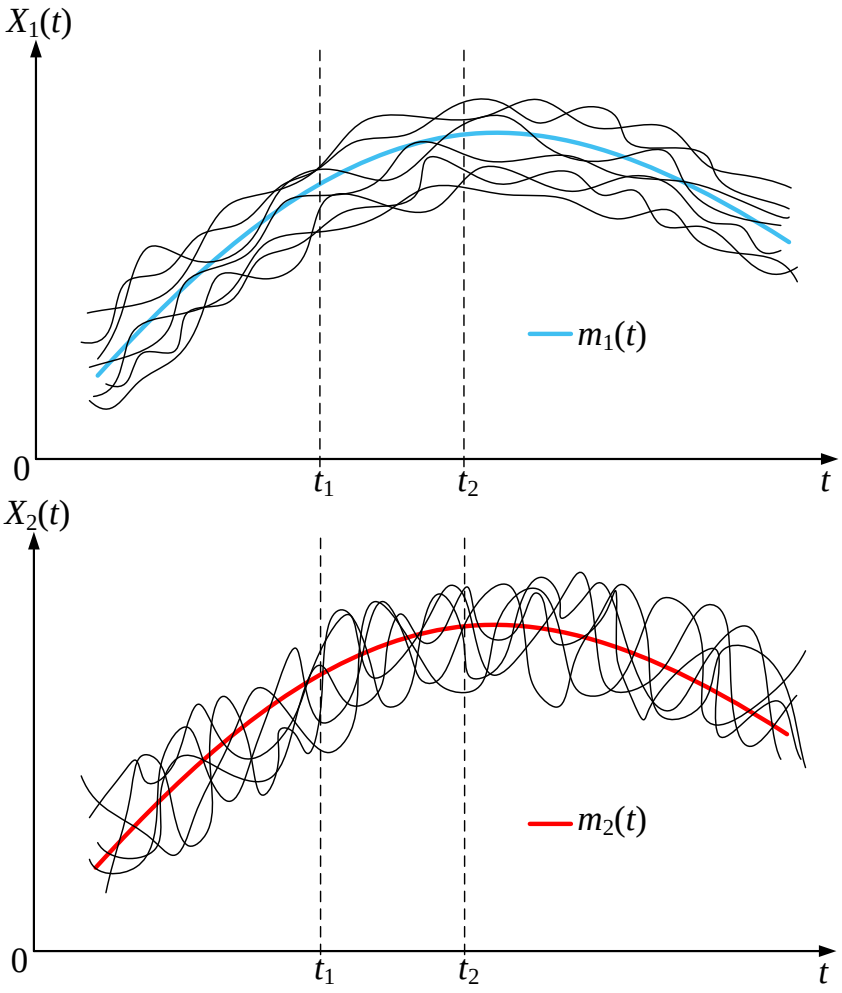


Рисунок 5.2 – Реалізації випадкових процесів $X_1(t)$ і $X_2(t)$

Внутрішня структура процесів на рис. 5.2 істотно відрізняється. Випадковий процес $X_1(t)$ характеризується плавними змінами реалізацій,

тоді як $X_2(t)$ має чітко виражену коливальну природу. Процес $X_1(t)$ демонструє підвищену передбачуваність: наприклад, між перетинами $X_1(t_1)$ і $X_1(t_2)$ існує суттєва ймовірнісна залежність. Це твердження не є справедливим для процесу $X_2(t)$: між його перетинами $X_2(t_1)$ і $X_2(t_2)$ майже відсутній стохастичний зв'язок, за умови, що ці моменти часу достатньо віддалені один від одного.

Для характеристики ступеня лінійної залежності між перетинами випадкового процесу використовується ще одна важлива величина – коваріаційна функція.

Коваріаційною функцією випадкового процесу називають функцію двох змінних, значення якої дорівнюють коефіцієнтам коваріації між перетинами процесу у відповідні моменти часу:

$$cov_x[X(t_1), X(t_2)] = M\{[X(t_1) - m(t_1)] \cdot [X(t_2) - m(t_2)]\} =$$
$$= M[X(t_1) \cdot X(t_2)] - m(t_1) \cdot m(t_2),$$

де t_1, t_2 – моменти часу, $m(t_1), m(t_2)$ – математичні сподівання процесу в ці моменти.

Коваріаційна функція дозволяє кількісно оцінити залежність між значеннями випадкового процесу в різні моменти часу, що є особливо важливим для аналізу динамічних і стохастичних систем.

Очевидно, що коваріаційні функції випадкових процесів $X_1(t)$ і $X_2(t)$ на рис. 5.2 істотно відрізняються. Це свідчить про те, що процес $X_1(t)$ має сильнішу статистичну залежність між своїми значеннями в різні моменти часу, тобто його реалізації є більш гладкими і передбачуваними. У той час як процес $X_2(t)$, навпаки, характеризується слабкою або майже відсутньою залежністю між перетинами при збільшенні інтервалу часу, що проявляється в його різко змінних, коливальних траєкторіях.

Подібно до того, як будь-яка випадкова величина повністю характеризується своїм законом розподілу, випадковий процес може бути описаний за допомогою так званого кінцевовимірного розподілу.

Кінцевовимірним розподілом випадкового процесу в моменти часу $t_1, t_2, \dots, t_n$ називається розподіл векторної випадкової величини $\big(X(t_1), X(t_2), \dots, X(t_n)\big)$, тобто розподіл сукупності значень процесу у фіксовані моменти часу:

$$F_{t_1,t_2,\dots,t_n}(x_1, x_2, \dots, x_n) = P\{X(t_1) < x_1, X(t_2) < x_2, \dots, X(t_n) < x_n\}.$$

Іншими словами, кінцевовимірний розподіл – це спільний закон розподілу кінцевої множини перетинів випадкового процесу. Через ці розподіли можна досліджувати поведінку процесу, його залежності між різними моментами часу, а також визначати клас процесу (наприклад, гаусівський, пуассонівський тощо).

При дослідженні випадкового процесу $X(t)$ часовий інтервал $(0, T)$ – це послідовність з M точок $t_j, j = \overline{1, M}$, у кожній з яких вимірюється значення перетину $x_i\big(t_j\big)$, де i – номер реалізації випадкового процесу, $i = \overline{1, N}$. Сукупність перетинів у кожній часовій точці – випадкові числа деякої випадкової величини зі своїми законами розподілу, математичними сподіваннями та дисперсіями. Оцінки математичного сподівання та дисперсії розраховуються за формулами:

$$\overline{x}\big(t_j\big) = \frac{\sum_{i=1}^{N} x_i\big(t_j\big)}{N}, \qquad S^2_{x(t_j)} = \frac{1}{N}\sum_{i=1}^{N}\Big(x_i\big(t_j\big) - \overline{x\big(t_j\big)}\Big)^2.$$

Ступінь залежності між перетинами визначає автокореляційна функція, її оцінка:

$$\overline{K}_{x(t_k,t_s)} = \frac{1}{N}\sum_{i=1}^{N}\big(x_i(t_k) - \overline{x}(t_k)\big)\cdot\big(x_i(t_s) - \overline{x}(t_s)\big),$$

де $x_i(t_k)$ і $x_i(t_s)$ – значення перетинів у точках t_k і t_s i-ї реалізації, $\overline{x}(t_k)$ і $\overline{x}(t_s)$ – оцінки математичних сподівань сукупності перетинів у точках t_k і t_s.

Для стаціонарних процесів математичне сподівання, кореляційна і коваріаційна функції розраховуються за такими формулами:

$$m_x(t) = \int_{-\infty}^{+\infty} xf(x)dx = const,$$

$$\rho_x(t_1, t_2) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} x_1 x_2 f_\tau(x_1, x_2)dx_1\,dx_2 = \rho_x(\tau),$$

$$cov_x(t_1, t_2) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} (x_1 - m_x)(x_2 - m_x)f_\tau(x_1, x_2)dx_1\,dx_2 = \rho_x(\tau) - m_x^2,$$

де $\tau = t_1 - t_2$.

Дисперсія стаціонарного процесу є постійною і дорівнює значенню кореляційної функції при нульовому значенню аргументу:

$$D_x = \rho_x(0) = \int_{-\infty}^{+\infty} (x - m_x)^2 f(x)dx.$$

Враховуючи те, що часто обмежуються застосуванням кореляційної теорії випадкових процесів, поняття стаціонарності розширили.

Випадковий процес називається стаціонарним у широкому значенні, якщо його математичне сподівання $m_x(t)$ не залежить від часу, а кореляційна функція залежить лише від різниці аргументів $\tau = t_1 - t_2$:

$$m_x(t) = const, \qquad \rho_x(t_1, t_2) = \rho_x(\tau).$$

Таким чином, математичне сподівання та кореляційна функція інваріантні відносно зміни аргументу.

Для ергодичних процесів при розрахунку оцінки математичного сподівання виконується властивість ергодичності, яка формально виглядає так:

$$\frac{1}{N}\sum_{i=1}^{N} x_i(t_j) = \frac{1}{M}\sum_{j=1}^{M} x_i(t_j).$$

5.2. Елементи дисперсійного аналізу

Дисперсійний аналіз – це сукупність статистичних методів, що призначені для обробки експериментальних даних, на які одночасно впливає

велика кількість чинників [3]. Основною метою дисперсійного аналізу є оцінка впливу окремих факторів на результативну ознаку та виявлення серед них найбільш значущих.

Розрізняють однофакторний, двофакторний та багатофакторний дисперсійний аналіз.

Однофакторний дисперсійний аналіз застосовується в тих випадках, коли наявні три або більше незалежних вибірок, отриманих із однієї генеральної сукупності шляхом варіювання певного чинника, для якого неможливо або недоцільно здійснити прямі кількісні вимірювання. При цьому вважається, що вибірки мають різні середні значення, але однакові дисперсії. При однофакторному аналізі n одиниць сукупності розподілені на m груп по значенням одного фактора.

Основна задача полягає в з'ясуванні, чи справді розбіжності між середніми значеннями зумовлені впливом досліджуваного фактора, чи ці відмінності є лише наслідком випадкових флуктуацій, викликаних обмеженим обсягом вибірок. Іншими словами, якщо всі вибірки репрезентують одну й ту ж генеральну сукупність, то варіація між групами не повинна перевищувати варіацію всередині самих груп.

Дисперсійний аналіз ґрунтується на припущенні, що досліджувана випадкова величина розподілена за нормальним законом. Його концепція базується на вивченні відмінностей між середніми значеннями груп, кожна з яких відповідає певному рівню або категорії досліджуваного чинника.

Хоча предметом аналізу є саме середні значення, метод називається дисперсійним, оскільки статистичне обґрунтування відмінностей між середніми досягається через порівняння оцінок дисперсій. Сутність підходу полягає в поділі загальної дисперсії на дві складові (правило додавання дисперсій): внутрішньогрупову дисперсію та міжгрупову (факторну) дисперсію, що оцінюється без урахування групової належності.

Якщо досліджуваний фактор чинить значущий вплив, тоді варіативність (дисперсія) спостережень усередині окремих груп суттєво

менша порівняно з загальною дисперсією по всій вибірці. Це пояснюється тим, що середні значення в різних групах істотно відрізняються між собою, що й зумовлює зростання загальної (міжгрупової) варіації.

Згідно правила додавання дисперсій для оцінювання впливу окремих факторів на варіативність індивідуальних значень результативної ознаки y під час дисперсійного аналізу окрім загального середнього значення ознаки (тобто середнього для всієї статистичної сукупності $\overline{x}$), обчислюються також середні значення ознаки в межах окремих груп сукупності (групові або часткові середні $\overline{x}_j$), а також три основні показники дисперсії: загальна дисперсія, міжгрупова дисперсія, середня внутрішньогрупова дисперсія.

Загальна дисперсія $D_{\text{заг}}$ характеризує загальну варіацію результативної ознаки під впливом усіх факторів, які визначають рівень ознаки в спостережуваних одиниць досліджуваної сукупності. Її величина визначається за такою формулою:

$$D_{\text{заг}} = \frac{\sum_{j=1}^{m}\left(x_j - \overline{x}\right)^2 \cdot f_j}{\sum_{j=1}^{m} f_j},$$

де x_j – значення ознаки, $\overline{x}$ – загальне середнє для всієї досліджуваної сукупності, m – кількість груп, f_j – число одиниць сукупності в окремій групі.

Міжгрупова дисперсія (дисперсія групових середніх, $D_{\text{м}}$) відображає систематичну варіацію, тобто відмінності в значеннях досліджуваної ознаки, що виникають під впливом фактора, покладеного в основу групування:

$$D_{\text{м}} = \frac{\sum_{j=1}^{m}\left(\overline{x}_j - \overline{x}\right)^2 \cdot f_j}{\sum_{j=1}^{m} f_j},$$

де $\overline{x}_j$ – середнє значення ознаки в окремій групі.

Внутрішньогрупова дисперсія знаходиться окремо для кожної групи, тому для одержання її значення по сукупності в цілому підраховують середню величину $\overline{D}_{\text{вг}}$, що показує варіацію за рахунок інших факторів (всіх

факторів, за винятком ознаки-фактора, покладеного в основу угруповання) і не залежить від умови, покладеної в основу угруповання:

$$\overline{D}_{\text{вг}} = \frac{\sum_{j=1}^{m} \sigma_j^2 \cdot f_j}{\sum_{j=1}^{m} f_j},$$

де $\sigma_j^2 = \frac{\sum_{i=1}^{f_j}(x_i - \overline{x}_i)^2}{f_j}$ – внутрішньогрупова дисперсія в окремій групі j.

Згідно з правилом додавання дисперсій загальна дисперсія являє собою суму групової дисперсії і середньої з внутрішньогрупових дисперсій:

$$D_{\text{заг}} = D_{\text{м}} + \overline{D}_{\text{вг}}.$$

Основна ідея однофакторного дисперсійного аналізу полягає в порівнянні дисперсії досліджуваної ознаки, зумовленої впливом фактора, з дисперсією випадкових похибок вимірювання цієї ознаки. Якщо виявлене розходження між цими дисперсіями є статистично значущим, то це свідчить про суттєвий вплив фактора на поведінку або рівень досліджуваного результативного показника.

Загальна варіація, яка позначається як SST (Total Sum of Squares), поділяється на дві складові:

- міжгрупову суму квадратів відхилень (SSB, Sum of Squares Between Groups), що відображає відмінності між групами,
- внутрішньогрупову суму квадратів відхилень (SSW, Sum of Squares Within Groups), яка характеризує варіацію ознаки всередині груп.

Кожна сума квадратів пов’язана з певним числом ступенів свободи – кількістю незалежних способів варіювання змінних, тобто незалежних відхилень.

Число ступенів свободи залежить від загальної кількості спостережень n та кількості визначених за ними параметрів (констант).

Загальна кількість ступенів свободи df_{total} також розкладається на міжгрупову кількість ступенів вільності df_{bg} (between groups) і

внутрішньогрупову df_{wg} (within groups), тобто $df_{total} = df_{bg} + df_{wg}$, де $df_{total} = n - 1, df_{bg} = m - 1, df_{wg} = n - m.$

Усі ці показники структуруються в таблиці дисперсійного аналізу, яка дозволяє оцінити значущість впливу фактора на результуючу ознаку (табл. 5.4).

Таблиця 5.4 – Однофакторний дисперсійний аналіз

Джерело варіації	Число ступенів свободи	Сума квадратів відхилень	Середня сума квадратів відхилень	Критерій Фішера F	Критичне значення $F_{\alpha(df_{bg},df_{wg})}$
Міжгрупова	df_{bg}	SSB	SSB/df_{bg}	$\frac{SSB/df_{bg}}{SSW/df_{wg}}$	$F_{\alpha(df_{bg},df_{wg})}$
Внутрішньогрупова	df_{wg}	SSW	SSW/df_{wg}		
Загальна	df_{total}	$SSB + SSW$			

В основі дисперсійного аналізу лежить поділ дисперсії на компоненти згідно правила додавання дисперсій:

$$SST = SSB + SSW.$$

Загальна сума квадратів відхилень SST розраховується за формулою:

$$SST = \sum_{j=1}^{n} (x_j - \overline{x})^2.$$

Міжгрупова сума квадратів відхилень SSB розраховується за формулою:

$$SSB = \sum_{j=1}^{m} (\overline{x}_j - \overline{x})^2 \cdot f_j.$$

Внутрішньогрупова сума квадратів відхилень SSW (within groups) розраховується так:

$$SSW = \sum_{j=1}^{m} \sum_{i=1}^{n/m} (x_{i,j} - \overline{x}_j)^2.$$

Далі розраховують міжгруповий середній квадрат відхилень (на ступінь свободи) S_1^2, а також внутрішньогруповий середній квадрат відхилень (на ступінь свободи) S_2^2:

$$S_1^2 = \frac{SSB}{m-1},$$

$$S_2^2 = \frac{SSW}{n-m}.$$

Якщо відмінності внутрішньогрупових середніх мають випадковий характер, тобто не пов'язані зі зміною факторної ознаки, то $S_1^2 = S_2^2$.

Це положення відповідає нульовій гіпотезі $H_0: S_1^2 = S_2^2$.

Статистичною гіпотезою називають припущення щодо властивостей генеральної сукупності, яке можна перевірити на основі вибіркових даних. Позначається така гіпотеза літерою H (від лат. hypothesis). Гіпотеза про те, що дві сукупності, які порівнюються за певними ознаками, не відрізняються, називається нульовою гіпотезою (або нуль-гіпотезою) і позначається H_0. У цьому випадку припускається, що справжня різниця між порівнюваними характеристиками дорівнює нулю, а виявлені за даними вибірки відхилення є випадковими. Нульова гіпотеза відхиляється тоді, коли за результатами вибірки отримується значення, яке є малоймовірним у випадку істинності цієї гіпотези.

Межею малоймовірного результату вважаються значення рівня значущості α, що зазвичай приймають рівними 0.05 (5 %), 0.01 (1 %) або 0.001 (0.1 %).

Статистичний критерій – це певне правило, яке визначає умови, за яких перевірювана гіпотеза повинна бути або відхилена, або прийнята. Критерій перевірки гіпотези дає змогу встановити, суперечать чи не суперечать фактичні дані сформульованій гіпотезі.

Процедура перевірки статистичної гіпотези включає такі етапи:

- формулювання дослідницької задачі у вигляді статистичної гіпотези;
- вибір статистичної характеристики (критерію) для перевірки гіпотези;

- визначення досліджуваної (нульової) та альтернативної гіпотез на основі аналізу можливих помилок і наслідків;
- встановлення допустимої області значень, критичної області та критичного значення статистичного критерію за відповідними таблицями.
- обчислення фактичного значення обраного статистичного критерію;
- порівняння фактичного значення з критичним і прийняття рішення щодо нульової гіпотези: вона або відхиляється, або не відхиляється.

Під час перевірки гіпотез можливі два типи помилок:

- помилка I роду – відхилення нульової гіпотези, коли вона насправді є істинною. Ймовірність такої помилки позначається літерою α;
- помилка II роду – прийняття альтернативної гіпотези, коли вона фактично є хибною. Ймовірність такої помилки позначається літерою β.

Типи помилок під час перевірки гіпотез наведено в таблиці 5.5.

Таблиця 5.5 – Типи помилок під час перевірки гіпотез

Рішення	Вірна гіпотеза	
	H_0	H_1
Прийняти	Немає помилки	Помилка II роду
Відхилити	Помилка I роду	Немає помилки

Критичною областю називають таку область значень статистичного критерію, потрапляння до якої призводить до відхилення нульової гіпотези. Ймовірність потрапляння значення критерію в цю область дорівнює прийнятому рівню значущості α. Область допустимих значень є доповненням критичної області. Якщо обчислене значення критерію

потрапляє до області допустимих значень, це свідчить про те, що висунута гіпотеза не суперечить фактичним даним (тобто H_0 не відхиляється).

Критичні точки або межі критичної області – це ті значення, які відокремлюють критичну область від допустимої. Отже:

- якщо обчислене значення критерію потрапляє в критичну область, нульова гіпотеза H_0 відхиляється, оскільки вона суперечить фактичним даним;
- якщо ж воно потрапляє в область допустимих значень, гіпотезу не відхиляють.

У даному випадку потрібно перевірити гіпотезу H_0 – припущення про випадковість відмінностей між груповими середніми значеннями, або про відсутність залежності цих відмінностей від досліджуваного фактора.

Правилом перевірки нульової гіпотези H_0 у дисперсійному аналізі виступає F-критерій Фішера:

$$F = \frac{S_1^2}{S_2^2},$$

де $S_1^2 \geq S_2^2$.

При заданому рівні значущості α та за відомими ступенями свободи міжгрупової варіації df_{bg} і внутрішньогрупової варіації df_{wg} визначається критичне значення F-критерію за таблицею значень F-критерію Фішера.

Необхідні припущення для коректного застосування F-критерію:

- нормальний розподіл значень залежної (результативної) змінної в кожній групі;
- однорідність (рівність) дисперсій у всіх групах.

У разі наявності декількох факторів аналогічні обчислення та перевірка за критерієм Фішера проводяться для кожного з них окремо.

Приклад. Відділ контролю якості IT-компанії, що має мережу сервісних центрів та точок продажу комп'ютерного обладнання та аксесуарів до нього, провів дослідження тривалості автономної роботи ноутбуків різних

виробників від повністю зарядженого акумулятора без підключення до живлення за умов максимального обчислювального навантаження.

Для тестування було відібрано по п’ять пристроїв кожного з чотирьох ($m = 4$) виробників ($f_1 = f_2 = f_3 = f_4 = 5$). Випробування проводилися на тестовому стенді – спеціалізованій платформі, яка забезпечує максимальне навантаження на систему (процесор, графіку, накопичувач тощо) та автоматично фіксує час роботи до повного розрядження батареї нижче критичного рівня.

Результати тестування автономної роботи комп’ютерних пристроїв за умов пікового навантаження наведено в таблиці 5.6.

Таблиця 5.6 – Час роботи ноутбуків, годин

Номер пристрою	Виробник 1	Виробник 2	Виробник 3	Виробник 4
1	21.2	25.4	22.4	26.0
2	19.5	24.3	24.0	21.7
3	23.1	25.1	21.6	19.0
4	19.9	24.1	21.5	22.1
5	20.5	24.7	22.6	24.2
Разом	104.2	123.6	112.1	113.0
Середнє арифметичне	20.84	24.72	22.42	22.6

Очевидно, що середні значення часу автономної роботи ноутбуків різних виробників дещо відрізняються. Необхідно визначити, чи є ці відмінності статистично значущими.

Розв’язання.

У цьому дослідженні виробник ноутбуків виступає як факторна ознака, вплив якої на надійність роботи пристроїв (тривалість роботи без підключення до електромережі) перевіряється за допомогою дисперсійного аналізу.

Для побудови таблиці дисперсійного аналізу необхідно виконати наступні кроки:

- обчислити середній час автономної роботи для кожної з груп пристроїв одного виробника. У останньому рядку табл. 5.6 наведено результати цих обчислень;
- знайти загальне середнє – середній час автономної роботи для всіх 20 ноутбуків. Для цього потрібно просумувати всі 20 значень часу роботи і поділити на кількість усіх спостережень ($n = 20$), за формулою середнього арифметичного:

$$\overline{x} = \frac{1}{n}\sum_{j=1}^{4}\sum_{i=1}^{5} x_{i,j} = 22.645 \text{ год.};$$

- обчислити суми квадратів відхилень:

$$SSB = \sum_{j=1}^{m}\left(\overline{x}_j - \overline{x}\right)^2 \cdot f_j = 38.0815,$$

$$SSW = \sum_{j=1}^{m}\sum_{i=1}^{n/m}\left(x_{i,j} - \overline{x}_j\right)^2 = 41.388,$$

$$SST = \sum_{j=1}^{n}\left(x_j - \overline{x}\right)^2 = 79.4695 = SSB + SSW;$$

- обчислити суми квадратів відхилень на ступінь свободи:

$$S_1^2 = \frac{SSB}{m-1} = \frac{38.0815}{3} = 12.6938,$$

$$S_2^2 = \frac{SSW}{n-m} = \frac{41.388}{16} = 2.58675;$$

- обчислити F-критерій Фішера:

$$F = \frac{S_1^2}{S_2^2} = \frac{12.6938}{2.58675} = 4.907252.$$

Для рівня значущості $\alpha = 0.05$ і числам степенів свободи $df_{bg} = 3$ і $df_{wg} = 16$ визначається критичне значення F-критерія по таблиці значень F-критерія Фішера: $F_{0.05(3,16)} = 3.24$. Обчислене значення F-критерія $F = 4.907252$ більше табличного, тобто відмінності між середніми значеннями часу роботи ноутбуків різних виробників від повністю зарядженого

акумулятора без підключення до живлення за умов максимального обчислювального навантаження статистично значущі, тобто гіпотеза H_0 відхиляється. Таблицю дисперсійного аналізу наведено в табл. 5.7.

Таблиця 5.7 – Однофакторний дисперсійний аналіз (приклад)

Джерело варіації	Число ступенів свободи	Сума квадратів відхилень	Середня сума квадратів відхилень	Критерій Фішера F	Критичне значення $F_{\alpha(df_{bg},df_{wg})}$
Міжгрупова	3	38.0815	12.69383	4.90725	3.24
Внутрішньогрупова	16	41.388	2.58675		
Загальна	19	79.4695			

Висновок. Оскільки отримане фактичне значення F-критерію перевищує критичне (табличне), це свідчить про те, що спостережувані відмінності в середньому часі автономної роботи між ноутбуками різних виробників не є випадковими. Отже, виробник справді впливає на показник надійності (тривалість автономної роботи), і нульову гіпотезу слід відхилити.

Непараметричним аналогом однофакторного дисперсійного аналізу є ранговий однофакторний критерій Краскела–Уолліса. Цей критерій застосовується для перевірки нульової гіпотези щодо рівності ефектів впливу у вибірках, середні значення яких невідомі, але вважаються рівними. При цьому кількість вибірок має перевищувати дві.

Нульова гіпотеза формулюється як твердження, що k вибірок обсягами $n_1, n_2, \ldots, n_k$ отримані з однієї і тієї самої генеральної сукупності. Рангові методи, до яких відноситься критерій Краскела–Уолліса, не потребують припущення про нормальність розподілу результатів спостережень і можуть застосовуватися як для кількісних даних з невідомим законом розподілу, так і для порядкових змінних.

Для кожної вибірки $i = \overline{1,k}$ розраховується сума рангів

$$R_j = \sum_{i=1}^{n_j} r_{ij},$$

причому

$$\sum_{i=1}^{k} R_i = \frac{N(N+1)}{2},$$

де $N = \sum_{i=1}^{k} n_i$ – уся сукупність спостережень.

Якщо між стовпцями не існує систематичної відмінності, то їхні середні ранги будуть наближені до загального середнього рангу, обчисленого для всієї сукупності, який дорівнює ${(N+1)}/{2}$. Відповідно, величини ${R_j}/{n_j} - {(N+1)}/{2}$ мають залишатися відносно малими за умови справедливості нульової гіпотези.

Обчислення критерію здійснюється за такою формулою:

$$H = \frac{12}{N(N+1)} \sum_{j=1}^{k} \frac{R_j^2}{n_j} - 3(N+1).$$

За умови $n_i \geq 5$ та $k \geq 4$ статистика критерію асимптотично наближається до χ^2-розподілу з кількістю ступенів свободи $k-1$. У цьому випадку нульову гіпотезу відхиляють на рівні значущості α, якщо $H > \chi^2_{1-\alpha,k-1}$, $\chi^2_{1-\alpha,k-1}$ – відповідний квантиль розподілу χ^2.

У разі наявності рівних значень серед спостережень описана схема аналізу може застосовуватися лише як наближена. Надійність висновків зменшується зі зростанням кількості збігів. Для підвищення достовірності використовують середні ранги; при цьому, якщо вони не є цілими числами, їх не округлюють. У випадках великої кількості збігів застосовують модифіковану форму статистики Краскела–Уолліса:

$$H' = \frac{H}{1 - \left(\sum_{j=1}^{g} \frac{T_j}{N^3 - N} \right)},$$

де g – кількість груп спостережень, що збігаються, $T_j = \left(t_j^3 - t_j\right)$, t_j – кількість спостережень, що збігаються в групі j.

Критерій Джонкхієра (Джонкхієра–Терпстра) застосовують у випадках, коли заздалегідь відомо, що групи результатів упорядковані за зростанням впливу досліджуваного фактора, вимірюваного в порядковій шкалі. Критерій Джонкхієра є більш потужним, ніж критерій Краскела–Уолліса, при перевірці гіпотези про монотонний вплив фактора.

Насамперед, для кожної пари вибірок з номерами $u, v (1 \le u < v \le k)$, де k – кількість рівнів фактора, обчислюють статистику Манна–Вітні:

$$U_{u,v} = \sum_{i=1}^{n_u} \sum_{j=1}^{n_v} \varphi(x_{ui}, y_{vj}),$$

де

$$\varphi(x_{ui}, y_{vj}) = \begin{cases} 1(x_{ui} < y_{vj}), \\ 0.5(x_{ui} = y_{vj}), \\ 0(x_{ui} > y_{vj}). \end{cases}$$

Далі обчислюється статистика Джонкхієра:

$$J = \sum_{1 \le u < v \le k} U_{u,v}.$$

Великі значення J свідчать проти нульової гіпотези про однорідність вибірок. Для вибірок великого обсягу статистика Джонкхієра апроксимується нормальним розподілом з параметрами:

$$M[J] = \frac{1}{4}\left(N^2 - \sum_{j=1}^{k} n_j^2\right), \qquad D[J] = \frac{1}{72}\left(N^2(2N+3) - \sum_{j=1}^{k} n_j^2(2n_j+3)\right).$$

Свідченням проти гіпотези однорідності є великі значення стандартизованої статистики

$$Z = \frac{J - M[J]}{\sqrt{D[J]}},$$

які перевищують відповідні квантилі стандартного нормального розподілу.

М-критерій Бартлетта застосовують для перевірки нульової гіпотези про рівність дисперсій кількох нормальних генеральних сукупностей, з яких

отримані вибірки, що в загальному випадку мають різні обсяги (за умови, що обсяг кожної вибірки не менший за чотири).

$$B = \frac{V}{C},$$

$$V = \left(\sum_{i=1}^{l}(k_i - 1)\right)\ln s^2 - \sum_{i=1}^{l}(k_i - 1)\ln s_i^2,$$

$$C = 1 + \frac{1}{3(l-1)}\left(\sum_{i=1}^{l}\frac{1}{k_i - 1} - \frac{1}{\sum_{i=1}^{l}(k_i - 1)}\right),$$

$$s^2 = \frac{\sum_{i=1}^{l}(k_i - 1)\, s_i^2}{\sum_{i=1}^{l}(k_i - 1)},$$

де s_i^2 – вибіркова дисперсія i-ї вибірки, k_i – обсяг i-ї вибірки, l – кількість вибірок, s^2 – зважене за кількістю ступенів вільності середнє значення вибіркових дисперсій.

G-критерій Кокрена (Кочрена) застосовують для перевірки нульової гіпотези про рівність дисперсій $k(k \geq 2)$ нормальних генеральних сукупностей на основі незалежних вибірок однакового обсягу. Значення статистики критерію обчислюють за формулою:

$$G = \frac{\max\limits_{1 \leq j \leq k} s_j^2}{\sum_{j=1}^{k} s_j^2},$$

де s_i^2 – вибіркова дисперсія j-ї групи.

Для вибірок рівного обсягу критерій Кокрена є більш потужним, ніж критерій Бартлетта. Критичні значення G визначають за спеціальними статистичними таблицями.

Непараметричний критерій Левене є альтернативою критерію Бартлетта у випадках, коли немає підстав вважати, що досліджувані вибірки підпорядковуються нормальному розподілу. Розрахункове значення статистики критерію обчислюється за формулою:

$$W = \frac{(N-k)\sum_{j=1}^{k} n_j\left(\overline{Z}_j - \overline{Z}\right)^2}{(k-1)\sum_{j=1}^{k}\sum_{i=1}^{n_j}\left(Z_{ij} - \overline{Z}_j\right)^2},$$

де $Z_{ij} = \left|x_{ij} - \overline{x}_j\right|$, x_{ij} – значення i-го спостереження в j-ій вибірці, $\overline{x}_j$ – середнє арифметичне значення спостережень у j-ій вибірці, $\overline{Z}$ – загальне середнє арифметичне величин Z_{ij} за всіма спостереженнями, $\overline{Z}_j$ – середнє арифметичне величин Z_{ij} у межах j-ї вибірки, n_j – обсяг j-ї вибірки, $N = \sum_{j=1}^{k} n_j$ – загальний обсяг вибірки.

Розраховане значення критерію W порівнюють із відповідним квантилем розподілу Фішера (F-розподілу) з числом ступенів свободи $(k-1)$ та $(N-k)$.

Більш робастним аналогом критерію Левене є критерій Брауна–Форсайта. Його відмінність полягає в способі визначення величин $Z_{ij} = \left|x_{ij} - \tilde{x}_j\right|$, де $\tilde{x}_j$ – медіана спостережень, що належать до j-ої вибірки.

Розглянуті критерії дозволяють встановлювати відмінності дисперсій сукупностей, але не дають змоги кількісно оцінити вплив фактора на досліджувану ознаку та визначити, для яких саме вибірок дисперсії відрізняються.

Для оцінки кількісного впливу досліджуваного фактора часто застосовують адитивну модель, яка передбачає, що значення ознаки є сумою впливу фактора та випадкової компоненти:

$$x_{ij} = a_j + \varepsilon_{ij}, \qquad j = 1, \dots, k, \qquad i = 1, \dots, n_j,$$

де a_j – невипадкові невідомі величини, що визначаються рівнями фактора, ε_{ij} – незалежні випадкові величини з однаковим розподілом, що відображають внутрішню мінливість, не пов’язану з рівнями фактора.

Цю модель можна також подати у вигляді:

$$x_{ij} = \mu + \tau_j + \varepsilon_{ij},$$

де $\mu = \frac{1}{k}\sum_{j=1}^{k} a_j$ – середній рівень, $\tau_j = a_j - \mu$ – відхилення від середнього для j-го рівня фактора. Незважаючи на іншу форму, кількість незалежних параметрів залишається рівною k, оскільки виконується умова $\sum_{j=1}^{k} \tau_j = 0$.

Для оцінки різниці відгуків між двома рівнями фактора (так званого зсуву) часто застосовують медіану Ходжеса–Лемана:

$$z_{uv} = \text{med}\{x_{ui} - x_{vi}, i = 1, \dots, n_u, j = 1, \dots, n_v\},$$

де $z_{uv} = -z_{vu}$. Таку статистику можна використовувати для оцінки $\tau_i - \tau_j$, але вона не задовольняє властивості адитивності: $z_{ij} + z_{jk} \neq z_{ik}$. Щоб виправити цей недолік, застосовують зважені скореговані оцінки зсуву (оцінки Спетволя):

$$W_{ij} = \Delta_i + \Delta_j,$$

де $\Delta_i = \frac{\sum_{u=1}^{k} n_u z_{iu}}{N}$ – зсув i-ї вибірки відносно всіх інших, усереднений з ваговими коефіцієнтами n_u. Ці оцінки задовольняють адитивності: $W_{ij} + W_{jk} = W_{ik}$. Недоліком є те, що оцінка зсуву між двома вибірками залежить від всіх інших вибірок.

Якщо нульову гіпотезу про рівність середніх відхиляють, наступним кроком є визначення, між якими вибірками існує суттєва різниця. Для цього застосовують метод лінійних контрастів. Лінійним контрастом називають лінійну комбінацію середніх значень вибірок:

$$L = \sum_{j=1}^{k} c_j m_j,$$

де c_j – відомі коефіцієнти, що задовольняють $\sum_{j=1}^{k} c_j = 0$, m_j – середнє арифметичне j-ї вибірки. Дисперсію лінійного контрасту розраховують за формулою:

$$D[L] = \sum_{j=1}^{k} \frac{c_j^2 s_j^2}{n_j},$$

де s_j^2 – вибіркова дисперсія j-ї вибірки. Довірчий інтервал для L визначають за допомогою F-розподілу.

Найпростішим прикладом лінійного контрасту є різниця двох середніх: $m_i - m_j$, для якої $c_i = 1$, $c_j = -1$, інші $c_u = 0$. Нульову гіпотезу приймають, якщо 0 потрапляє у відповідний довірчий інтервал.

Для виявлення вибірок із істотно різними дисперсіями часто застосовують метод множинних порівнянь Шеффе. Статистику для перевірки нульової гіпотези $L = L_0$ обчислюють за формулою:

$$t = \frac{\sum_{i=1}^{k} c_i \overline{x}_i - L_0}{\sqrt{M/N \sum_{i=1}^{k} \frac{c_i^2}{n_i}}},$$

де N – загальний обсяг спостережень, n_i – обсяг i-ї вибірки, $\overline{x}_i$ – середнє значення вибірки, M – середній квадратичний залишок. Розраховане значення порівнюють з критичним модулем для прийняття або відхилення гіпотези.

Випадкова величина може зазнавати впливу не одного, а кількох чинників. Інакше кажучи, однофакторна модель може виявитися незначущою, якщо вплив фактора А, визначений за допомогою F-відношення, не є істотним на тлі великого внутрішньогрупового розсіювання (залишкової варіації). Це розсіювання може бути зумовлене не лише випадковими причинами, але й дією ще одного, «зовнішнього» фактора В.

У такій ситуації фактор В додатково включається до моделі – проводиться двофакторний дисперсійний аналіз з метою зменшення впливу неврахованих чинників і посилення виявлення систематичних ефектів. Аналогічним чином виникає потреба в розгляді трьох- або багатофакторних моделей. Відповідно, у процесі дисперсійного аналізу змінюються не один, а кілька факторів.

Принципової різниці між багатофакторним і однофакторним дисперсійним аналізом не існує. Багатофакторний підхід не змінює загальну логіку дисперсійного аналізу, а лише дещо її ускладнює, оскільки, окрім оцінювання впливу кожного фактора окремо на залежну змінну, необхідно також оцінити їхню спільну дію (взаємодію). Отже, основне нововведення багатофакторного аналізу – це можливість врахування взаємодії між факторами. Водночас зберігається можливість окремо аналізувати вплив

кожного фактора. У цьому контексті багатофакторний дисперсійний аналіз (особливо при його комп'ютерній реалізації) є більш ефективним, адже дозволяє в межах одного обчислення одночасно оцінити як вплив кожного фактора, так і характер їхньої взаємодії.

Матриця спостережень при двофакторному дисперсійному аналізі на відміну від матриці однофакторного дисперсійного аналізу (в якій дані всередині кожного стовпця є однорідними, тобто утворюють вибірки з однаковими умовами) містить спостереження, які в стовпцях не є однорідними, якщо вплив додаткового (заважаючого) фактора є суттєвим. Інакше кажучи, структура варіації порушується через присутність додаткового джерела впливу.

При позначенні внеску фактора А на рівні i як a_i, а внеску фактора В на рівні j як b_j та без урахування взаємодії між факторами А і В (ефекти факторів не залежать один від одного) кожне спостереження x_{ij} може бути подане у вигляді адитивної моделі:

$$x_{ij} = a_i + b_j + \varepsilon_{ij}, i = \overline{1,n}, j = \overline{1,m}.$$

Як і у випадку однофакторного дисперсійного аналізу, вважається, що випадкові похибки ε_{ij} підпорядковуються нормальному закону розподілу $N(0, \sigma_\varepsilon^2)$, при цьому дисперсія σ_ε^2 є однаковою для всіх комбінацій рівнів факторів i та j.

Внески факторів a_i та b_j не можна однозначно відновити лише з вихідних даних. Наприклад, якщо збільшити всі b_j на деяку постійну величину і водночас зменшити всі a_i на цю ж саму величину, то значення x_{ij} залишаться незмінними. Це свідчить про наявність надлишковості параметрів у моделі.

Щоб уникнути цієї неоднозначності та забезпечити коректну інтерпретацію внесків кожного фактора, використовуються відхилення τ_i та β_j, що відображають вплив факторів А та В відповідно, від загального

середнього значення μ (яке оцінюється за допомогою середнього значення всієї вибірки $\overline{x}$):

$$\tau_i + \beta_j = a_i + b_j - \mu,$$

$$\sum_{i=1}^{n} \tau_i = 0, \qquad \sum_{j=1}^{m} \beta_j = 0.$$

Ефекти обробки $\tau_1, \dots, \tau_n$ – це відхилення, що виникають у результаті впливу фактора А, і визначають, як саме змінюється відгук залежно від цього фактору. Величини $\beta_j, \dots, \beta_m$ називаються ефектами блоків – вони відображають відхилення значень від загального середнього внаслідок дії фактора В.

Тоді:

$$x_{ij} = \mu + \tau_i + \beta_j + \varepsilon_{ij}, i = \overline{1,n}, j = \overline{1,m}.$$

Як і в однофакторному дисперсійному аналізі, нульова гіпотеза H_0 про відсутність ефектів обробки має вигляд: $\tau_1 = \tau_2 = \cdots = \tau_n = 0$. У двофакторному дисперсійному аналізі загальна сума квадратів відхилень розкладається не на дві, а вже на три складові: $SST = SA + SB + SSR$. SA та SB зумовлені впливом факторів А і В, а залишкова частина SSR (Sum of Squares Residuals) – відображає випадкову змінність самих спостережень, яка виникає через дію неврахованих факторів. Інакше кажучи, це частина варіації, яку не можна пояснити впливом досліджуваних факторів А і В.

$$\sum_{j=1}^{m}\sum_{i=1}^{n}\left(x_{i,j} - \overline{x}\right)^2 =$$

$$= m\sum_{i=1}^{n}(\overline{x}_i - \overline{x})^2 + n\sum_{j=1}^{m}\left(\overline{x}_j - \overline{x}\right)^2 + \sum_{j=1}^{m}\sum_{i=1}^{n}\left(x_{i,j} - \overline{x}_j - \overline{x}_i + \overline{x}\right)^2,$$

де $\overline{x}_i$ – середнє по i-му стовпцю матриці спостережень, $\overline{x}_j$ – середнє по j-му блоку матриці спостережень. Матрицю двофакторного дисперсійного аналізу наведено в табл. 5.8.

Таблиця 5.8 – Двофакторний дисперсійний аналіз

Джерело варіації	Число ступенів свободи	Сума квадратів відхилень	Середня сума квадратів відхилень
Головні ефекти	$n+m-2$	$SA+SB$	$S_A^2=\frac{SA+SB}{n+m-2}$
Фактор А	$n-1$	SA	$S_a^2=\frac{SA}{n-1}$
Фактор В	$m-1$	SB	$S_b^2=\frac{SB}{m-1}$
Залишкова варіація	$(n-1)(m-1)$	SSR	$S_R^2=\frac{SSR}{(n-1)(m-1)}$
Повна варіація	$nm-1$	SST	$S^2=\frac{SST}{nm-1}$

При виконанні гіпотези H_0 про відсутність ефектів обробки сума статистик S_A^2 і S_R^2 є незміщеною, збіжною та ефективною оцінкою загальної дисперсії $D_x=\sigma_x^2$.

Для перевірки нульової гіпотези дисперсія за фактором А порівнюється із залишковою дисперсією. З цією метою обчислюється F-статистика:

$$F_a=\frac{S_a^2}{S_R^2},$$

що має F-розподіл зі ступенями свободи $(n-1)$ і $(n-1)(m-1)$.

Чим більшими є відмінності між ефектами обробки $\tau_1,\dots,\tau_n$, тим більшою є тенденція F-статистики до зростання. На рівні значущості α гіпотеза H_0 відкидається, якщо

$$F_a > F_{\alpha(n-1;(n-1)(m-1))},$$

де $F_{\alpha(n-1;(n-1)(m-1))}$ – критичне значення.

У цьому випадку вплив фактора А на відгук слід вважати статистично значущим. Аналогічно перевіряється гіпотеза про відсутність впливу фактора В.

За F-відношенням

$$F_A=\frac{S_A^2}{S_R^2},$$

перевіряється значущість двофакторної моделі з незалежною дією факторів.

Якщо припущення, необхідні для застосування двофакторного дисперсійного аналізу, не виконуються, використовують непараметричний ранговий критерій Фрідмана (Фрідмана, Кендалла та Сміта). Цей критерій не залежить від типу розподілу досліджуваних величин. Передбачається лише, що випадкові компоненти ε_{ij} мають однаковий неперервний розподіл та незалежні одна від одної.

Для перевірки нульової гіпотези $H_0: \tau_1 = \tau_2 = \cdots = \tau_k = 0$ вихідні дані подаються у вигляді прямокутної матриці, де n рядків відповідають рівням фактора B, а k стовпців – рівням фактора A. Кожна комірка матриці відображає результат вимірювання певного параметра на окремому об'єкті або на групі об'єктів за фіксованих рівнів обох факторів. У разі використання групи об'єктів у таблицю вносяться середні значення параметра, отримані за всіма вимірюваннями у вибірці.

Для застосування критерію необхідно перетворити вихідні дані від безпосередніх результатів вимірювань до їхніх рангів. Ранжирування виконується окремо в межах кожного рядка для кожного фіксованого значення і упорядковуються величини $x_{i,j}$, отримуючи k рангів $r_{i,j}$. Такий підхід дозволяє нівелювати вплив фактора B, що має однакові значення в межах кожного рядка. Обчислення критерію виконується за формулою:

$$S = \left[\frac{12}{nk(k+1)}\sum_{j=1}^{k}\left(\sum_{i=1}^{n} r_{i,j}\right)^2\right] - 3n(k+1).$$

Для перевірки нульової гіпотези $H_0: \beta_1 = \beta_2 = \cdots = \beta_n = 0$ вихідні дані впорядковуються за стовпцями та процедура повторюється з заміною n на k і навпаки.

За умови справедливості нульової гіпотези та при $n \to \infty$ S-статистика Фрідмана асимптотично збігається зі статистикою χ^2 із $k-1$ ступенями свободи. Відповідно, нульова гіпотеза приймається на рівні значущості α, якщо виконується умова

$$S < \chi^2_{1-\alpha,k-1}.$$

Критерій Пейджа (L-критерій) використовується для перевірки нульових гіпотез $H_0: \tau_1 = \tau_2 = \cdots = \tau_k = 0$ або $H_0: \beta_1 = \beta_2 = \cdots = \beta_n = 0$ проти впорядкованих альтернатив

$$\tau_1 < \tau_2 < \cdots < \tau_k \text{ або } \beta_1 < \beta_2 < \cdots < \beta_n,$$

де принаймні одна з нерівностей є строгою. Для впорядкованих альтернатив L-критерій має вищу потужність порівняно з критерієм Фрідмана.

Значення статистики визначається за формулами:

$$L_1 = \sum_{j=1}^{k} j \sum_{i=1}^{n} r_{i,j}\,, \qquad L_2 = \sum_{i=1}^{n} i \sum_{j=1}^{k} r_{i,j}.$$

Для великих вибірок застосовується апроксимація статистики Пейджа:

$$L_1^* = \frac{L_1 - \frac{1}{4}nk(k+1)}{\sqrt{\frac{1}{144}nk(k+1)(k-1)}}, \qquad L_2^* = \frac{L_2 - \frac{1}{4}nk(n+1)}{\sqrt{\frac{1}{144}nk(n+1)(n-1)}},$$

які за умови справедливості відповідних нульових гіпотез мають асимптотичний стандартний нормальний розподіл.

У випадку наявності однакових значень у рядках вихідної матриці використовуються середні ранги. При цьому точність статистичних висновків знижується пропорційно до кількості таких збігів.

Q-критерій Кокрена застосовують у випадках, коли групи однорідних суб'єктів зазнають впливів, кількість яких перевищує два, і для кожного з впливів можливі лише два варіанти відгуку – умовно-негативний (0) та умовно-позитивний (1). Нульова гіпотеза формулюється як твердження про відсутність відмінностей між ефектами впливу. Значення критерію обчислюється за формулою:

$$Q = (c-1)\frac{c\sum_{j=1}^{c} T_j^2 - \left(\sum_{j=1}^{c} T_j\right)^2}{c\sum_{i=1}^{r} T_i - \sum_{i=1}^{r} T_i^2},$$

де $T_j = \sum_{i=1}^{r} x_{i,j}\,, j = 1,2,\ldots,c$ – суми стовпців, $T_i = \sum_{j=1}^{c} x_{i,j}\,, i = 1,2,\ldots,r$ – суми рядків, c – кількість стовпців (вибірок), r – кількість рядків (параметрів).

Довірчий рівень перевірки визначається за функцією розподілу χ^2 з числом ступенів свободи $c-1$.

Оскільки двофакторний дисперсійний аналіз дозволяє встановити лише наявність ефектів впливу, але не ідентифікує, для яких саме стовпців ці ефекти є значущими, застосовується метод множинних порівнянь Шеффе для пов'язаних вибірок. Значення критерію Шеффе обчислюється за формулою:

$$t = \frac{(\sum_{i=1}^{r} c_i \overline{x}_i)^2}{\frac{(r-1)S}{c} \sum_{i=1}^{r} c_i^2},$$

де $c_i, i = 1,2,\dots,r$ – константи, $S = \sum_{i=1}^{r}\sum_{j=1}^{c} x_{i,j}^2 - \frac{\left(\sum_{i=1}^{r}\sum_{j=1}^{c} x_{i,j}\right)^2}{rc}$ – залишковий середній квадрат, c – кількість стовпців (вибірок), r – кількість рядків (параметрів).

Рівень значущості визначається за функцією розподілу Фішера:

- для перевірки ефекту рядків – з параметрами $(r-1)$ та $(r-1)(c-1)$;
- для перевірки ефекту стовпців – із параметрами $(c-1)$ та $(r-1)(c-1)$.

5.3. Сутність кореляційного аналізу

Вивчення зв'язків між змінними становить інтерес для дослідника з погляду виявлення причинно-наслідкових залежностей.

Дві випадкові величини X_1 і X_2 можуть бути:

- пов'язані функціональною залежністю – тобто жорстко, як це властиво залежності змінних у математичному аналізі;
- незалежними – коли значення однієї не впливає на розподіл іншої;
- пов'язані стохастичною (ймовірнісною) залежністю, за якої зміна однієї величини спричиняє зміну ймовірнісного розподілу іншої.

Коваріація визначає величину статистичного зв’язку між двома випадковими величинами X_1 і X_2. Хоча коваріація дозволяє визначити напрямок зв’язку між змінними (позитивний або негативний), вона має певні обмеження:

- вона є розмірною величиною, тобто залежить від одиниць вимірювання змінних;
- її числове значення може бути будь-яким (позитивним, від’ємним або нульовим), і не має уніфікованої шкали для порівняння сили зв’язку.

Саме через це коваріація менш зручна для практичного аналізу залежностей між змінними і менш поширена в статистичній практиці, ніж коефіцієнт кореляції, який є безрозмірним і нормованим показником сили зв’язку.

Під час роботи з експериментальними даними похибки вимірювань можуть суттєво відрізнятися залежно від характеру фізичних процесів. Для наближення (апроксимації) експериментальних точок певною функцією необхідно насамперед встановити наявність залежності між досліджуваними величинами. З цією метою застосовуються методи кореляційного аналізу.

У більшості випадків дослідник має в розпорядженні лише таблицю результатів експерименту, яку називають ще кореляційною таблицею.

Кореляційна таблиця – це таблиця експериментальних даних або результатів спостережень, підготовлена для побудови графіків, попереднього аналізу та подальшої числової обробки. У ряді випадків наявність кореляційної залежності можна встановити візуально, побудувавши кореляційне поле.

Кореляційне поле – це система координат (x, y) з нанесеними на площину точками двовимірної вибірки. По суті, таке поле є графічним зображенням експериментальної залежності $y = f(x)$.

За допомогою кореляційного аналізу можна визначити, наскільки тісним є зв'язок між двома (або більше) випадковими величинами, що спостерігаються та фіксуються під час моделювання конкретної системи [3].

Кореляційний аналіз результатів моделювання зводиться до оцінки розкиду значень $x_1, x_2, \dots, x_n$ випадкової величини відносно середнього значення $\overline{x}$, тобто до оцінювання сили кореляційного зв'язку.

Кореляційний аналіз дозволяє розв'язувати такі ключові задачі:

- виявлення наявності зв'язку між результативною ознакою та одним або кількома чинниками, а також встановлення характеру (форми) цього зв'язку;
- кількісне визначення того, як змінюється залежна змінна під впливом відповідних факторів;
- оцінка сили зв'язку між результативним показником та окремими факторами або їхньою сукупністю;
- аналіз загальної варіації залежної ознаки та визначення внеску кожного окремого чинника в цю змінність;
- здійснення статистичної перевірки достовірності отриманих показників кореляційної залежності.

Основні етапи кореляційного аналізу охоплюють:

- відбір найважливіших ознак для подальшого дослідження;
- визначення напряму (прямий чи обернений) та характеру (лінійний, нелінійний) взаємозв'язку між залежною та незалежними змінними, а також вибір відповідної математичної моделі;
- обчислення показників, що описують силу та форму кореляційного зв'язку;
- статистичне оцінювання надійності та значущості одержаних кореляційних коефіцієнтів.

Кореляційна залежність – це узгоджені зміни двох (парна кореляція) або більшої кількості ознак (множинна кореляція). Її сутність полягає в тому, що зміна значення однієї змінної супроводжується закономірною зміною (зростанням або зменшенням) іншої змінної або інших змінних.

Кореляційний аналіз – це статистичний метод, який дозволяє за допомогою коефіцієнтів кореляції визначити, чи існує зв'язок між змінними і наскільки він тісний.

Коефіцієнт кореляції є двовимірною описовою статистикою, тобто кількісною мірою взаємозв'язку (спільної змінюваності) двох змінних.

На рис. 5.3 наведені кореляційні поля при різних значеннях коефіцієнта кореляції r.

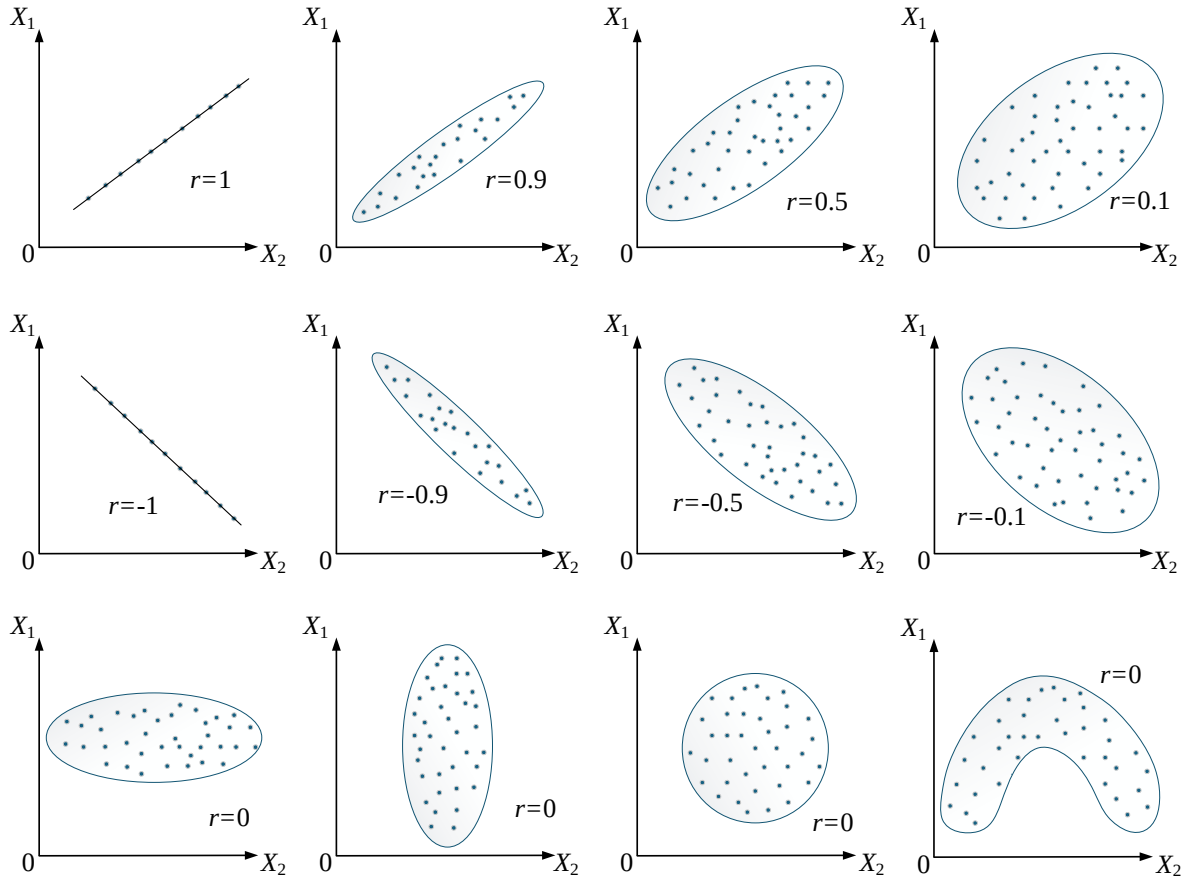


Рисунок 5.3 – Кореляційні поля при різних значеннях коефіцієнта кореляції

Абсолютне значення коефіцієнта кореляції відображає силу взаємозв'язку. Знак коефіцієнта кореляції вказує на напрям зв'язку:

- плюс $(+)$ – позитивна кореляція (із зростанням однієї змінної зростає й інша);
- мінус $(-)$ – негативна кореляція (зі збільшенням однієї змінної інша зменшується).

При позитивній лінійній кореляції вищим значенням однієї змінної зазвичай відповідають вищі значення іншої, а нижчим – нижчі. У випадку негативної лінійної кореляції більш високі значення однієї змінної пов’язані з нижчими значеннями іншої, і навпаки.

Основні види зв’язку між змінними:

- прямий причинно-наслідковий зв’язок – значення змінної X_1 прямо визначає значення змінної X_2;
- зворотній причинно-наслідковий зв’язок – значення X_2 визначає змінну X_1;
- спільний фактор – залежність, зумовлена третьою (прихованою) змінною X_3, що визначає одночасно і X_1 і X_2;
- двоспрямована або циклічна залежність – значення змінної X_1 визначає значення змінної X_2, а X_2, у свою чергу, визначає значення змінної X_1;
- залежність, зумовлена кількома прихованими змінними;
- випадкова (псевдозалежність) без реального зв’язку.

Важливо: навіть якщо між двома змінними існує тісна кореляція, це ще не означає, що один показник є причиною іншого. Нерідко на результати впливають сторонні, не враховані чинники. У таких випадках кореляцію називають хибною (або спотвореною).

Для виявлення хибної залежності застосовується часткова кореляція: якщо між двома змінними спостерігається кореляція, можливо, що вона зумовлена дією третьої змінної, яка є спільною причиною їх коливань. Для перевірки цього вплив третьої змінної усувається, і кореляція між

початковими змінними розраховується за фіксованих значень цієї третьої змінної. Отримана таким чином кореляція називається частковою.

Для порядкових (рангових) даних застосовують такі коефіцієнти кореляції:

- ρ – коефіцієнт рангової кореляції Спірмена:

$$\rho = 1 - \frac{6}{N(N^2-1)}\sum_{i=1}^{N}\left(x^i - y^i\right)^2,$$

де x^i – ранг фактора X_i, y^i – ранг фактора Y_i, i – номер спостереження, N – обсяг вибірки (кількість спостережень);

- τ – коефіцієнт рангової кореляції Кендалла.

Коефіцієнт τ обчислюється на основі ранжирування ознак. Якщо розглянути всі можливі пари об'єктів $(i,j), i,j = 1,2,\dots,N$, то про кожну пару можна сказати, однаково або ні впорядковані ознаки в цій парі. Через P позначається кількість пар об'єктів, в яких ознаки x і y впорядковані однаково, через Q – кількість пар об'єктів, в яких ознаки x і y впорядковані по різному. Тоді

$$P - Q = \sum_{i<j} sgn\left(x^i - x^j\right) \cdot sgn\left(y^i - y^j\right),$$

де x^i – ранг фактора X_i, y^i – ранг фактора Y_i, i – номер спостереження,

$$sgn(u) = \begin{cases} 1, u > 0; \\ 0, u = 0; \\ -1, u < 0. \end{cases}$$

Розрахунок коефіцієнта рангової кореляції Кендалла:

$$\tau = \frac{2(P-Q)}{N(N-1)};$$

- γ – коефіцієнт рангової кореляції Гудмена – Краскела.

Якщо в даних є багато співпадаючих значень, то використовується статистика Гудмена–Краскела:

$$\gamma = \frac{P-Q}{P+Q}.$$

Ця міра має пряму ймовірнісну інтерпретацію, оскільки вона є різницею між відносними частотами впорядкованих і невпорядкованих пар для двох сукупностей.

Для змінних з інтервальною або номінальною шкалою використовується коефіцієнт кореляції Пірсона (також відомий як кореляція добутків відхилень від середнього, або кореляція моментів):

$$r_{x,y} = \frac{cov(x,y)}{\sigma_x \sigma_y} = \frac{\sum_{i=1}^{N}\left(X_i - \overline{X}\right)\left(Y_i - \overline{Y}\right)}{\sqrt{\sum_{i=1}^{N}\left(X_i - \overline{X}\right)^2 \cdot \sum_{i=1}^{N}\left(Y_i - \overline{Y}\right)^2}}.$$

Для визначення взаємного впливу декількох змінних будується кореляційна матриця, де вказуються коефіцієнти попарно для усіх змінних, причому відносно головної діагоналі елементи матриці симетричні, а всі елементи головної діагоналі (r_{11}, r_{22},.., r_{nn}) матриці дорівнюють одиниці, бо відображають зв'язок величини із собою:

$$R = \begin{pmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ r_{21} & r_{22} & \dots & r_{2n} \\ \dots & \dots & \dots & \dots \\ r_{n1} & r_{n2} & \dots & r_{nn} \end{pmatrix} = \begin{pmatrix} 1 & r_{12} & \dots & r_{1n} \\ r_{21} & 1 & \dots & r_{2n} \\ \dots & \dots & \dots & \dots \\ r_{n1} & r_{n2} & \dots & 1 \end{pmatrix}.$$

Симетричність матриці дозволяє в програмах обробки статистичної інформації виводити лише половину матриці, наприклад, нижче головної діагоналі.

Коефіцієнт кореляції Пірсона оцінює лише лінійний зв'язок між змінними. Він не здатний виявити нелінійні залежності (останній графік на рис. 5.3). Також він є дуже чутливим до викидів (аномальних значень). Також слід пам'ятати, що кореляція не означає причинно-наслідкового зв'язку між змінними.

Прикладом, який демонструє нечутливість коефіцієнта кореляції Пірсона є квартет Анскомбе для чотирьох послідовностей даних (табл. 5.9), для яких статистичні характеристики ідентичні, але з різними кореляційними полями. Перший набір – це лінійна модель з шумом, у другому наборі

помітна квадратична залежність, третій – лінійна залежність з викидом, а останній – константа з викидом.

Таблиця 5.9 – Набори даних для квартета Анскомбе

I		II		III		IV	
x	y	x	y	x	y	x	y
10.0	8.04	10.0	9.14	10.0	7.46	8.0	6.58
8.0	6.95	8.0	8.14	8.0	6.77	8.0	5.76
13.0	7.58	13.0	8.74	13.0	12.74	8.0	7.71
9.0	8.81	9.0	8.77	9.0	7.11	8.0	8.84
11.0	8.33	11.0	9.26	11.0	7.81	8.0	8.47
14.0	9.96	14.0	8.10	14.0	8.84	8.0	7.04
6.0	7.24	6.0	6.13	6.0	6.08	8.0	5.25
4.0	4.26	4.0	3.10	4.0	5.39	19.0	12.50
12.0	10.84	12.0	9.13	12.0	8.15	8.0	5.56
7.0	4.82	7.0	7.26	7.0	6.42	8.0	7.91
5.0	5.68	5.0	4.74	5.0	5.73	8.0	6.89

Всі набори мають однакові статистичні характеристики:

$$\overline{x} = 9.0, \qquad \overline{y} = 7.5,$$

$$D_x = 10, \qquad D_y = 3.75,$$

$$r_{x,y} = 0.816.$$

Кореляційні поля для квартета Анскомбе наведені на рис. 5.4.

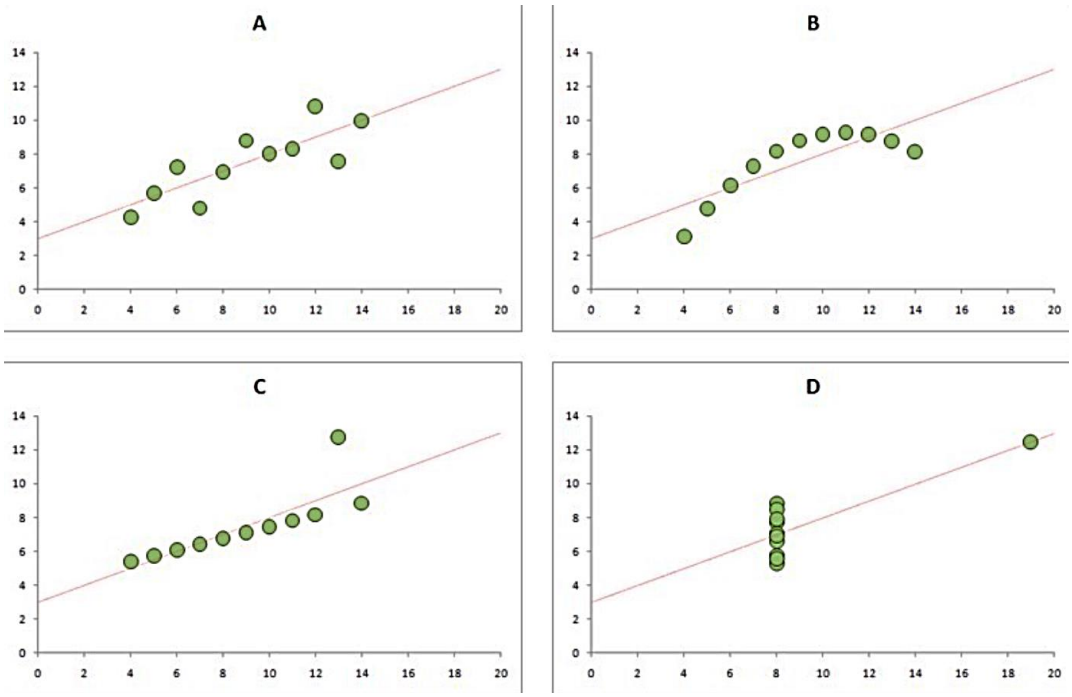


Рисунок 5.4 – Кореляційні поля для квартета Анскомбе

Попри «статистичну ідентичність», видно, що на рис. 5.4 наведено зовсім різні набори з погляду вибору моделі, яка описує дані.

Набори даних, які є однаковими за низкою статистичних характеристик, але дають різні графіки, часто використовуються для ілюстрації важливості графічного подання під час аналізу даних.

Якщо хоча б одна з двох змінних має порядкову шкалу або не є нормально розподіленою, доцільно використовувати рангову кореляцію Спірмена або Кендалла.

Застосування коефіцієнта Кендалла є більш доцільним у випадках, коли в даних наявні викиди (аномальні значення).

Щоб забезпечити достовірні і значущі результати при застосуванні кореляційного аналізу, слід:

- забезпечити достатній обсяг вибірки – кореляційний аналіз вимагає репрезентативної вибірки для отримання надійних висновків. Чим більша вибірка, тим нижчою є ймовірність виявлення випадкових або хибних кореляцій;
- врахувати контекст дослідження – при тлумаченні коефіцієнтів кореляції важливо враховувати контекст аналізованих змінних, це допомагає правильно інтерпретувати отримані статистичні залежності;
- виконати аналіз можливих нелінійних зв’язків – кореляційний аналіз базується на припущенні про лінійну залежність між змінними. Проте в реальних умовах ці зв’язки можуть мати нелінійний характер. Візуалізація даних у вигляді діаграм розсіювання дає змогу виявити можливі нелінійні патерни у взаємозв’язках між змінними.

Переваги кореляційного аналізу:

- узагальненість виявлення закономірностей – кореляційний аналіз дає змогу виявляти взаємозв’язки між ознаками в більш

загальному, абстрагованому вигляді, ніж це можливо за допомогою, наприклад, статистичного групування;

- зменшення впливу випадкових коливань – завдяки статистичній природі методів, кореляційний аналіз дозволяє мінімізувати вплив випадкових флуктуацій у даних;
- багатовимірний підхід – можливість одночасного аналізу взаємозв'язків між кількома змінними;
- оцінка сили зв'язку – отримання кількісних показників тісноти зв'язку між змінними (коефіцієнтів кореляції);
- екстраполяція на генеральну сукупність – за допомогою вибіркових коефіцієнтів кореляції та регресії можна здійснювати оцінку відповідних характеристик генеральної сукупності з заданим рівнем статистичної ймовірності;
- інформаційна щільність – методи кореляційного аналізу забезпечують високу щільність інформації при аналізі залежностей між ознаками.

5.4. Регресійний аналіз

Після проведення кореляційного аналізу, який дозволяє виявити наявність статистично значущих зв'язків між змінними та оцінити їхню силу, зазвичай переходять до математичного опису форми залежності із застосуванням методів регресійного аналізу.

Для точного задання рівняння регресії необхідно знати умовний закон розподілу результативної змінної Y. Проте, у статистичній практиці така інформація, як правило, є недоступною, тому обмежуються пошуком адекватних апроксимацій функції $f(x_1, x_2, \dots, x_m)$, побудованих на основі попереднього змістовного аналізу досліджуваного явища або вихідних статистичних даних.

У підсумку стохастичний зв’язок між Y та множиною факторних ознак X зводиться до строгих детермінованих співвідношень, які відображають основну тенденцію залежності $Y = f(x_1, x_2, \dots, x_m) + E$.

Використання регресійного аналізу ґрунтується на підході «чорної скриньки», коли досліджуваний об’єкт розглядається як система, на вхід якої подається вектор незалежних змінних (рис. 5.5).

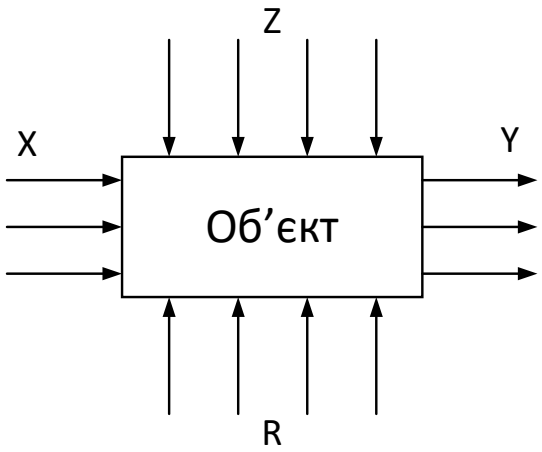


Рисунок 5.5 – Схематичне зображення «чорної скриньки»

На досліджуваний об’єкт впливають такі типи чинників:

- фактори X – незалежні змінні, які можна виміряти і контролювати;
- некеровані фактори Z – незалежні змінні, які можна виміряти, але неможливо контролювати;
- неконтрольовані фактори R – змінні, що впливають на об’єкт, проте їх неможливо безпосередньо спостерігати чи виміряти.

Вхідні параметри (фактори), що використовуються в регресійному аналізі, повинні задовольняти таким вимогам:

- фактори X мають істотно впливати на вихідну змінну Y;
- фактори X повинні бути керованими, тобто їх можна змінювати або задавати за вибором експериментатора;
- фактори X повинні бути незалежними між собою;

- точність вимірювання факторів X має бути щонайменше на порядок вищою за точність вимірювання вихідної змінної Y.

Головною метою регресійного аналізу є побудова емпіричної функції регресії, що якомога точніше відображає справжню функціональну залежність.

Для досягнення цієї мети розроблено методи регресійного аналізу, що дають змогу розв'язувати такі основні задачі:

- вибір загальної форми або класу функції регресії (регресійної моделі);
- оцінювання параметрів регресійної моделі (рівняння регресії);
- аналіз точності побудованого рівняння регресії;
- перевірка адекватності моделі статистичним даним;
- уточнення структури обраної функціональної залежності.

На рис. 5.6 наведено схему взаємозв'язку зазначених задач.

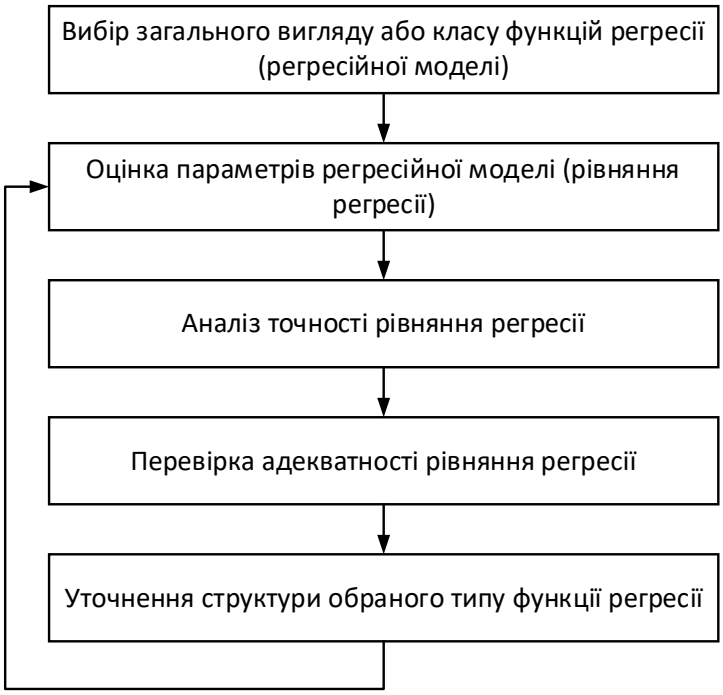


Рисунок 5.6 – Етапи регресійного аналізу

З наведеної схеми зв'язку видно, що у випадку, коли обране рівняння регресії виявляється неадекватним статистичним даним вибірки,

здійснюється уточнення його структури, після чого повторно проводяться оцінка параметрів, аналіз точності та перевірка адекватності оновленої моделі.

З метою найбільш точного отримання за вихідними статистичними даними значення результативного показника та невідомої функції регресії найчастіше застосовують:

1. Метод найменших квадратів – полягає в мінімізації квадрата відхилень спостережуваних значень результативного показника y_i, $i = 1, 2, \dots, n$, від модельних значень $\tilde{y}_i = f(x_i, \beta)$, де $\beta = (\beta_0, \beta_1, \dots, \beta_k)^T$ – коефіцієнти рівняння регресії, x_i – вектор аргументів у i-му спостереженні:

$$\sum_{i=1}^{n} \left(y_i - f(x_i, \beta)\right)^2 \to \min_{\beta}.$$

Розв'язання цієї задачі дає середньоквадратичну регресію.

2. Метод найменших модулів – полягає в мінімізації суми абсолютних відхилень спостережуваних значень від модельних:

$$\sum_{i=1}^{n} |y_i - f(x_i, \beta)| \to \min_{\beta}.$$

Отримувана при цьому регресія називається середньоабсолютною (медіанною).

3. Метод мінімаксу – зводиться до мінімізації найбільшого за модулем відхилення спостережуваного значення від модельного:

$$\max_{1 \le i \le n} |y_i - f(x_i, \beta)| \to \min_{\beta}.$$

У результаті отримується мінімаксна регресія.

У практичних задачах часто розглядають випадкову величину Y, що залежить від певної множини змінних $x_1, x_2, \dots, x_k$ та невідомих параметрів $\beta_0, \beta_1, \dots, \beta_k$. Тоді сукупність $(y, x_1, x_2, \dots, x_k)$ розглядається як $(k+1)$-вимірна генеральна сукупність, з якої відібрана випадкова вибірка обсягом n. Кожне спостереження має вигляд $(y_i, x_{i1}, x_{i2}, \dots, x_{ik})$, $i = 1, 2, \dots, n$. Задача полягає в оцінюванні невідомих параметрів $\beta_0, \beta_1, \dots, \beta_k$ за результатами

спостережень. Описана задача є однією з основних задач регресійного аналізу.

Найчастіше зустрічається лінійна функціональна залежність між величинами. Лінійна регресія спрямована на передбачення значення залежної (цільової) змінної на основі значень однієї або кількох незалежних змінних. Множинна лінійна регресія є узагальненням простої лінійної регресії, оскільки передбачає використання кількох незалежних змінних для пояснення та прогнозування цільової змінної. Лінійна регресія ґрунтується на простій математичній моделі, яка описує залежність між змінними. Загальний вигляд рівняння лінійної регресії можна подати так:

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k + \varepsilon$$

Основною метою методу лінійної регресії є оцінювання параметрів $\beta_0, \beta_1, \ldots, \beta_k$ таким чином, щоб мінімізувати різницю між прогнозованими значеннями та фактичними спостереженнями цільової змінної.

Для коректного застосування методу лінійної регресії необхідне виконання ряду статистичних припущень:

- лінійність – передбачається, що зв’язок між незалежними змінними та цільовою змінною має лінійний характер;
- незалежність похибок – помилки моделі повинні бути незалежними одна від одної;
- дисперсія похибок вважається сталою для будь-яких значень незалежних змінних;
- нормальність розподілу похибок – помилки повинні підкорятися нормальному закону розподілу.

Попри широку застосовність і універсальний характер, лінійна регресія має низку суттєвих обмежень, які можуть впливати на точність прогнозів і коректність інтерпретації моделі. Усвідомлення цих обмежень є важливим для адекватної оцінки результатів і вибору відповідних методів аналізу залежно від специфіки задачі. Основні недоліки лінійної регресії:

1. Припущення лінійності. Лінійна регресія виходить з того, що між незалежними змінними та цільовою змінною існує лінійна залежність. У реальних даних такі відношення трапляються зрідка, особливо у випадках із великою кількістю факторів та складними взаємодіями. За нелінійного характеру зв’язків оцінки параметрів стають зміщеними, а точність моделі – низькою. Для таких задач доцільніше застосовувати складніші методи, зокрема поліноміальну регресію, метод опорних векторів або нейронні мережі.

2. Чутливість до викидів. Лінійна регресія сильно реагує на аномальні спостереження, оскільки мінімізує суму квадратів відхилень. Викиди, що суттєво відхиляються від загальної тенденції, здатні значно змістити параметри моделі, викривлюючи регресійну пряму. Для зменшення цього впливу застосовують попереднє очищення даних або стійкіші методи, наприклад регресія на основі медіани (метод найменших модулів відхилень).

3. Мультиколінеарність. Якщо незалежні змінні сильно корелюють між собою, виникає проблема мультиколінеарності. Вона зумовлює нестабільність оцінки коефіцієнтів, ускладнюючи визначення внеску окремих факторів у результат. У таких умовах навіть невеликі зміни у вхідних даних можуть призвести до суттєвих коливань параметрів моделі. Для подолання цієї проблеми застосовують виключення надлишкових ознак або спеціальні методи (Lasso, Ridge), які накладають штрафи на коефіцієнти і тим самим зменшують вплив мультиколінеарності.

Таким чином, лінійна регресія не завжди є оптимальним інструментом для аналізу даних, особливо у випадках складних, масштабних і неоднорідних вибірок.

Нелінійні регресійні моделі поділяються на два основні класи:

- моделі, нелінійні щодо включених до аналізу факторів, але лінійні за оцінюваними параметрами. У таких випадках залежність між змінними описується нелінійною функцією, проте параметри входять до моделі лінійно, що дозволяє застосовувати

метод найменших квадратів. У табл. 5.10 наведено перелік деяких нелінійних регресійних залежностей, розрахунок параметрів яких ведеться як для лінійної регресії. До таких регресій належить також поліноміальна регресія загального вигляду:

$$\hat{y} = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_k x^k + \varepsilon;$$

Таблиця 5.10 – Нелінійні регресійні залежності, що зводяться до лінійних

Нелінійна регресія	Лінійний вигляд
$y = e^{a+b\cdot x}$	$\ln(y) = a + b \cdot x$
$y = \frac{1}{a + b \cdot x}$	$\frac{1}{y} = a + b \cdot x$
$y = a + \frac{b}{x}$	$y = a + b \cdot \frac{1}{x}$
$y = \frac{1}{a + \frac{b}{x}}$	$\frac{1}{y} = a + b \cdot \frac{1}{x}$
$y = a + b \cdot \ln(x)$	$y = a + b \cdot (\ln(x))$
$y = a \cdot x^b$	$\ln(y) = \ln(a) + b \cdot \ln(x)$
$y = a + b \cdot x^2$	$y = a + b \cdot (x^2)$
$y = (a + b \cdot x)^2$	$\sqrt{y} = a + b \cdot x$
$y = e^{a+\frac{b}{x}}$	$\ln(y) = a + b \cdot \frac{1}{x}$

- моделі, нелінійні за оцінюваними параметрами. Тут параметри входять у функцію в нелінійній формі, тому оцінювання вимагає використання ітераційних чисельних методів, таких як градієнтні алгоритми чи метод Ньютона.

Контрольні запитання

1. Які випадкові величини використовуються під час імітаційного моделювання?

2. Дайте характеристику багатовимірним випадковим величинам.

3. Що таке таблиця розподілу випадкової величини?

4. Як виглядає графічне подання закону розподілу дискретної випадкової величини?

5. Назвіть основні характеристики положення, розсіювання, зв’язку та закону розподілу випадкових величин.

6. Якими властивостями обладує математичне сподівання?

7. Якими властивостями обладує дисперсія?

8. Що таке коефіцієнт кореляції?

9. Які вимоги висуваються до оцінок параметрів генеральної сукупності?

10. Для чого використовується коваріаційна функція?

11. Для чого використовують однофакторний дисперсійний аналіз?

12. У чому полягає правило додавання дисперсій?

13. Що таке статистичний критерій?

14. Чим відрізняються між собою помилки I та II роду?

15. Поясніть сутність F-критерію Фішера.

16. Поясніть сутність критерію Краскела–Уолліса.

17. Для чого використовується критерій Джонкхієра?

18. В яких випадках використовують М-критерій Бартлетта, G-критерій Кокрена, критерій Левене, критерій Брауна–Форсайта?

19. Коли застосовується медіана Ходжеса–Лемана?

20. Опишіть сутність методу множинних порівнянь Шеффе.

21. Коли використовують двофакторний дисперсійний аналіз?

22. Як виглядає правило додавання дисперсій при двофакторному дисперсійному аналізі?

23. У чому полягає сутність кореляційного аналізу?

24. Що таке кореляційне поле?

25. Які коефіцієнти кореляції і в яких випадках розраховуються?

26. Що демонструє квартет Анскомбе?

27. Які основні задачі вирішуються за допомогою регресійного аналізу?

28. Які методи для отримання значення результативного показника використовують при регресійному аналізі?

29. Чим відрізняється лінійна регресія від нелінійної?

6. МОДЕЛЮВАННЯ В GPSS WORLD

Мова програмування – це сукупність (система) символів, що розпізнаються ЕОМ та позначають операції, які можливо реалізувати на ЕОМ. Виділяють такі типи мов програмування:

- машинно-орієнтовані;
- проблемно- (процедурно-) орієнтовані;
- об’єктно-орієнтовані мови.

Процедурно-орієнтовані мови не прив’язані до конкретної ЕОМ і призначені для певного класу задач. Вони містять інструкції, зручні для формулювання способів розв’язання типових задач цього класу.

Розвиток імітаційного моделювання розпочався в середині 60-х рр. XX ст. з використанням універсальних мов програмування та статистичних випробувань (метод Монте-Карло). Подальший розвиток обумовлено переходом до нової методології розробки моделей створення підпрограм, які реалізують виконання часто використовуваних операцій і дозволяють скоротити час розробки моделей.

У 1961 р. з’явилася спеціалізована мова імітаційного моделювання GPSS. Подальший розвиток був пов’язаний із розробкою спеціалізованих мов моделювання дискретно-подійних (SIMSCRIPT, Q-GERT, SIMULA, SLAM, GASP IV, SIMAN) та безперервних (CSMP III, CSSL III, CSSL IV, ASCL, DARE-P) моделей. Основним недоліком використання таких інструментів є необхідність знання спеціалізованої мови.

На початку 90-х років XX ст. були розроблені інтегровані засоби імітаційного моделювання для ЕОМ, наприклад, Сімула 67 МВК «Ельбрус».

У середині 90-х років XX ст. почався розвиток універсальних та спеціалізованих середовищ імітаційного моделювання, зумовлений швидким розвитком персональних ЕОМ. Нове покоління середовищ імітаційного моделювання (Arena, Extend та ін.) дозволило розробляти та проводити імітаційні експерименти без знання спеціалізованих мов. До кінця 90-х років

XX ст. були розроблені спеціалізовані середовища імітаційного моделювання, що відрізняються від універсальних середовищ наявністю бібліотек та шаблонів елементів виробничих систем. До 2000 р. вже було створено понад 50 середовищ імітаційного моделювання.

Класичні мови моделювання є процедурно-орієнтованими і мають низку специфічних рис. Можна сказати, що основні мови моделювання створені як засіб програмного забезпечення імітаційного підходу до дослідження складних систем.

Мови моделювання дозволяють описувати модельовані системи в термінах, заснованих на ключових поняттях імітації. З їх допомогою можна організувати процес комунікації між замовником і розробником моделі. Існують мови моделювання для неперервних і дискретних процесів.

На сьогодні склалась ситуація, коли не варто протиставляти мови загального призначення і мови імітаційного моделювання, деякі з них базуються на конструкціях мов загального призначення: наприклад, FORSIM – на мові FORTRAN, ПЛІС – на мові PL тощо.

Переваги мов імітаційного моделювання порівняно з універсальними мовами:

- мова моделювання містить абстрактні конструкції, які безпосередньо відображають поняття, в яких формалізовано модель, і близькі до концептуального рівня опису моделі. Це спрощує програмування, дозволяє автоматизувати виявлення багатьох помилок;
- ефективний вбудований механізм просування модельного часу (календар подій, методи інтегрування тощо), засоби для вирішення часових вузлів;
- вбудовані генератори випадкових чисел і типових впливів;
- автоматизоване збирання статистики та результатів моделювання з можливістю їх представлення в табличній чи графічній формі;

- зручності для проведення імітаційних експериментів (встановлення початкових умов, перезапуск тощо).

Недоліки мов імітаційного моделювання:

- обмеженість стандартними формами виводу результатів;
- невелика поширеність мов моделювання – зазвичай не входять до стандартного пакету операційної системи;
- потреба в додатковому навчанні, брак фахівців, які добре володіють цими мовами;
- відсутність гнучкості і широких можливостей, властивих універсальним мовам.

Архітектуру мови імітаційного моделювання – тобто концепцію взаємозв'язку елементів мови як складної системи – та технологію переходу від системи до її машинної моделі можна подати так:

- об'єкти моделювання описуються мовними атрибутами;
- атрибути взаємодіють із процесами, які відповідають реальним явищам у системі;
- процеси залежать від умов, що визначають логіку і послідовність взаємодії в часі;
- умови впливають на події, що відбуваються як усередині системи, так і при її взаємодії з середовищем;
- події змінюють стан моделі системи в просторі і часі.

Мови моделювання оцінюються за такими критеріями:

- зручність опису функціонування;
- зручність введення даних, зміни структури, алгоритмів і параметрів;
- ефективність аналізу та виведення результатів;
- простота налагодження і контролю;
- доступність для сприйняття та використання.

Більшість мов моделювання описують поведінку систем у часі через модифікований подієвий алгоритм (список поточних і майбутніх подій).

Класифікація мов моделювання, залежно від систем, для яких вони використовуються:

1. Неперервні системи – описуються диференціальними рівняннями, які встановлюють зв'язок між вхідними і вихідними функціями. Якщо вихідні змінні дискретні – використовуються різницеві рівняння.

2. Комбіновані системи – у системі можливі події двох типів:

- події, що залежать від стану;
- події, що залежать від часу.

Стан описується змінними, деякі з яких змінюються неперервно. Користувач описує умови настання подій, закони зміни змінних, правила переходу між станами. Наприклад, так працює GASP на основі FORTRAN.

3. Дискретні системи – використовуються в імітаційному моделюванні з різними підходами до опису динаміки:

- на транзактах. Мова створена для моделювання складних дискретних систем, де об'єкти (транзакти) – це, наприклад, заявки в системі обслуговування (GPSS);
- на процесах. Процес – це послідовність подій, логічно пов'язаних. Опис оформлюється як процедура, що одночасно виконується для всіх об'єктів певного класу (SIMULA, ПЛІС);
- на подіях. Створюється список (календар) подій. Час просувається за подіями. Програма складається з процедур обслуговування подій (SIMSCRIPT).

Зараз найбільш популярними пакетами імітаційного моделювання є: Arena, AnyLogic, GPSS World, Process Charter, Powersim, Ithlink, Extend+BPR, Vensim.

Програмні пакети імітаційного моделювання вирізняються підходами до побудови моделей. Process Charter реалізує моделювання у вигляді блок-

схем, тоді як Powersim і Ithink використовують нотацію системної динаміки. У Extend моделі формуються за допомогою комбінації блоків. Усі зазначені продукти, за винятком Process Charter, підтримують аналіз чутливості – можливість багаторазового запуску моделі з різними вхідними параметрами для порівняння результатів.

Серед відомих засобів імітаційного моделювання – Rockwell Arena, що базується на процесорі та мові моделювання SIMAN. Основні сфери застосування цього пакету охоплюють виробничі процеси, логістику, складування, безпеку, військову справу та медицину. Arena має інтуїтивно зрозумілий графічний інтерфейс з бібліотекою шаблонів.

Іншим потужним інструментом є AnyLogic – програмне забезпечення для моделювання складних систем і процесів. Воно підтримує ієрархічну побудову моделей, створення та організацію модулів у бібліотеки. AnyLogic розроблений на базі мови Java і функціонує на платформі Eclipse, яка є сучасним стандартом для бізнес-застосунків.

Пакет MatLab SimuLink for Windows призначений для моделювання з використанням графічних блоків, кожен з яких має власні параметри. До моделей можуть входити різні джерела сигналів, віртуальні прилади та засоби візуалізації результатів математичного моделювання в зручній формі.

6.1. Основи побудови та принципи функціонування мови імітаційного моделювання GPSS

GPSS World – це універсальна система моделювання для ОС Windows, яка використовує мову GPSS (General Purpose Simulation System). Програмне середовище підтримує створення моделей, інтерпретацію результатів, мультимедіа, відео та об’єктно-орієнтоване програмування.

Основна ідея мови GPSS полягає в тому, що складну систему можна представити як набір елементів, які взаємодіють між собою за визначеними логічними правилами. При цьому передбачається, що в межах моделі

можливо виокремити обмежену кількість абстрактних об’єктів, які імітують ключові компоненти системи.

Логічні правила, що визначають поведінку об’єктів, є типовими і описуються стандартним набором операцій. Сукупність програмних модулів, які моделюють функції об’єктів і реалізують ці логічні операції, формує програмну модель досліджуваної системи.

У галузі програмного забезпечення лише незначна кількість мов програмування змогла зберегти свою актуальність протягом тривалого часу. Довговічність GPSS пояснюється низкою причин:

- вона проста у вивченні та використанні;
- багато користувачів досягли значних успіхів у розв’язанні реальних задач саме завдяки мові GPSS;
- найважливіші класи об’єктів (заявки (транзакти), канали, накопичувачі, логічні перемикачі тощо) та їх властивості широко застосовуються в реальних обчислювальних мережах, виробничих і комерційних системах тощо;
- мова має широкий спектр застосування;
- постійно вдосконалюється;
- легко розширюється при побудові складніших моделей;
- доступне широке використання анімації;
- користувачі можуть без труднощів зрозуміти внутрішню логіку та алгоритми GPSS;
- інтерфейс простий і зручний;
- під час побудови моделі мова дозволяє оперувати безпосередньо поняттями, характерними для імітованої системи.

GPSS World – це потужне універсальне середовище для моделювання як дискретних, так і неперервних процесів, призначене для професійного моделювання найрізноманітніших процесів та систем [6].

У GPSS World підтримуються додаткові можливості, серед яких:

- динамічні графічні вікна для всіх класів об’єктів і змінних, де в реальному часі відображається проміжна та підсумкова статистика;
- гнучка процедурна мова PLUS (Programming Language Under Simulation), яка може використовуватися як для побудови моделей, так і для виконання експериментів;
- засоби підтримки факторного аналізу, традиційного дисперсійного аналізу (ANOVA) та регресійного аналізу, а також оптимізація на основі методології оптимального планування експерименту;
- додано підтримку елементів неперервного моделювання;
- усунуто проблеми з цілочисельним модельним часом.

За допомогою цієї системи можна ефективно моделювати як виробничі, так і невиробничі процеси: роботу торговельних і розважальних закладів, портів, організацію дорожнього руху, ведення бойових дій, діяльність редакцій, установ, мережі Internet, різних систем масового обслуговування тощо.

Система має широкий набір команд для управління процесом моделювання, які можна використовувати як в інтерактивному режимі, так і включати безпосередньо в модель. Забезпечено можливість проведення як експериментів, згенерованих самою системою, так і користувацьких та оптимізаційних екмпериментів.

У складі системи GPSS World є також спеціальна програма-планувальник, яка виконує такі функції:

- забезпечення просування за заданими розробником маршрутами динамічних об’єктів (транзактів);
- планування подій, які відбуваються в моделі, шляхом реєстрації часу появи кожної події і виконання їх у наростаючій часовій послідовності;

- реєстрація статистичної інформації про функціонування моделі;
- просування модельного часу в процесі моделювання системи.

Для забезпечення правильної послідовності обробки подій у часі призначений системний таймер, який зберігає значення абсолютного модельного часу. Усі проміжки часу описують додатними числами.

У GPSS World реалізовано процедури візуалізації функціонування моделі з використанням методів анімації.

Застосування GPSS World не лише суттєво прискорює процес моделювання та дослідження найрізноманітніших систем масового обслуговування і неперервних процесів, але й дозволяє проводити оптимізаційні експерименти.

Однак ця система має кілька недоліків. Дискретне моделювання основане на трьох основних парадигмах: сканування активностей (activity scanning), планування подій (event scheduling) і взаємодія процесів (process interaction). GPSS була однією з перших систем, яка запровадила парадигму взаємодії процесів. У цій формалізації транзакт (умовна одиниця активності) переміщується через блок-схему, яка репрезентує систему. Рух транзактів між різними блоками може спричиняти конкуренцію за ресурси з іншими транзактами, які одночасно переміщуються системою. Це, у свою чергу, може викликати затримки в нормальному русі транзактів. Тому GPSS дозволяє моделювати лише один пристрій: якщо в моделі є два або більше паралельно працюючих об’єктів, то таку модель необхідно моделювати за допомогою кількох паралельно працюючих пристроїв. Ще серед недоліків можна вказати відсутність графічної інтерпретації даних, що позбавляє можливості візуального представлення моделі в реальності та ускладнює процес обробки інформації, а також складність представлення процесів обробки інформації у вигляді алгоритмів.

Об’єкти системи GPSS створюються в середовищі GPSS World і можуть зберігатися у файлах. Основними об’єктами є [11, 13, 14, 5]:

- модель – містить опис у вигляді послідовності операторів, а також набір налаштувань, закладок у тексті програми, повідомлень про помилки тощо. Програми мовою GPSS називають моделями;
- експеримент – створюється в результаті трансляції моделі; подальше виконання може здійснюватися поетапно за допомогою команд (як тих, що описані в моделі, так і введених користувачем вручну);
- звіт – створюється після (успішного) завершення виконання експерименту. Містить статистику як по моделі загалом, так і по окремих блоках;
- текст – містить частину опису моделі (може підключатися до тексту програми, використовуватись під час введення/виведення тощо).

Мова GPSS складається з трьох основних типів елементів:

- транзактів, які є динамічними елементами;
- блоків, що є статичними елементами;
- операторів, які слугують для задання початкових даних і керування процесом моделювання.

В алфавіт мови GPSS входять такі символи:

- A, B, ..., Z – великі літери латинського алфавіту,
- 1, 2, ..., 9, 0 – цифри,
- ? –знак запитання,
- @ – комерційне а (знак залишку від ділення),
- # – решітка (номер) – використовується як знак множення,
- $ – знак долара,
- ^ – «карет» - знак зведення у ступінь,
- + – плюс,
- - – мінус,

- _ – підкреслення,
- * – зірочка,
- / – коса риска (знак поділу),
- (– ліва дужка,
-) – права дужка,
- ‘ – апостроф,
- . – крапка,
- , – кома,
- – пробіл.

Під час моделювання між системою, описаною мовою GPSS (GPSS-системою), і модельованою реальною системою встановлюється певна логічна відповідність, яка дозволяє на основі даних, отриманих при аналізі GPSS-системи, робити певні висновки про реальну систему. Такі висновки називаються інтерпретацією GPSS-систем.

Правила інтерпретації GPSS-систем становлять семантику мови GPSS. На відміну від синтаксису, семантика є неформальною стороною мови, а отже, не може бути описана з математичною однозначністю. Як правило, семантику описують і вивчають на основі прикладів інтерпретації різних мовних конструкцій.

Разом із синтаксичним та семантичним аспектами мови важливе значення має її прагматичний аспект. Прагматика мови включає відомості про її практичне застосування, зокрема типові прийоми написання програм, методи організації їх налагодження та виконання тощо.

Мова GPSS реалізує парадигму взаємодії процесів (process interaction paradigm). Побудова моделі вимагає щонайменше трьох основних компонентів: сутностей (транзактів), процесів та операцій (блоків), які входять до складу цих процесів. Кожен блок виконує певну дію над транзактом, тим самим формуючи поведінку та динаміку моделі в часі.

Блоки мови GPSS – це підпрограми, написані на макроасемблері або мовою С, що містять набір параметрів (операндів) для взаємодії з ними. Як і в усіх мовах моделювання, у GPSS існує внутрішній механізм передавання управління, що реалізується в модельному часі. Це дає змогу відображати динамічні процеси в реальних системах.

Передача управління від блоку до блоку в програмах GPSS відбувається через рух транзактів у модельному часі. Звернення до підпрограм блоків відбувається саме через це переміщення транзактів.

Змістовне значення транзактів задає розробник моделі. З погляду програми, транзакт – це структура даних, яка містить такі поля: ім’я або номер транзакта; час його появи; поточний модельний час; номер блоку, в якому він зараз перебуває; номер наступного блоку, в який він має переміститися; момент часу початку переміщення; пріоритет транзакта; параметри транзакта.

У мові GPSS усі транзакти нумеруються в порядку їх появи в моделі. Параметри транзакта відображають властивості моделюваного динамічного об’єкта.

На початку моделювання в GPSS-моделі не існує жодного транзакта. У процесі моделювання транзакти входять у модель у певні моменти часу відповідно до логіки функціонування модельованої системи. Так само транзакти можуть залишати модель залежно від специфіки моделювання.

У загальному випадку в моделі може одночасно існувати кілька транзактів, але в кожен момент часу рухається лише один з них. Коли транзакт починає свій рух, він переміщується від блоку до блоку відповідно до блок-схеми (логіки роботи моделі). У той момент, коли транзакт входить у блок, викликається відповідна йому підпрограма. Далі транзакт (як правило) намагається перейти до наступного блоку. Його переміщення продовжується доти, поки не виконається одне з таких умов:

- транзакт входить у блок, функцією якого є затримка транзакта на певний час;

- транзакт входить у блок, функцією якого є видалення транзактів з моделі;
- транзакт намагається увійти до блоку, який з певних причин не може його прийняти.

Якщо відбулася одна з наведених умов, транзакт залишається на місці, а в модельному просторі починає рух інший транзакт. Таким чином, моделювання в системі продовжується далі згідно з логікою обробки інших транзактів.

Об’єкти в модельованій системі призначені для різних цілей. Вибір об’єктів у модельованій системі залежить від характеристик моделі, а в деяких випадках – і від розробника моделі. Зовсім необов’язково, щоб у кожній моделі були задіяні всі типи об’єктів. Обов’язковою умовою є лише наявність блоків і транзактів, інакше модель не зможе працювати. Об’єкти поділяються на 7 категорій:

1. Динамічна – транзакти.

2. Операційна – блоки.

3. Апаратна (статична) – одноканальні пристрої, багатоканальні пристрої, логічні ключі.

4. Обчислювальна – змінні, функції, генератори випадкових чисел.

5. Статистична – черги, таблиці.

6. Запам’ятовуюча – комірки, матриці комірок.

7. Групувальна – числові групи, групи транзактів, списки.

Динамічними об’єктами є транзакти, які створюються в певних точках моделі, просуваються планувальником через блоки, а потім знищуються. Транзакти є аналогами одиниць виконання – ниток у реальній системі.

Об’єкти апаратної категорії – це абстрактні елементи, на які може бути декомпозована реальна система. Впливаючи на ці об’єкти, транзакти можуть змінювати їхній стан і впливати на рух інших транзактів. До об’єктів цього типу належать одноканальні і багатоканальні пристрої та логічні ключі.

Одноканальні пристрої – це обладнання, яке в будь-який момент часу може бути зайняте лише одним транзактом. Багатоканальні пристрої призначені для імітації обладнання, що здійснює паралельну обробку. Вони можуть одночасно використовуватись кількома транзактами. Події, які відбулися раніше в системі, можуть заблокувати, змінити рух транзактів і настання подальших подій. Для моделювання таких ситуацій використовуються логічні ключі. Транзакт може встановити ці ключі в стан «увімкнутий» або «вимкнутий». Інші транзакти можуть перевіряти стан ключів для визначення подальшого шляху свого проходження.

Операційні об’єкти (блоки) задають логіку функціонування моделі системи і визначають шляхи переміщення транзактів між об’єктами апаратної категорії. У блоках можуть відбуватись події чотирьох основних типів:

- створення або знищення транзактів;
- зміна числового атрибута об’єкта;
- затримка транзакта на певний період часу;
- зміна маршруту руху транзакта в моделі.

Версія GPSS, реалізована в системі GPSS World, містить 53 типи блоків [7]. Залежно від призначення блоки поділяються на кілька груп:

1. Блоки модифікації атрибутів транзактів (GENERATE, SPLIT, TERMINATE, ASSEMBLE, ADVANCE, MATCH, GATHER, ASSIGN, INDEX, MARK, PLUS, PRIORITY).

2. Блоки зміни послідовності руху (передачі управління) транзактів (DISPLACE, TRANSFER, LOOP, TEST, GATE).

3. Блоки, пов’язані з групувальною категорією (ADOPT, ALTER, EXAMINE, JOIN, REMOVE, SCAN).

4. Блоки опису об’єктів апаратної категорії:

- одноканальні пристрої (SEIZE, RELEASE, PREEMPT, RETURN, FUNAVAIL, FAVAIL);

- багатоканальні пристрої (ENTER, LEAVE, SAVAIL, SUNAVAIL);
- ключі (LOGIC).

5. Блоки збереження значень для їх подальшого використання (SAVEVALUE, MSAVEVALUE).

6. Блоки для забезпечення отримання статистичних результатів:

- черги (QUEUE, DEPART);
- таблиці (TABULATE).

7. Блоки для організації списку користувача (LINK, UNLINK).

8. Блоки для організації введення-виведення (OPEN, CLOSE, READ, WRITE, SEEK).

9. Спеціальні блоки (BUFFER, COUNT, EXECUTE, INTEGRATION, SELECT, TRACE, UNTRACE).

Перелік всіх об’єктів мови GPSS World наведено в таблиці 6.1.

Таблиця 6.1 – Об’єкти мови GPSS World

Блок	Призначення
1	2
ADOPT	Зміна приналежності транзакта до сімейства
ADVANCE	Затримка транзакта
ALTER	Модифікація параметрів або пріоритету групи
ASSEMBLE	Об’єднання транзактів
ASSIGN	Присвоєння параметрам транзакту значень
BUFFER	Розміщення транзакта в списку поточних подій
CLOSE	Закриття потоку даних (файла)
COUNT	Запис кількості об’єктів за умовою в параметр транзакта
DEPART	Зменшення довжини черги
DISPLACE	Відшукання транзакта і переміщення його до нового блока
ENTER	Займання багатоканального пристрою
EXAMINE	Перевірка членства транзакта в групі
EXECUTE	Дозвіл транзакту на виконання операції іншого блока
FAVAIL	Режим готовності одноканального пристрою
FUNAVAIL	Режим неготовності одноканального пристрою
GATE	Визначення стану пристроїв і логічних ключів
GATHER	Об’єднання транзактів у задану кількість
GENERATE	Введення транзактів у модель
INDEX	Зміна значення параметра транзакта
INTEGRATION	Перемикання процесу інтегрування
JOIN	Приєднання транзакта до групи

Закінчення табл. 6.1

1	2
LEAVE	Звільнення багатоканального пристрою
LINK	Введення транзактів до списку користувача
LOGIC	Зміна стану логічного ключа
LOOP	Організація в моделі циклів
MARK	Занесення до транзакта значення абсолютного модельного часу
MATCH	Синхронізація руху транзактів
MSAVEVALUE	Занесення значень матриці в комірки пам’яті
OPEN	Відкриття потоку даних (файла)
PLUS	Обчислення виразу та запис його в параметр транзакта
PREEMPT	Переривання обслуговування
PRIORITY	Присвоєння транзакту пріоритету
QUEUE	Збільшення довжини черги
READ	Читання рядка з потоку даних (файла)
RELEASE	Звільнення одноканального пристрою
REMOVE	Видалення транзакта з групи
RETURN	Поновлення обслуговування
SAVAIL	Режим готовності багатоканального пристрою
SAVEVALUE	Занесення інформації в комірку пам’яті
SCAN	Перевірка приналежності транзакта до групи
SEEK	Встановлення позиції поточного рядка в потоці даних
SEIZE	Займання одноканального пристрою
SELECT	Запис результату вибору об’єктів за умовою у параметр транзакту
SPLIT	Копіювання транзакта
SUNAVAIL	Режим неготовності багатоканального пристрою
TABULATE	Табулювання значення заданого аргументу
TERMINATE	Вилучення транзактів з моделі та завершення моделювання
TEST	Порівняння значень величин і управління напрямком руху транзакта
TRACE	Відслідковування руху транзактів
TRANSFER	Зміна траєкторії руху активного транзакта
UNLINK	Виведення транзактів зі списку користувача
UNTRACE	Припинення трасування руху транзактів
WRITE	Запис рядка в потік даних (файл)

Обчислювальна категорія призначена для опису ситуацій у процесі моделювання, коли зв’язки між компонентами моделюваної системи найпростіше та найкомпактніше виражаються у вигляді математичних (аналітичних і логічних) співвідношень. Для цього, як об’єкти обчислювальної категорії, введено арифметичні та булеві змінні і функції.

Змінні є складними виразами, що включають константи, системні числові атрибути, бібліотечні арифметичні функції, а також арифметичні та логічні операції.

Вирази можуть використовуватись у змінних і операторах GPSS. При використанні в змінних, вирази визначаються командами GPSS. При використанні в операторах GPSS вирази задаються як частина мови PLUS.

Кожному об’єкту відповідають атрибути, які описують його стан у даний момент часу. Вони доступні для використання протягом усього процесу моделювання і називаються системними числовими атрибутами. Загалом у GPSS World існує понад 50 системних числових атрибутів (табл. 6.2).

Таблиця 6.2 – Стандартні числові атрибути об’єктів GPSS World

Типи об’єктів	Назва	Призначення
1	2	3
Системні (атомарні)	C1	Поточне значення умовного часу. Автоматично змінюється в моделі і встановлюється в 0 операторами управління CLEAR або RESET.
	AC1	Поточне значення абсолютного часу. Автоматично змінюється в моделі. Встановлюється в 0 оператором CLEAR. Дійсне число.
	TG1	Поточне значення лічильника числа завершень. Ціле число.
	Z1	Розмір вільної оперативної пам’яті у байтах. Ціле число.
Транзакти	P	Значення параметра активного транзакта. Ціле, дійсне число або рядок.
	PR	Пріоритет активного транзакта. Ціле число.
	M1	Час перебування в моделі активного транзакта. Дорівнює різниці поточного значення абсолютного часу та часу створення активного транзакта. Дійсне число.
	MP	Транзитний час перебування в моделі активного транзакта. Дорівнює різниці поточного значення абсолютного часу та вмісту параметра активного транзакта. Дійсне число.
	XN1	Номер активного транзакта. Ціле число.
Блоки	N	Загальне число транзактів, які ввійшли в блок. Ціле число.
	W	Поточне число транзактів у блоці. Ціле число.
Одноканальні пристрої	F	Стан пристрою. Дорівнює 0, якщо пристрій вільний, і 1 – в інших випадках. Ціле число.
	FI	Прапорець переривання пристрою: 1 – пристрій знаходиться в стані переривання, 0 – інакше. Ціле число.
	FV	Прапорець готовності пристрою до використання: 1 – готовий, 0 – інакше. Ціле число.
	FR	Коефіцієнт використання пристрою в частках тисячі. Дійсне число.
	FC	Загальне число зайнять пристрою. Ціле число.
	FT	Середній час використання пристрою одним зайняттям. Дійсне число.

Продовження табл. 6.2

1	2	3
Багатоканальні пристрої	S	Поточний вміст каналів пристрою. Ціле число.
	R	Число вільних одиниць каналів пристрою. Ціле число.
	SR	Коефіцієнт використання каналів у частках тисячі. Дійсне число.
	SA	Зважений за часом середній вміст каналів. Дійсне число.
	SM	Максимальна кількість каналів. Ціле число.
	SC	Загальна кількість каналів, що використовувалися. Ціле число.
	ST	Середній час перебування транзактів у багатоканальному пристрої. Дійсне число.
	SE	Прапорець незайнятості пристрою (1 – вільний, 0 – зайнятий). Ціле число.
	SF	Прапорець наповненості пристрою:(1 – наповнений, 0 – не наповнений. Ціле число.
	SV	Прапорець готовності пристрою (1 – готовий, 0 – не готовий). Ціле число.
Датчики випадкових чисел	RN	Число, що обчислюється датчиком випадкових чисел (може бути до 7 датчиків). Датчик генерує послідовність рівномірно розподілених цілих випадкових чисел в інтервалі 0–999. При використанні датчика в якості аргумента функції або об’єкта в змінній значення буде дробом від 0 до 0.999999.
Ключі	LS	Стан логічного ключа: (1 – встановлено, 0 – не встановлено). Ціле число.
Черги	Q	Поточна довжина черги. Ціле число.
	QA	Зважена за часом середня довжина черги. Дійсне число.
	QM	Максимальна довжина черги. Ціле число.
	QC	Загальне число входів у чергу. Ціле число.
	QZ	Число нульових входів у чергу. Ціле число.
	QT	Середній час перебування транзактів у черзі (разом з нульовими входами). Дійсне число.
	QX	Середній час перебування повідомлень у черзі (без нульових входів). Дійсне число.
Таблиці	TB	Середнє значення аргументів таблиці. Дійсне число.
	TC	Загальне число аргументів таблиці. Ціле число.
	TD	Обчислене середньоквадратичне відхилення для аргументів таблиці. Дійсне число.
Датчики випадкових чисел	RN	Число, що обчислюється датчиком випадкових чисел (може бути до 7 датчиків). Датчик генерує послідовність рівномірно розподілених цілих випадкових чисел в інтервалі 0–999. При використанні датчика в якості аргумента функції або об’єкта в змінній значення буде дробом від 0 до 0.999999.
Функції	FN	Обчислене значення функції. Дійсне число.
Змінні	V	Обчислене значення цілої змінної або змінної з плаваючою крапкою. Дійсне число.
Булеві змінні	BV	Обчислене значення булевої змінної. Дійсне число.
Збережені величини	X	Значення збереженої величини. Ціле, дійсне число або рядок.

Закінчення табл. 6.2

1	2	3
Матриці збережених величин	MX(a,b)	Зміст елемента матриці збережених величин, розташованого в рядку a, стовпці b. Ціле, дійсне число або рядок.
Числові групи	GN	Поточне число членів у числовій групі. Ціле число.
Групи транзактів	GT	Поточне число членів у групі транзактів. Ціле число.
Сімейства транзактів	A1	Номер сімейства активного транзакта. Ціле число.
	MB	Флаг синхронізації: 1 – транзакт у деякому блоці належить тому ж сімейству, що і активний транзакт; 0 – інакше. Ціле число.
Списки користувача	CH	Поточне число транзактів у списку користувача. Ціле число.
	CA	Зважене за часом середнє число транзактів у списку користувача. Дійсне число.
	CM	Максимальне число транзактів у списку користувача. Ціле число.
	CC	Загальне число входів транзактів у список користувача. Ціле число.
	CT	Середній час перебування транзактів у списку користувача. Дійсне число.

Булеві змінні дозволяють користувачеві перевіряти в одному блоці GPSS одночасно кілька умов, виходячи зі стану або значення цих умов та їхніх атрибутів. У таких блоках відбувається звернення до булевої змінної, вираз якої містить перевірку кількох умов. Булеві змінні можуть бути представлені комбінаціями стандартних числових атрибутів, пов’язаних між собою булевими операторами, включаючи і інші змінні. Булеві змінні визначаються так само, як і арифметичні, але замість арифметичних операцій перевіряються різні логічні умови.

За допомогою функцій користувач може здійснювати обчислення неперервних або дискретних функціональних залежностей між аргументом функції (незалежною змінною) та залежним значенням функції. Усі функції в GPSS задаються табличним способом за допомогою команд опису функцій. Так само як і змінні, функції не пов’язані з конкретними блоками.

Окрім бібліотечних арифметичних функцій, GPSS World має 24 вбудовані генератори випадкових чисел (табл. 6.3) з різними потоками ймовірнісних розподілів. Параметр RNj – номер генератора (можна

використовувати в моделі до 7 генераторів), інші параметри залежать від використовуваного розподілу [7]: min, max – найменше і найбільше значення отриманої випадкової величини; m – зсув; s – масштабний параметр; a і b – параметри форми; t – кількість незалежних випробувань; p – ймовірність успішних випробувань; c – кількість успішних випробувань, після якої повертається кількість неуспішних випробувань; v – середнє значення розподілу.

Таблиця 6.3 – Генератори випадкових чисел у GPSS World

Функція	Розподіл	Тип результату
Beta(RNj,min,max,a,b)	Бета	Дійсний
Binomial(RNj,t,p)	Біноміальний	Цілий
DUniform(RNj,min,max)	Дискретнорівномірний	Цілий
Exponential(RNj,m,s)	Показниковий	Дійсний
ExtValA(RNj,m,s)	Екстремального значення типу A	Дійсний
ExtValB(RNj,m,s)	Екстремального значення типу B	Дійсний
Geometric(RNj,p)	Геометричний	Цілий
Gamma(RNj,m,s,a)	Ґамма	Дійсний
InvGauss(RNj,m,s,a)	Обернений Ґаусса	Дійсний
InvWeibull(RNj,m,s,a)	Обернений Вейбулла	Дійсний
Laplace(RNj,m,s)	Лапласа	Дійсний
Logistic(RNj,m,s)	Логістичний	Дійсний
LogLaplace(RNj,m,s,a)	Логлапласів	Дійсний
LogLogis(RNj,m,s,a)	Логлогістичний	Дійсний
LogNormal(RNj,m,s,a)	Логнормальний	Дійсний
NegBinom(RNj,c,p)	Від’ємний біноміальний	Цілий
Normal(RNj,m,s)	Нормальний	Дійсний
Pareto(RNj,m,s)	Парето	Дійсний
Pearson5(RNj,m,s,a)	Пірсона типу 5	Дійсний
Pearson6(RNj,m,s,a,b)	Пірсона типу 6	Дійсний
Poisson(RNj,v)	Пуассона	Цілий
Triangular(RNj,min,max,mode)	Трикутний	Дійсний
Uniform(RNj,a,b)	Рівномірний	Дійсний
Weibull(RNj,m,s,a)	Вейбулла	Дійсний

Вбудовані в GPSS World генератори випадкових чисел засновані на мультиплікативно-конгруентному алгоритмі Лемера з максимальним періодом. Алгоритм генерує псевдовипадкові числа у відкритому інтервалі від 0 до 2147483647, і до моменту самоповторення може згенерувати $2^{31}-2=2147483646$ унікальних псевдовипадкових чисел. Додатково в GPSS

World використовується крок перемішування. Однак експерименти, виконані в GPSS World, виявили недоліки: наприклад, послідовності псевдовипадкових чисел, які генеруються процедурами Uniform(1,0,1), Uniform(2,0,1), Uniform(3,0,1) і Uniform(4,0,1) мають довжину періоду 2047999999 з числами в діапазоні значень від 0.000000 до 0.999999. Тому знання довжини періоду використовуваних генераторів випадкових чисел необхідне для коректної організації довгих прогонів імітаційних моделей.

Об’єкти запам’ятовуючої категорії забезпечують доступ до збережених значень. Комірки збережених величин і матриці таких комірок використовуються для зберігання числової інформації. Наприклад, значення, записане в комірку, може відображати коефіцієнт використання пристрою в певний момент часу. Будь-який активний транзакт може здійснити запис та читання інформації в цих об’єктах. Матриці можуть мати до шести вимірів.

До статистичних об’єктів належать черги та таблиці.

У будь-якій системі рух потоку транзактів може затримуватись через недоступність пристроїв (одноканальний пристрій зайнятий або багатоканальний пристрій переповнений). У такому випадку затримані транзакти ставляться в чергу. Облік черг є однією з основних функцій планувальника. Він автоматично накопичує статистику щодо пристроїв і черг. Окрім цього, користувач може самостійно збирати додаткову статистичну інформацію, вказуючи спеціальні точки в моделі.

Для спрощення обробки статистичної інформації в GPSS передбачено таблиці. Таблиці використовуються для отримання вибіркових розподілів випадкових величин. Таблиці складаються з частотних класів (діапазонів значень), в які заноситься кількість потраплянь конкретного числового атрибуту в той чи інший клас. Для кожної таблиці обчислюються також математичне сподівання і середньоквадратичне відхилення. Після завершення експерименту з моделлю результати, що містяться в таблицях, можуть бути виведені на дисплей або надруковані.

До групувальної категорії в GPSS належать числові групи, групи транзактів і списки (ланцюги подій).

У складних імітаційних моделях нерідко виникає потреба в роботі з множинами транзактів, які мають спільні характеристики або значення параметрів. Зазвичай такі параметри задаються як числові атрибути транзактів і можуть змінюватися під час проходження моделі, зокрема за допомогою блоку ASSIGN. Однак, такий підхід не завжди забезпечує гнучке управління – іноді необхідно одночасно змінити або проаналізувати значення параметрів усіх транзактів певної групи. Для цього в GPSS передбачено об’єкти групувальної категорії, що дозволяють працювати з множинами транзактів або числових значень. Ці об’єкти можуть функціонувати в одному з двох режимів:

- режим транзакта, де група містить номери транзактів;
- числовий режим, де група зберігає набір числових значень.

Режим встановлюється при першому зверненні до групи і має підтримуватися незмінним протягом усього часу роботи моделі.

Списки є важливим механізмом для управління станами транзактів у процесі моделювання. Залежно від ситуації, транзакт може перебувати лише в одному зі списків, що відображають його поточний стан або очікувану подію. GPSS використовує п’ять основних типів списків:

- список (ланцюг) поточних подій (Current Events Chain, CEC) – містить транзакти, для яких заплановані дії мають бути виконані в момент, що вже настав або дорівнює поточному модельному часу. Ці транзакти готові до подальшого просування по блоках моделі. У цьому списку транзакти впорядковуються за спаданням пріоритетів; при однакових пріоритетах враховується порядок надходження до списку;
- список (ланцюг) майбутніх подій (Future Events Chain, FEC) – містить транзакти, події для яких мають відбутися в

майбутньому. Усі елементи впорядковуються за часом виконання, незалежно від пріоритетів;

- список (ланцюг) повторних спроб (Retry Chain) – у цей список потрапляють транзакти, які не можуть перейти до наступного блоку через невиконання умов входу (наприклад, у блоках GATE, TEST, TRANSFER ALL, TRANSFER BOTH). Транзакти залишаються в цьому списку доти, доки відповідні системні змінні не зміняться. Після цього здійснюється повторна перевірка умов, і, у разі їх виконання, транзакт виходить зі списку та продовжує рух у моделі;
- список (ланцюг) затримки пристроїв (Delay Chain) – використовується для управління транзактами, які очікують доступу до одноканальних або багатоканальних пристроїв;
- користувацький список (User Chain) – дозволяє самостійно визначати умови збереження та вилучення транзактів зі списку.

Центральним завданням, яке виконує симулятор, є визначення того, який транзакт слід обрати наступним для просування в моделі після того, як його попередник припинив свій рух. З цією метою симулятор розглядає кожен транзакт як елемент певного списку. У відносно простих моделях зазвичай використовуються лише два основні списки: список поточних подій CEC і список майбутніх подій FEC.

Поточні активні дії, які мають відбутися негайно, реалізуються як транзакти, що зберігаються в списку поточних подій CEC. Транзакти, які мають бути активовані в майбутньому, розміщуються в списку майбутніх подій FEC. Впорядкування транзактів за пріоритетом у CEC дозволяє швидко визначати, які події мають бути виконані першими.

Різні події в реальних системах відбуваються протягом певного проміжку часу. Якщо ці події мають бути відображені в моделі, їх виникнення повинно відбуватися в межах модельного часу. За роботу з

модельним часом відповідає таймер, який автоматично обслуговується інтерпретатором GPSS.

На початку моделювання в інтерпретаторі планується прибуття першого транзакта. Після цього таймер встановлює модельний час у значення, яке відповідає моменту появи цього транзакта. Транзакт, разом з іншими, що прибувають у той самий момент, входить у модель і просувається через блоки, які визначені його логікою. Події, що виникають унаслідок проходження транзакта через блоки, плануються на наступні моменти часу. При цьому, у фіксований момент часу моделі виконується лише одна група подій, без накладання нових.

Далі інтерпретатор просуває таймер до найближчого моменту часу, на який заплановано наступну подію. Якщо в цей момент немає активних транзактів для обробки, таймер знову просувається вперед. Таким чином, модельний час у GPSS змінюється не поступово, а стрибкоподібно – від події до події.

Основні риси роботи таймера GPSS:

- цілочисельний формат часу. Таймер GPSS оперує лише цілими числами. Це означає, що події можуть відбуватись лише в цілі моменти часу, що дозволяє пришвидшити обчислення і зменшити використання оперативної пам’яті, оскільки цілочисельні операції виконуються швидше;
- гнучке трактування одиниці часу. Одиниця виміру часу, яку відзначає таймер, визначається розробником, але не передається явно інтерпретатору. Її вибір реалізується неформально – через значення параметрів, що вводяться до моделі. Якщо всі часи задаються в хвилинах – умовною одиницею часу буде хвилина; якщо в мілісекундах – мілісекунда. Таким чином, розробник сам обирає найзручніший масштаб часу для адекватного відображення динаміки реальної системи;

- GPSS реалізує підхід «наступної події». Тобто після того, як модель завершила обробку всіх подій у поточному моменті часу, таймер автоматично просувається до найближчого моменту, де запланована нова подія. Якщо між подіями існує відрізок часу без активності – цей інтервал пропускається, що дозволяє зосередити ресурси лише на значущих етапах моделювання.

Положення транзакта в списку подій тісно пов’язане з моментом його повторного входження в модель для продовження руху. Послідовність обробки визначається часом виникнення відповідних подій у процесі моделювання. Загалом цикл обробки подій у GPSS поділяється на три фази.

1. Фаза зміни модельного часу. Після завершення обробки всіх транзактів, які мають бути виконані в поточний момент модельного часу, планувальник (scheduler) переходить до визначення наступного значення модельного часу. Це значення встановлюється рівним часу настання найближчої події з майбутнього.

Оскільки в списку майбутніх подій FEC транзакти впорядковані за зростанням часу їхнього повернення в модель, нове значення модельного часу визначається як час події першого транзакта в FEC. Кожен транзакт у цьому списку асоційований із конкретним моментом часу, коли має відбутися відповідна подія.

Після встановлення нового значення модельного часу планувальник перевіряє, чи не існують інші події, заплановані на той самий момент. Для цього здійснюється послідовна перевірка наступних транзактів у FEC. Якщо час події для наступного транзакта також збігається з новим модельним часом, цей транзакт також переноситься до списку поточних подій CEC.

Цей процес повторюється до тих пір, поки:

- не буде знайдено транзакт, для якого час події перевищує поточне значення модельного часу;
- або поки FEC не спорожніє.

Після завершення перегляду FEC гарантується, що усі транзакти, події яких мають бути оброблені в поточний момент модельного часу, вже перенесені до CEC, і можуть бути далі оброблені відповідно до логіки моделі.

2. Фаза формування списку поточних подій. Після вибору всіх транзактів зі списку майбутніх подій FEC, які мають бути оброблені в поточний момент модельного часу, планувальник переходить до аналізу списку поточних подій CEC.

У цій фазі планувальник намагається просунути всі транзакти, які перебувають в активному стані, і тим самим змінити стан моделі відповідно до логіки блоків GPSS.

На початку обробки встановлюється прапорець зміни стану (flag of state change) у значення 0. Цей прапорець фіксує, чи відбулися зміни стану будь-яких пристроїв у поточний момент часу. Оскільки жодні дії ще не були виконані, на початку обробки значення прапорця дорівнює нулю.

Далі планувальник послідовно переглядає CEC, починаючи з першого транзакта. Для кожного транзакта перевіряється його індикатор перегляду, який вказує, чи перебуває транзакт у активному стані, чи в стані затримки.

Якщо транзакт активний, планувальник просуває його через блоки моделі, наскільки це дозволяє логіка, до першого блокуючого елемента (наприклад, до блока з ресурсними обмеженнями або затримкою в часі). Якщо ж транзакт перебуває в стані затримки – через блокування з боку стану ресурсу (тобто він розміщений у списку затримки), або якщо просування вже завершене – планувальник переходить до обробки наступного транзакта зі списку. Цей процес триває, доки не буде оброблено всі транзакти CEC, що належать до поточного значення модельного часу.

3. Фаза просування транзактів. Якщо транзакт, присутній у списку поточних подій CEC, перебуває в активному стані, планувальник намагається просунути його через блоки моделі. У разі, якщо просування є можливим,

транзакт переміщується до наступного блоку, і виконується операція, визначена цим блоком.

Після переміщення транзакта, планувальник перевіряє, чи є блок, до якого він потрапив, блоком типу BUFFER. Якщо так, обробка цього транзакта негайно припиняється, і планувальник повертається до перегляду CEC, переходячи до наступного транзакта. Якщо ж блок не є BUFFER, планувальник перевіряє, чи змінився стан пристрою під час виконання попереднього блоку. Якщо зміна зафіксована встановлюється прапорець зміни стану в значення 1 та скидаються в 0 індикатори перегляду всіх транзактів, що перебувають у списку затримки, пов’язаному з цим пристроєм.

Якщо блок не містить явно заданої затримки в часі, планувальник негайно просуває транзакт далі – у наступний блок. Цей процес триває доти, доки не буде виконано одну з наступних умов:

- транзакт не зможе увійти до наступного блоку через блокування,
- транзакт потрапить у блок ADVANCE з явною затримкою в часі,
- транзакт буде знищено в блоках TERMINATE або ASSEMBLE.

Якщо транзакт не може бути просунутий через стан пристрою (наприклад, ресурс зайнятий), то індикатор перегляду транзакта встановлюється в 1 (означає перехід у стан затримки) і транзакт переноситься у відповідний список затримки. Далі планувальник знову перевіряє, чи змінився стан пристрою внаслідок цієї обробки.

У разі, якщо просування зупинилося через часову затримку, транзакт переміщується в список майбутніх подій FEC, і також перевіряється зміна стану пристрою.

Якщо протягом обслуговування активного транзакта стан пристрою змінився, перегляд CEC починається заново – знову обробляються всі транзакти, що перебувають в активному стані.

Якщо ж стан пристроїв не змінився, планувальник продовжує роботу з СЕС, перевіряючи, чи залишилися в ньому транзакти, які потребують обробки.

Описаний цикл дій виконується для кожного нового моменту модельного часу Clock, доки лічильник завершення моделювання (Transaction Counter, TC) не дорівнюватиме нулю. Це ціле число, яке зменшується на значення, задане в параметрі блоку TERMINATE, і ініціалізується значенням, заданим у команді START.

На рис 6.1 наведено основний алгоритм обробки подій у GPSS.

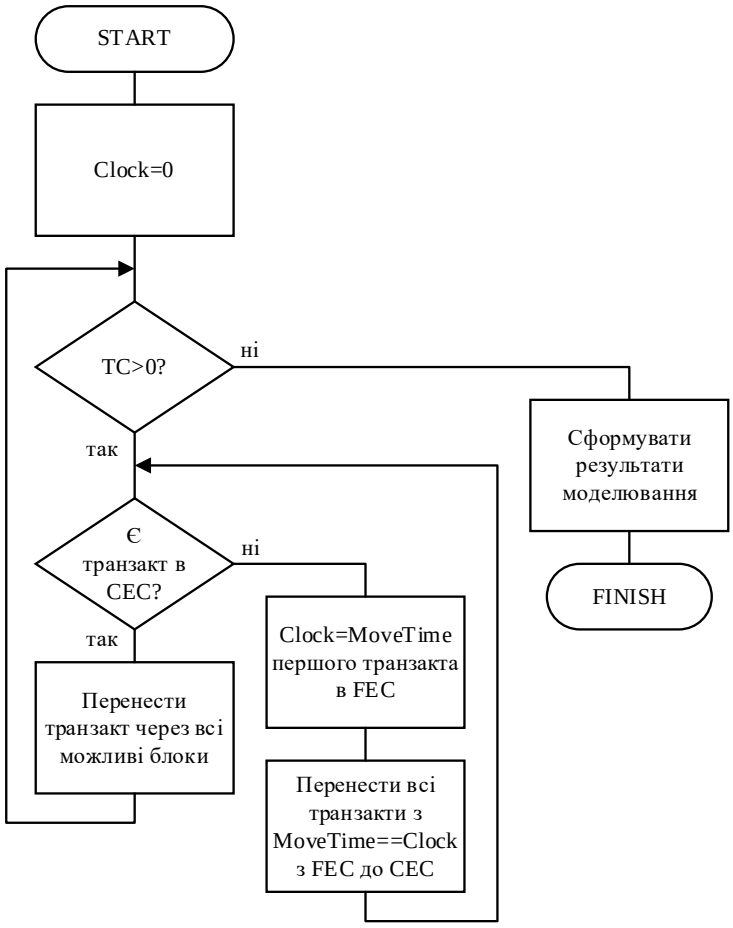


Рисунок 6.1 – Основний цикл роботи GPSS

Знання механізму формування ланцюгів подій може істотно полегшити процес налагодження і модифікації моделей.

Приклад процесу формування і обробки списків подій у моделі. У модельованій системі функціонують транзакти $i = 3$ типів, причому кожен

тип обробляється окремим каналом обслуговування. Для кожного типу транзактів заздалегідь відомі закон надходження (розподіл інтервалів між подіями надходження) і тривалість обслуговування (закон розподілу часу обробки транзакта). Таким чином, у системі існують i паралельних незалежних процесів, які можна позначити як P_i. Кожному з цих процесів відповідає послідовність подій $C_{i,j}$, де i – номер типу транзакта, j – порядковий номер події у відповідному процесі P_i. Часова діаграма функціонування моделі в умовах, коли по одному транзакту кожного типу обробляється одночасно наведена на рис. 6.2.

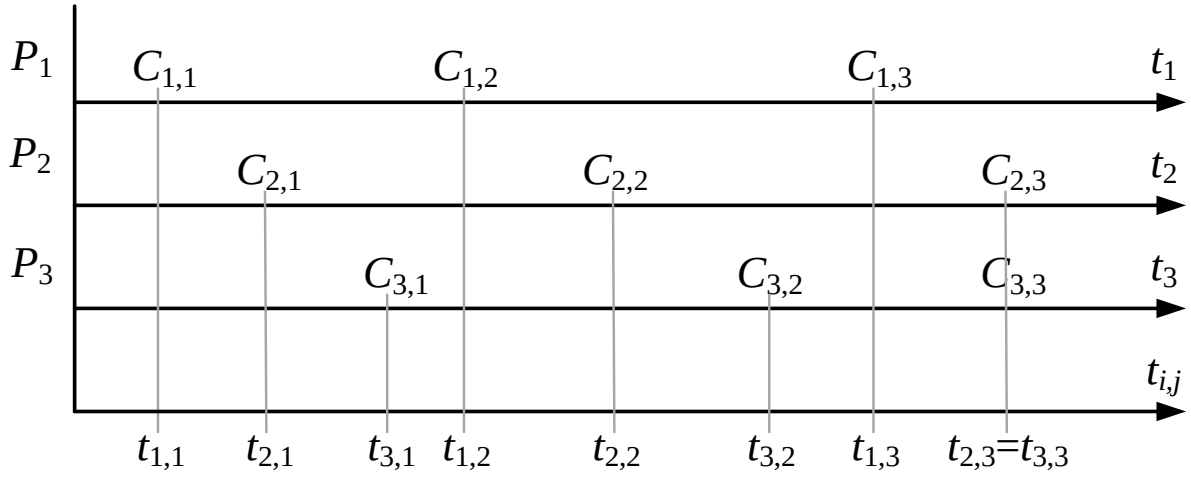


Рисунок 6.2 – Часова діаграма функціонування моделі

Для кожного процесу будується власний список подій, однак усі ці списки є спільними для всієї моделі, тобто розглядаються в рамках загальної подієвої структури. Формування починається зі створення списку майбутніх подій FEC (події, модельний час яких перевищує поточне значення модельного часу). Очевидно, що на момент формування списку майбутніх подій FEC час настання кожної з подій повинен бути відомий (визначений за заданими розподілами або алгоритмами генерації випадкових величин).

На першому кроці моделювання встановлюється модельний час $t_m = 0$ (початковий момент модельного часу). До списку майбутніх подій FEC вносяться перші події кожного процесу: $C_{1,1}$, $C_{2,1}$, $C_{3,1}$.

Далі, зі списку майбутніх подій FEC вибирається подія з найменшим часом настання ($C_{1,1}$), і переноситься в список поточних подій CEC. Якщо немає інших подій, які повинні настати одночасно з нею, ця подія обробляється, і після завершення обробки виключається зі списку поточних подій CEC.

Після цього знову оновлюється список майбутніх подій FEC: додаються нові події, час яких було визначено в результаті обробки попередніх, або коригуються вже наявні. Цей процес повторюється ітеративно, поки не буде вичерпано заздалегідь встановлений інтервал моделювання або не буде виконано умову зупинки.

Динаміку змін у списках майбутніх (FEC) і поточних (CEC) подій, що ілюструє описаний процес для розглянутого прикладу, наведено в табл. 6.4.

Таблиця 6.4 – Динаміка змін CEC і FEC для розглянутого прикладу

Час настання події, $t_{i,j}$	Список поточних подій CEC, $C_{i,j}$	Список майбутніх подій FEC, $C_{i,j}$
0		$C_{1,1}, C_{2,1}, C_{3,1}$
$t_{1,1}$	$C_{1,1}$	$C_{2,1}, C_{3,1}, C_{1,2}$
$t_{2,1}$	$C_{2,1}$	$C_{3,1}, C_{1,2}, C_{2,2}$
$t_{3,1}$	$C_{3,1}$	$C_{1,2}, C_{2,2}, C_{3,2}$
$t_{1,2}$	$C_{1,2}$	$C_{2,2}, C_{3,2}, C_{1,3}$
$t_{2,2}$	$C_{2,2}$	$C_{3,2}, C_{1,3}, C_{2,3}$
$t_{3,2}$	$C_{3,2}$	$C_{1,3}, C_{2,3}, C_{3,3}$
$t_{1,3}$	$C_{1,3}$	$C_{2,3}, C_{3,3}$
$t_{2,3}$	$C_{2,3}, C_{3,3}$	

Значення таймера доступні програмісту через системні стандартні числові атрибути C1 (відносний час) и AC1 (абсолютний час).

Простий приклад моделі обробника запитів у серверній системі:

```
GENERATE 300,20      ;Новий запит (транзакт) надходить раз на 300±20 мс
SEIZE SERVER         ;Запит захоплює ресурс обробки
ADVANCE 280,200      ;Запит обробляється протягом 280–480 мс
RELEASE SERVER       ;Ресурс звільняється, доступний для наступного запиту
TERMINATE 1          ;Запит виконано, видаляється з системи
```

Оскільки модельний час у GPSS є цілим числом, цілком можливо, що декілька подій можуть відбутися одночасно (як події $C_{2,3}$ та $C_{3,3}$ у прикладі вище). Причому, ймовірність такого збігу зростає зі збільшенням обраної одиниці модельного часу.

У деяких випадках одночасне настання кількох подій, або так званий часовий вузол, може серйозно порушити логіку моделі. Небезпека некоректної обробки часових вузлів є характерною для моделей, що використовують списки користувача.

Для правильної обробки часового вузла необхідно забезпечити такий порядок розміщення транзактів у списку поточних подій, за якого транзакт, що звільняє модельний ключовий ресурс, завжди розташовується першим. Цього можна досягти шляхом налаштувань пріоритетів транзактів.

Простий приклад моделі обробника запитів з пріоритетами в серверній системі:

```
GENERATE 300,20      ;Новий запит (транзакт) надходить раз на 300±20 мс
PRIORITY RN1         ;Призначення запиту випадкового пріоритету
SEIZE SERVER         ;Запит захоплює ресурс обробки
ADVANCE 280,200      ;Запит обробляється протягом 280–480 мс
RELEASE SERVER       ;Ресурс звільняється, доступний для наступного запиту
TERMINATE 1          ;Запит виконано, видаляється з системи
```

6.2. Методи збору статистики в імітаційній моделі

Більшість статистичних показників у середовищі GPSS збирається автоматично і не потребує додаткових зусиль з боку розробника моделі.

Наприклад, при моделюванні одноканальних і багатоканальних систем масового обслуговування (СМО) з очікуванням, моделі зазвичай будуються з метою аналізу характеристик, пов’язаних з очікуванням транзактів у черзі – таких як довжина черги, час очікування тощо. Проте, черга транзактів формується в ланцюгу поточних подій, і за замовчуванням недоступна для прямого спостереження з боку дослідника.

Для реєстрації статистичних даних про процес очікування транзактів у моделі слід використовувати спеціальні статистичні об’єкти, такі як черги (QUEUE) або таблиці (TABLE/QTABLE).

Збір статистики в GPSS здійснюється за допомогою:

- блоків QUEUE і DEPART – для обліку параметрів черги;
- об’єктів TABLE і QTABLE – для збору даних про розподіли випадкових величин.

Види статистики, які підтримуються в GPSS:

1. Статистика для одноканальних пристроїв:
   - кількість транзактів, що пройшли через пристрій;
   - середній час, протягом якого один транзакт займав пристрій;
   - завантаження пристрою (частка часу, коли пристрій був зайнятий);
   - відсоток часу, коли пристрій був доступним;
   - кількість транзактів, які захопили пристрій за допомогою блоку SEIZE;
   - кількість транзактів, які захопили пристрій за допомогою PREEMPT (примусове захоплення з витісненням).
2. Статистика для багатоканальних пристроїв (накопичувачів):
   - обсяг накопичувача (кількість каналів пристрою), заданий у моделі;
   - середнє значення використаного обсягу накопичувача протягом усього часу моделювання;
   - кількість транзактів, які проходили через накопичувач;
   - середній час використання однієї одиниці пам’яті;
   - середнє завантаження накопичувача;
   - відсоток доступності накопичувача;
   - поточне значення вмісту накопичувача на момент завершення моделювання;

- максимальне значення вмісту накопичувача за весь період моделювання.

3. Статистика для черг:

- максимальна довжина черги протягом моделювання;
- середня довжина черги;
- загальна кількість транзактів, що пройшли через чергу;
- кількість «нульових» транзактів – таких, що не зазнали затримки в черзі;
- відсоток «нульових» транзактів;
- середній час затримки транзакта в черзі з урахуванням «нульових»;
- середній час затримки транзакта в черзі без урахування «нульових» (це значення завжди більше або дорівнює попередньому, оскільки враховує лише ті транзакти, які реально очікували в черзі).

Об’єкти типу «черга» створюються в моделі за допомогою блоків-реєстраторів черг: QUEUE (стати в чергу) та DEPART (вийти з черги), які мають такий формат:

ім’я QUEUE A,B

ім’я DEPART A,B

У полі A вказується номер або ім’я черги, а в полі B – кількість одиниць, на які змінюється поточна довжина черги при вході транзакта в блок QUEUE (збільшується) або DEPART (зменшується). Зазвичай поле B залишають порожнім, і тоді за замовчуванням його значення дорівнює 1.

Для збору статистики про транзакти, що блокуються перед певним блоком моделі, блоки QUEUE і DEPART розміщуються перед та після цього блоку відповідно. Під час проходження транзактів через ці блоки змінюють такі стандартні числові атрибути (середньостатистичні характеристики) черг:

- Q – поточна довжина черги;

- QM – максимальна довжина черги;
- QA – ціла частина середньої довжини черги;
- QC – загальна кількість транзактів, що увійшли в чергу;
- QZ – кількість транзактів, які пройшли через чергу без очікування (так звані «нульові» входи);
- QT – ціла частина середнього часу очікування з урахуванням «нульових» входів;
- QX – ціла частина середнього часу очікування без урахування «нульових» входів.

Приклад моделі обробника запитів у серверній системі з накопиченням статистики:

```
GENERATE 300,20      ;Новий запит (транзакт) надходить раз на 300±20 мс
PRIORITY RN1         ;Призначення запиту випадкового пріоритету
QUEUE QUE            ;Запит стає в чергу QUE
SEIZE SERVER         ;Запит захоплює ресурс обробки
DEPART QUE           ;Запит залишає чергу – фіксується час перебування в черзі
ADVANCE 280,200      ;Запит обробляється протягом 280–480 мс
RELEASE SERVER       ;Ресурс звільняється, доступний для наступного запиту
TERMINATE 1          ;Запит виконано, видаляється з системи
```

Дослідників часто цікавить не лише середнє значення часу очікування в черзі, але і його дисперсія, а також статистичний розподіл вибірки часу очікування, який зазвичай подається графічно у вигляді гістограми. Такий розподіл дозволяє оцінити ймовірність того, що час очікування перевищить або не перевищить певне задане значення.

Для збору і обробки даних про вибірковий розподіл часу очікування в черзі використовуються Q-таблиці (таблиці типу QTABLE).

Щоб створити таку таблицю в моделі, її слід попередньо визначити за допомогою оператора QTABLE, який має такий формат:

ім’я QTABLE A,B,C,D

де ім’я – назва таблиці (для подальших посилань на неї), A – номер або ім’я черги, для якої потрібно отримати розподіл часу очікування, B – верхня межа

першого частотного інтервалу таблиці, C – ширина частотних інтервалів, D – кількість частотних інтервалів.

Діапазон усіх можливих значень часу очікування в черзі (з поля A) поділяється на інтервали частот, кількість яких задається в полі D. Перший інтервал охоплює значення від мінус нескінченності до включно B. Другий – значення, що перевищують B, але не більші за B+C і так далі. Усі проміжні інтервали мають однакову ширину, задану в полі C. Останній інтервал охоплює всі значення, що перевищують останню межу. Операнди B, C і D повинні бути цілими константами. Операнд B може бути непозитивним, хоча для Q-таблиці це не має сенсу (оскільки час не може бути від'ємним). Значення C і D мають бути строго додатними.

Під час проходження транзакта через блоки QUEUE і DEPART фіксується його час очікування, і до лічильника відповідного інтервалу таблиці додається одиниця. Одночасно накопичуються дані для обчислення середнього значення і середньоквадратичного відхилення (тобто кореня квадратного з дисперсії) часу очікування.

Після завершення моделювання ці значення – середнє, стандартне відхилення, а також кількість входів у кожен частотний інтервал – виводяться в стандартному звіті GPSS.

Як і інші об'єкти GPSS, таблиці мають стандартні числові атрибути:

- TC – загальна кількість транзактів, що увійшли в пов'язану з таблицею чергу;
- TB – ціла частина середнього часу очікування;
- TD – ціла частина середньоквадратичного відхилення часу очікування.

Приклад моделі обробника запитів у серверній системі з накопиченням статистики в таблиці:

```
WT QTABLE QUE,100,100,10    ;Таблиця збору статистики часу в черзі QUE
                            ;(10 інтервалів по 100 мс)
GENERATE 300,20             ;Новий запит надходить кожні 280–320 мс
PRIORITY RN1                ;Призначення запиту випадкового пріоритету
```

```
QUEUE QUE              ;Зайти в чергу та записати момент входу
SEIZE SERVER           ;Спроба захопити сервер
DEPART QUE             ;Вийти з черги та записати момент виходу
ADVANCE 280,200        ;Час обробки (280–480 мс)
RELEASE SERVER         ;Звільнення сервера
TERMINATE 1            ;Запит завершено
```

Для накопиченої статистики засобами GPSS World (Table Window) можна побудувати гістограму (рис. 6.3).

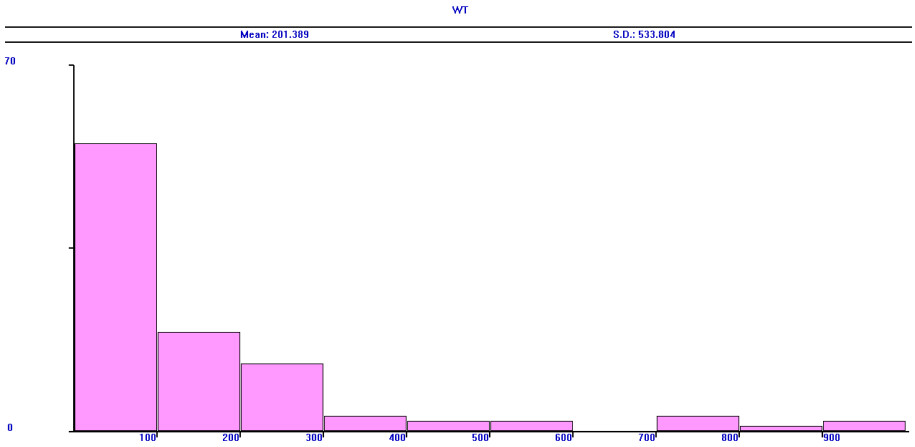


Рисунок 6.3 – Гістограма частот часу очікування запитів в черзі

Окрім QTABLE, у GPSS існує можливість створення таблиць для довільних статистичних характеристик об’єктів моделі. Для цього використовується оператор TABLE, формат якого аналогічний QTABLE:

ім’я TABLE A,B,C,D

Різниця полягає в тому, що в полі A записується стандартний числовий атрибут, вибірковий розподіл якого слід отримати, а операнди B, C і D визначають частотне розбиття діапазону його можливих значень.

Однак, на відміну від QTABLE, оновлення таких таблиць не виконується автоматично. Для цього використовується блок TABULATE, який має формат:

ім’я TABULATE A

У полі A зазначається номер або ім’я таблиці, яка була визначена оператором TABLE.

Коли транзакт входить у блок TABULATE, поточне значення аргумента (з оператора TABLE) записується до відповідного частотного інтервалу таблиці. Одночасно оновлюються лічильники таблиці: кількість входів TC, середнє значення TB та середньоквадратичне відхилення TD.

6.3. Блок-діаграми GPSS

У GPSS для подання модельованої системи S у вигляді машинної моделі використовується мова блок-діаграм. Блок-діаграмою в GPSS називають графічне відображення операцій, що відбуваються в моделюваній системі S. У цьому контексті блок-діаграма описує взаємодії, які відбуваються всередині системи S у процесі її функціонування.

Запроваджений набір блоків для побудови блок-діаграм однозначно визначає набір операторів мови, що забезпечують опис структури модельованої системи S, а також логічні правила, які регламентують її функціонування.

У блок-діаграмах GPSS блоки відповідають операціям, що виконуються над динамічними об’єктами, тоді як стрілки між блоками відображають маршрути переміщення цих об’єктів у системі. Альтернативні ситуації зображуються кількома стрілками, що виходять з одного блока. Блок-діаграма являє собою набір фігур із характерною формою блоків мови GPSS, з’єднаних між собою лініями. Кожен блок є підпрограмою, реалізованою засобами макроасемблера.

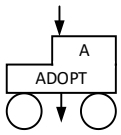
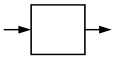
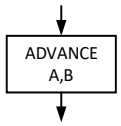
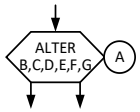
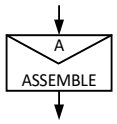
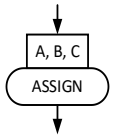
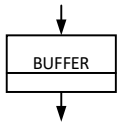
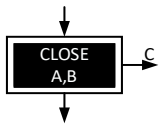
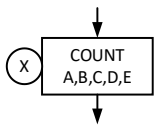
Таким чином, процес створення моделі мовою блок-діаграм GPSS зводиться до декомпозиції вихідної системи S до рівня елементарних операцій, передбачених пакетом GPSS, формування фіксованої схеми, яка відображає послідовність цих операцій над динамічними об’єктами, та визначення набору логіко-імовірнісних правил просування потоків об’єктів по заданій схемі.

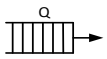
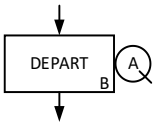
Побудова блок-діаграм GPSS вимагає від розробника знання набору операторів пакета GPSS, що безпосередньо відповідає набору блоків для

опису блок-діаграм. Побудова блок-діаграми становить проміжний етап у створенні імітаційної моделі досліджуваної системи S з використанням операторів пакета GPSS.

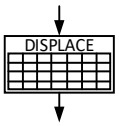
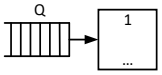
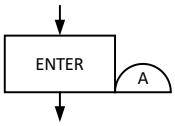
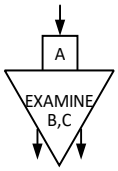
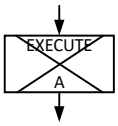
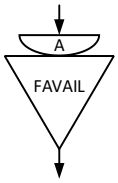
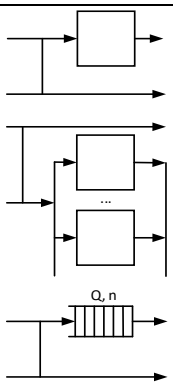
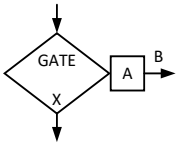
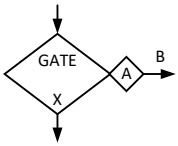
У табл. 6.5 наведено графічне зображення операторів мови GPSS та відповідних їм елементів Q-схем, якими описуються СМО.

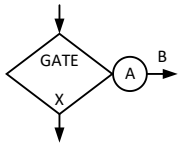
Таблиця 6.5 – Символи блок-діаграм GPSS

Оператор	Q-схема	Блок-діаграма	Опис
1	2	3	4
ADOPT A			Змінює приналежність транзакту з одного сімейства до нового сімейства A
ADVANCE A,B			Затримує транзакт на середній час A, B – модифікатор розкиду або модифікатор функції
ALTER X A,B,C,D,E,F,G			Змінює атрибути членів групи A
ASSEMBLE A			Об’єднує транзакти одного сімейства кількістю A
ASSIGN A,B,C			Присвоює параметру A транзакта значення B, модифіковане значенням C
BUFFER			Перериває обробку транзакта та повертає симулятор до початку списку поточних подій
CLOSE A,B,C			Закриває потік даних B, у номер або параметр транзакта A записується код помилки, у випадку помилки перехід на мітку C
COUNT X A,B,C,D,E			Визначає кількість об’єктів, що задовольняють умові X у діапазоні [B...C] і записує їх у параметр A транзакта

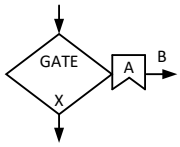
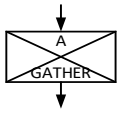
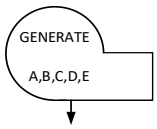
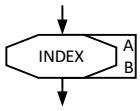
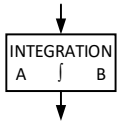
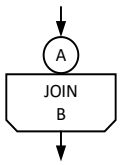
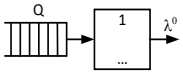
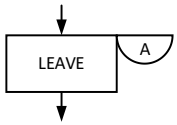
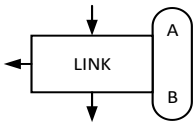
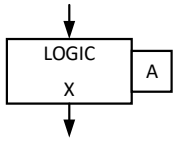
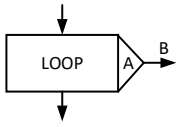
DEPART A,B			Забезпечує в черзі А звільнення В місць

Продовження табл. 6.5

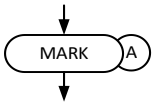
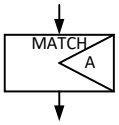
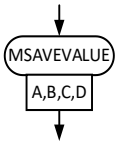
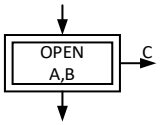
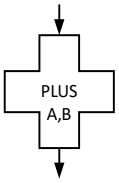
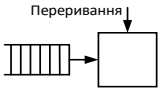
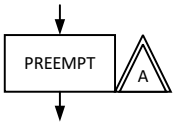
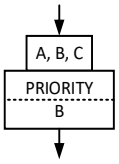
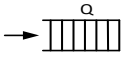
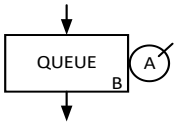
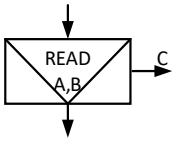
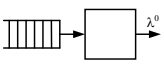
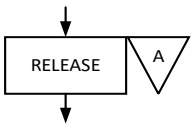
1	2	3	4
DISPLACE A,B,C,D			Шукає транзакт А і переміщує його до блока В
ENTER A,B			Забезпечує вхід транзакта в багатоканальний пристрій А з займанням В каналів
EXAMINE A,B,C			Змінює маршрут руху транзакта залежно від стану членів групи А
EXECUTE A			Дозволяє транзакту виконати будь-яку операцію іншого блока А
FAVAIL A			Переводить пристрій А у доступний стан
FUNAVAIL A,B,C,D,E,F,G,H			Переводить пристрій А у недоступний стан
GATE X{LS / LR} A,B			Управляє потоком транзактів залежно від стану X ключа А
GATE X{U / NU / I / NI / FV / FNV} A,B			Управляє потоком транзактів залежно від стану X одноканального пристрою А

GATE X{SE / SNE / SF / SNF / SV / SNV} A,B			Управляє потоком транзактів залежно від стану X багатоканального пристрою A

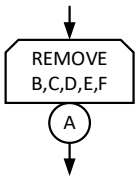
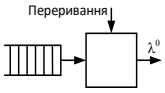
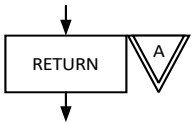
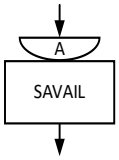
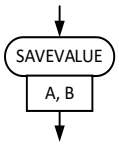
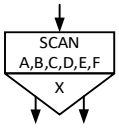
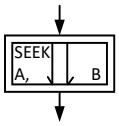
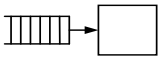
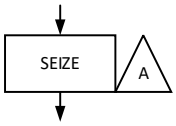
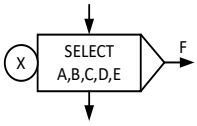
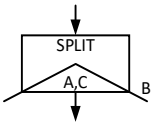
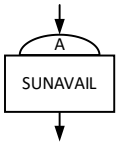
Продовження табл. 6.5

1	2	3	4
GATE X$\begin{Bmatrix} M \\ NM \end{Bmatrix}$ A,B			Управляє потоком транзактів залежно від стану синхронізації X транзакту A
GATHER A			Накопичує задане число A транзактів одного сімейства
GENERATE A,B,C,D,E			Блок генерації транзактів через A одиниць часу з модифікатором B і затримкою C, D транзактів, з пріоритетом E
INDEX A,B			Записує суму значень параметра A і величини B у параметр 1 типу A
INTEGRATION A,B			Вмикання/вимикання параметром B інтегрування змінної A
JOIN A,B			Додає до групи A транзакт або числове значення B
LEAVE A,B			Звільняє в багатоканальному пристрої A B каналів
LINK A,B,C			Приєднує транзакт до списку користувача A з дисципліною обслуговування B
LOGIC X A			Змінює стан логічного ключа A логічним оператором X
LOOP A,B			Організує циклічний рух транзактів на мітку B у моделі кількістю A циклів

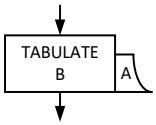
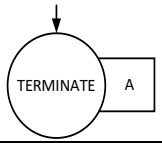
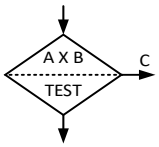
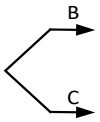
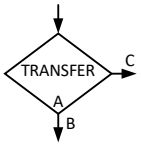
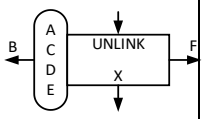
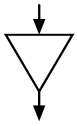
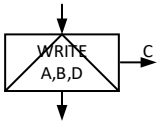
Продовження табл. 6.5

1	2	3	4
MARK A			Ставить позначку часу або записує значення таймера в параметрі А
MATCH A			Синхронізація руху транзактів одного сімейства між двома блоками MATCH з міткою А
MSAVEVALUE A,B,C,D			Записує значення D у комірку з рядком В та стовпцем С у матриці А
OPEN A,B,C			Ініціює потік даних А, у випадку помилки перехід на мітку С
PLUS A,B			Обчислює вираз А і записує результат у параметр В транзакта
PREEMPT A,B,C,D,E			Захоплює (займає з перериванням) пристрій з номером А
PRIORITY A,B			Присвоює пріоритет А транзакту та розміщує його в СЕС – В
QUEUE A,B			Забезпечує зайняття в черзі А В місць
READ A,B,C			Зчитує текстовий рядок з потоку даних з номером В у транзакт А, у випадку помилки перехід на мітку С
RELEASE A			Звільняє пристрій з номером А

Продовження табл. 6.5

1	2	3	4
REMOVE X A,B,C,D,E,F			Видаляє транзакти кількістю B з групи A транзактів або число C з числової групи
RETURN A			Звільняє пристрій з номером A від захоплення
SAVAIL A			Переводить багатоканальний пристрій A у доступний стан
SAVEVALUE A,B			Зберігає задане значення B у комірку A
SCAN X A,B,C,D,E,F			Шукає інформацію за умовою X у групі транзактів A для присвоєння її активному транзакту
SEEK A,B			Встановлює нову поточну позицію A рядка в потоці даних B
SEIZE A			Займає пристрій з номером A
SELECT X A,B,C,D,E,F			Обирає перший об’єкт у параметр A у заданому діапазоні [B..C], що задовольняє умові при порівнянні D і E
SPLIT A,B,C			Створює копії транзакта кількістю A, які виходять на мітку B
SUNAVAIL A			Переводить багатоканальний пристрій A у недоступний стан

Закінчення табл. 6.5

1	2	3	4
TABULATE A,B			Заносить спостережуване значення в таблицю A B разів
TERMINATE A			Знищує A транзактів
TEST X A,B,C			Перевіряє співвідношення X між A і B і направляє транзакт до наступного блока при виконанні або за адресою C при невиконанні співвідношення
TRACE			Вмикає відстежування всіх транзактів у системі
TRANSFER A,B,C,D			Змінює напрям руху транзактів згідно режиму A
UNLINK X A,B,C,D,E,F			Видаляє транзакт зі списку користувача A
UNTRACE			Вимикає відстежування всіх транзактів у системі
WRITE A,B,C,D			Передає текстовий рядок A у потік даних з номером B, у випадку помилки перехід на мітку C

Приклад побудови моделі одноканальної СМО з необмеженою чергою в системі GPSS World. Вихідні дані: інтенсивність надходження заявок $\lambda = 20\ \text{с}^{-1}$; інтенсивність обслуговування заявок $\mu = 40\ \text{с}^{-1}$; показник (коефіцієнт) завантаженості системи $\rho = \frac{\lambda}{\mu} = 0.5$; загальна кількість заявок, які необхідно промоделювати $N = 1000$. Вхідний потік та потік обслуговування підпорядковуються рівномірному розподілу з відхиленням 20 % від середніх значень: надходження заявки відбувається кожні

$1/\lambda = 50 \pm 10$ мс, обробка заявки – кожні $1/\mu = 25 \pm 5$ мс. Масштаб модельного часу $\Delta t = 1$ мс.

Q-схему імітаційної моделі одноканальної розімкнутої СМО без обмежень наведено на рис. 6.4.

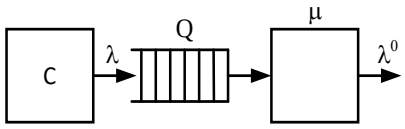


Рисунок 6.4 – Q-схема одноканальної СМО з необмеженою чергою

Блок-діаграму наведено на рис. 6.5.

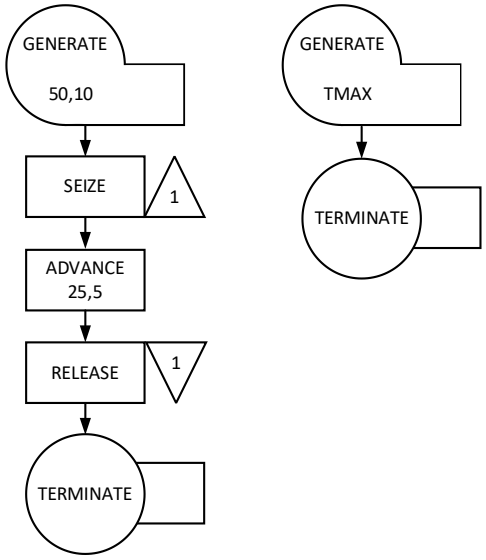


Рисунок 6.5 – Блок-діаграма імітаційної моделі

Загальний модельний час $TMAX$ розраховується як $TMAX = {}^{N}/_{(\lambda \cdot \Delta t)} = {}^{1000}/_{(20 \cdot 10^{-3})} = 50000.$

За блок-діаграмою будується модель мовою GPSS:

```
GENERATE 50,10
QUEUE 1
SEIZE 1
DEPART 1
ADVANCE 25,5
RELEASE 1
TERMINATE

GENERATE 50000
TERMINATE 1
```

Приклад побудови СМО з двома вхідними і двома вихідними потоками заявок. Q-схему СМО наведено на рис. 6.6.

Заявки з обох джерел надсилаються до загального накопичувача Q, з якого – в одноканальний пристрій обслуговування. Після обробки заявки надсилаються до приймачів П1 і П2, причому, у приймач П1 повинні надсилатися заявки, що надійшли до системи з джерела С1, а в П2 – що надійшли з С2.

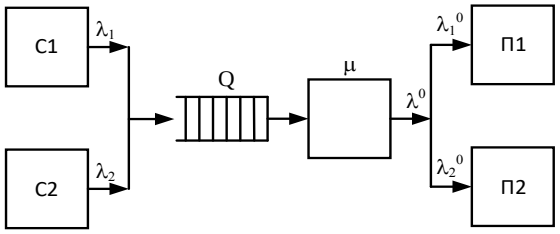


Рисунок 6.6 – Q-схема СМО з двома вхідними і двома вихідними потоками заявок

Блок-діаграму імітаційної моделі наведено на рис. 6.7.

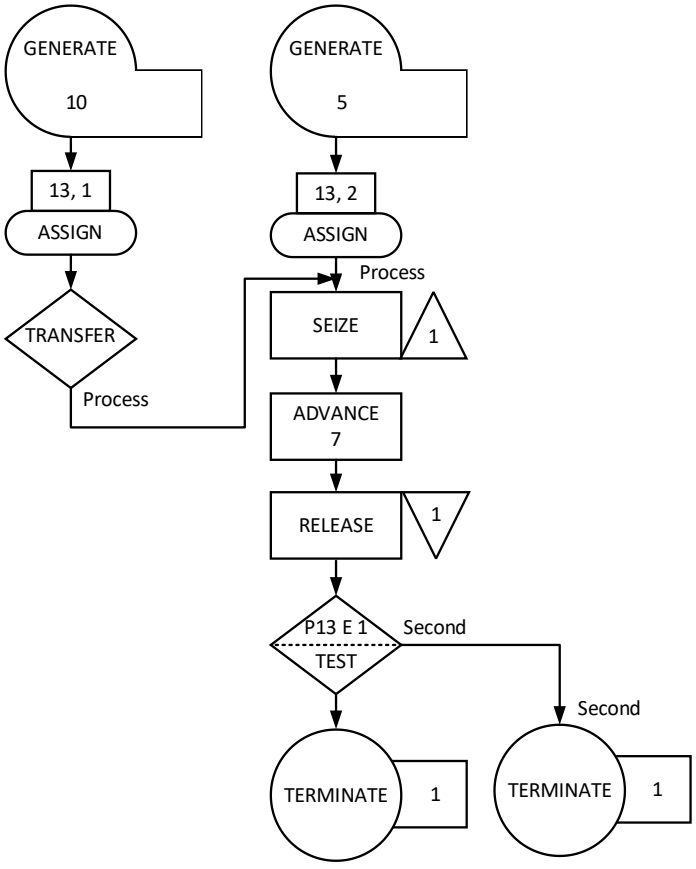


Рисунок 6.7 – Блок-діаграма імітаційної моделі

Потоки транзактів створюються двома блоками GENERATE. Після виходу з блоків GENERATE транзакти потрапляють у блоки ASSIGN, де їхньому 13-му параметру присвоюється значення 1 або 2 залежно від джерела. Ці значення використовуються далі для визначення приналежності заявок. Після виходу з блока ASSIGN транзакти другого потоку потрапляють безпосередньо в блок SEIZE з міткою Process, з яким пов'язано чергу транзактів. Транзакти першого потоку направляються в цей блок за допомогою блока безумовного переходу TRANSFER. Після завершення обробки на пристрої (блок RELEASE) заявки переходять у блок TEST, котрий направляє їх або на наступний блок (TERMINATE), або по мітці Second (другий блок TERMINATE). Умовою перевірки в блоці TEST є рівність (E) значень 13-го параметра транзакта і константи 1.

Модель мовою GPSS:

```
                        ; Перший потік заявок
      GENERATE 10
      ASSIGN 13,1       ; Помітка першого потоку
      TRANSFER ,Process ; Перехід до обробки
                        ; Другий потік заявок
      GENERATE 5
      ASSIGN 13,2       ; Помітка другого потоку
                        ; Пристрій обслуговування для обох потоків
Process SEIZE 1         ; Зайняття каналу
      ADVANCE 7         ; Затримка
      RELEASE 1         ; Звільнення каналу
      TEST E P13,1,Second ; Сортування потоків
                        ; Вихід для першого потоку заявок
      TERMINATE 1
                        ; Вихід для другого потоку заявок
Second TERMINATE 1
```

Приклад побудови СМО з обробкою заявки частками (перша частка пристроєм К1, друга частка – послідовно двома пристроями К2 і К3), причому обробка частин заявки пристроями К1 і К3 повинна виконуватись одночасно (для цього перша частка заявки повинна очікувати в накопичувачі Q1). Після обробки обидві частини заявки виводяться з системи. Q-схему СМО наведено на рис. 6.8.

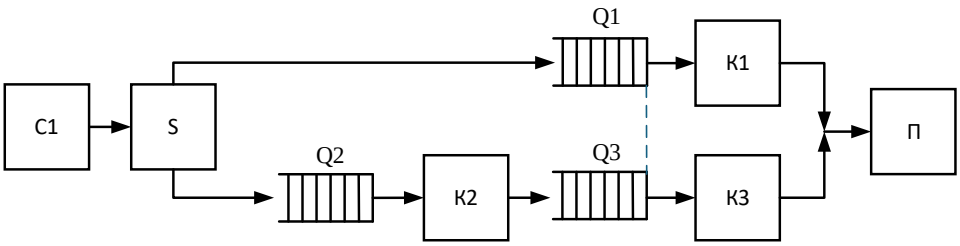


Рисунок 6.8 – Q-схема СМО з синхронізацією обробки заявок

Блок-діаграму імітаційної моделі наведено на рис. 6.9.

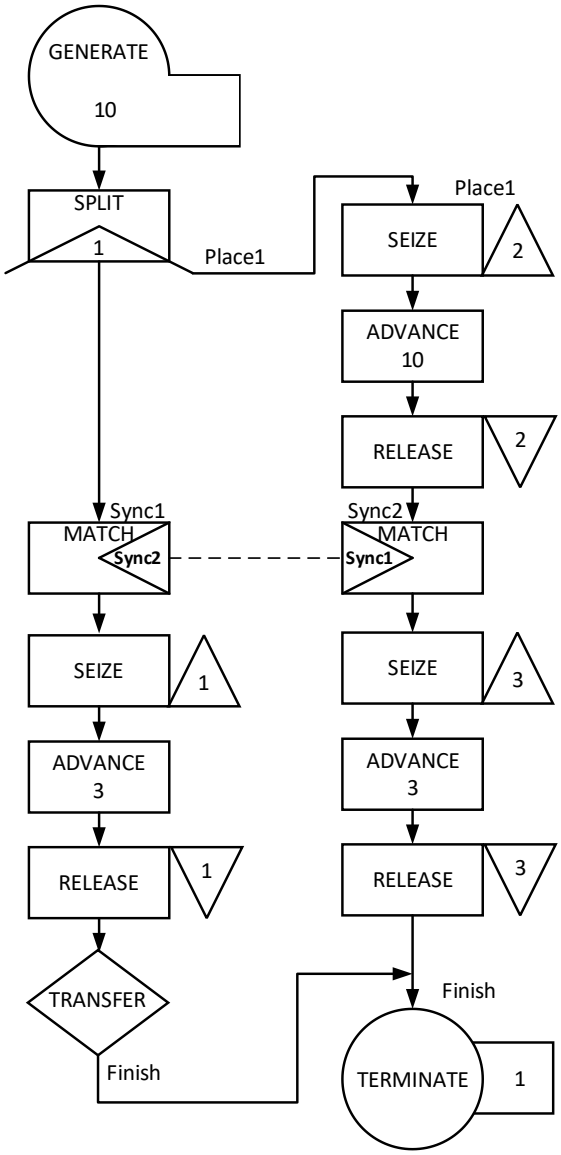


Рисунок 6.9 – Блок-діаграма імітаційної моделі

Модель мовою GPSS:

```
      GENERATE 10,,,100
      SPLIT 1,Place1        ; Поділ транзакта на 2 частини
Sync1 MATCH Sync2           ; Синхронізація з другим потоком
```

```
        SEIZE 1
        ADVANCE 3
        RELEASE 1
        TRANSFER ,Finish        ; Переспрямування на вихід
Place1  SEIZE 2
        ADVANCE 10
        RELEASE 2
Sync2   MATCH Sync1             ; Синхронізація з першим потоком
        SEIZE 3
        ADVANCE 3
        RELEASE 3
Finish  TERMINATE 1             ; Вихід з системи
```

Контрольні запитання

1. Чим характеризуються мови імітаційного моделювання?
2. Опишіть архітектуру мови імітаційного моделювання.
3. Назвіть основні характеристики мови моделювання GPSS.
4. Що таке транзакт?
5. Яким чином подається модель у системі GPSS World?
6. Які сім категорій об’єктів використовуються в GPSS?
7. Що таке стандартні числові атрибути?
8. Який принцип лежить в основі побудови генераторів випадкових чисел у GPSS?
9. Яке призначення черг у GPSS?
10. За що відповідають списки в GPSS?
11. Яким чином реалізовано таймер у GPSS?
12. Назвіть три фази циклу обробки подій у GPSS.
13. Охарактеризуйте методи збору статистики в GPSS.
14. Яким чином побудувати гістограму частот у GPSS?
15. Що таке блок-діаграми та яке їх призначення?

7. ОРГАНІЗАЦІЯ КОМП'ЮТЕРНИХ ЕКСПЕРИМЕНТІВ

У процесі проведення машинного експерименту з моделлю, спрямованого на оцінювання характеристик функціонування досліджуваних систем, необхідно забезпечити умови, що уможливлюють виявлення впливу факторів, функціонально пов'язаних із цільовою характеристикою. Для цього слід здійснити відбір релевантних факторів $x_i, i = \overline{1,m}$, які визначально впливають на досліджуваний показник; встановити допустимі інтервали їх варіювання; визначити координати точок факторного простору $(x_1, x_2, \dots, x_m)$, у межах якого передбачається проведення експерименту; а також обґрунтувати необхідну кількість реалізацій і послідовність їх виконання.

Побудова моделі, що описує реакцію досліджуваної системи на багатофакторний вплив, є однією з ключових задач математичного планування експерименту. Залежність вигляду $Y = \varphi(X)$ може бути подана у формі таблиць або графічних залежностей, а для її апроксимації застосовуються методи регресійного аналізу.

Експерименти в GPSS World можуть виконуватися лише за умови наявності повністю відлагодженої та коректно функціонуючої моделі, яка відображає вплив усіх факторів, що підлягають оцінюванню.

У процесі імітаційного експерименту, з урахуванням випадкових результатів кожного прогону, необхідно:

- класифікувати фактори на суттєві та несуттєві;
- розподілити та кількісно оцінити вплив окремих факторів і їхніх комбінацій на цільову функцію;
- визначити оптимальну комбінацію факторів.

Перша задача розв'язується методами дисперсійного аналізу, друга – регресійного та факторного аналізу (зокрема скринінгу), а третя – засобами статистичної оптимізації.

Концептуально GPSS World підтримує три підходи до проведення експериментів: відсіювальні експерименти, експерименти користувача та оптимізаційні експерименти [17].

У GPSS World для кожного з трьох підходів до проведення експериментів (відсіювальні, користувацькі та оптимізаційні) передбачені спеціальні інструменти підтримки.

Відсіювальні експерименти застосовуються для визначення найбільш значущих факторів, що впливають на систему. Їх результати дозволяють виявити малозначимі фактори, дослідження яких можна мінімізувати.

Користувацькі експерименти більш гнучкі, однак потребують самостійного проєктування та запуску. GPSS World надає засоби підтримки на кожному етапі їх створення. Основним завданням є забезпечення бібліотечної процедури дисперсійного аналізу ANOVA необхідними даними. ANOVA дозволяє аналізувати експерименти з кількістю факторів до 6, включаючи дво- та трифакторні взаємодії.

Оптимізаційні експерименти безпосередньо орієнтовані на задачі пошуку оптимальних умов функціонування системи та кількісного аналізу її поведінки. Центральним елементом є поверхня відгуку – рівняння, що прогнозує результати моделювання. GPSS World забезпечує автоматичну генерацію PLUS-експерименту для побудови поверхні відгуку та пошуку оптимуму. Для цього застосовуються метод найшвидшого підйому/спуску та метод локального пошуку. У разі успіху результати у вигляді рівняння поверхні відгуку та прогнозованих оптимальних умов виводяться у вікно «Journal» об’єкта «Процес моделювання».

З позиції представлення поведінки моделюючої системи імітаційні моделі відносяться до класу описових моделей. Це означає, що вони відображають систему за допомогою опису її функціонування в динаміці. Якщо в моделі враховуються випадкові фактори, то в процесі імітації, як правило, виконується значна кількість прогонів моделі – як з різними наборами вхідних даних, так і з різними значеннями послідовностей

випадкових чисел, що генеруються. Для детермінованої моделі (тобто без урахування випадковостей) достатньо лише одного прогона для кожної комбінації вхідних параметрів. Проте на практиці повністю детерміновані моделі зустрічаються вкрай рідко.

У результаті експериментування з імітаційною моделлю накопичується велика кількість вихідних даних, які потребують структурування, аналізу та інтерпретації. Лише після цього вони можуть бути використані для прийняття рішень щодо функціонування реальної системи, яка моделюється. Для коректної інтерпретації даних, що отримуються від моделі, необхідно правильно організувати імітаційний експеримент.

Організація експерименту означає розробку плану його проведення, який дозволяє за мінімальну кількість прогонів і при найменших витратах отримати статистично значущі висновки або знайти оптимальне рішення задачі. При цьому зазвичай визначають:

- набір вхідних даних для кожного експерименту;
- кількість прогонів імітаційної моделі;
- тривалість одного прогона;
- тривалість перехідного періоду, після завершення якого починається збір вихідних результатів;
- стратегію збору та накопичення даних для кожного прогона;
- методи оцінки точності отриманих результатів, включно з побудовою довірчих інтервалів;
- чутливість моделі до різних вхідних даних, типів розподілів та сценаріїв функціонування системи;
- умови проведення експерименту та досліджувані сценарії;
- правила генерації потоків випадкових чисел у системі та для ймовірнісних вхідних параметрів;
- стратегію досягнення мети експерименту (наприклад, порівняння альтернативних рішень або оптимізація цільової функції).

Кінцева мета експериментів полягає в отриманні достатнього обсягу статистичної інформації для прийняття обґрунтованих управлінських чи технічних рішень. Зазвичай імітаційні експерименти виконуються для:

- пошуку екстремальних значень характеристик системи (оптимізаційний експеримент);
- визначення важливих факторів, що впливають на систему (факторний або відсіювальний експеримент).

Для цього використовують методи планування факторних експериментів і апроксимацію поверхонь відгуку поліномами різного порядку. Для пошуку екстремумів застосовуються чисельні методи оптимізації.

У таких експериментах будується функціональна залежність між вхідними змінними (факторами) та вихідними змінними (відгуками). Ця залежність відображає критерій ефективності моделюючої системи. Таким чином, пошук оптимального рішення зводиться до визначення числових характеристик цього критерію. Для знаходження екстремумів досліджується поверхня відгуку в різних точках факторного простору.

Особливе значення має вибір початкової точки у факторному просторі, оскільки саме вона визначає ефективність експерименту.

Інший вид експериментів – це структурна оптимізація, яка полягає в пошуку найкращої структури моделі системи. У цьому випадку досліджуються не лише варіації параметрів однієї моделі, а й різні структурні варіанти моделей. Вони можуть відрізнятися топологією, алгоритмами функціонування або параметрами. Для структурної оптимізації зазвичай відсутній єдиний числовий критерій, що ускладнює застосування класичних методів оптимізації. Проте, оскільки кількість розглянутих варіантів зазвичай невелика, застосовують методи перебирання гіпотез та сценаріїв, орієнтуючись на порівняння результатів.

При структурній оптимізації ключовим завданням є виявлення «вузьких місць» у системі та їх усунення, що дозволяє підвищити пропускну

здатність і балансувати систему. Оптимальне рішення визначається шляхом порівняння альтернативних структурних варіантів.

Основні проблеми, які виникають при проведенні імітаційних експериментів:

- Задання початкових умов. Експерименти зазвичай починаються зі стану «порожньо і вільно», тобто без транзакцій у системі. Це породжує «перехідний період» або «розігрів моделі», після якого система виходить на стаціонарний режим. Дані цього періоду зміщують оцінки, тому їх рекомендується відкидати (наприклад, за допомогою команди RESET). Команда RESET залишає транзакти в моделі, але обнуляє статистику та звільняє об’єкти обслуговування. Стандартний числовий атрибут C1 повертає значення модельного часу з моменту виконання останньої команди RESET; абсолютний числовий атрибут AC1 показує абсолютний модельний час з початку моделювання. Оператор RESET зазвичай використовується при моделюванні нестабільних (нестаціонарних) процесів, коли необхідно збирати статистику по окремих інтервалах стаціонарності, тобто аналізувати поведінку системи лише після досягнення певного сталого режиму роботи або виключити вплив перехідного періоду на статистичні дані, що збираються, адже дані переходу моделі з початкового стану до стаціонарного режиму можуть спотворювати оцінки характеристик системи.
- Визначення правила зупинки. Тривалість прогона безпосередньо впливає на точність оцінок. Занадто короткі прогони можуть призвести до статистичної ненадійності.
- Стан системи в момент зупинки. Часто виникає питання: як враховувати незавершені транзакти? Їх наявність може викривлювати оцінку середніх показників (наприклад, середньої тривалості виконання роботи).

- Різні швидкості процесів у моделі. Точність оцінок орієнтується на «найповільніший процес», у той час як для швидких процесів результати можуть мати надлишкову точність. Якщо різниця в часі надто велика, використовуються аналітико-імітаційні моделі або укрупнення станів.

Після того як модель була протестована в різних умовах, виникає завдання її подальшого дослідження. У цей момент з'являється необхідність у проведенні спеціально організованих експериментів. Для того щоб визначити реакції (відгуки) факторів, які знаходяться під контролем експериментатора, потрібно змоделювати кожну з можливих комбінацій і перевірити, чи перевищує отриманий результат очікуване значення лише внаслідок випадкових змін.

Модель повинна тестуватися в найнапруженіших умовах. Вона має бути оснащена інструментами, які забезпечують усі необхідні вимірювання для статистичного аналізу. У типовій задачі експеримент складається з трьох прогонів для кожного з чотирьох рівнів факторів. Прогони виконуються при однакових умовах; змінюється лише початкове значення генератора послідовностей псевдовипадкових чисел. Це робиться для створення бази варіативності результатів і забезпечення статистичної надійності.

У системі GPSS World визначення експерименту відбувається під час трансляції спеціальної процедури PLUS, яка називається EXPERIMENT.

Відмінність між звичайною процедурою та експериментом полягає в тому, що:

- ключове слово PROCEDURE замінюється ключовим словом EXPERIMENT;
- у межах експерименту можна використовувати процедури бібліотеки «DoCommand».

Таким чином, експерименти успадковують усю потужність мови PLUS, що робить їх повністю програмованими та здатними до автоматизації.

Для запуску раніше визначеного експерименту може бути використана команда CONDUCT.

Якщо виклик явно не визначений у коді моделі, команда CONDUCT може бути застосована інтерактивно. Для запуску PLUS-експерименту достатньо використати текст виклику процедури як єдиний операнд команди CONDUCT.

Після запуску експерименту можливості взаємодії з моделлю обмежуються. Завжди можна переглянути модельний час (View / Clock), але для детального вивчення моделі необхідно призупинити експеримент.

У процесі експериментування розрізняють:

- вхідні змінні (фактори),
- вихідні змінні (відгуки).

Кожен фактор може приймати одне чи кілька значень (рівні фактора). Комбінація рівнів усіх факторів задає один зі станів моделі і визначає умови проведення конкретного експерименту.

Факторні експерименти бувають двох основних типів:

- відсіювальні експерименти (screening experiments). Використовуються для виявлення тих факторів, які істотно впливають на відгуки моделі;
- експерименти для пошуку екстремумів. Вони плануються так, щоб дослідити поверхню відгуку і знайти точки максимуму або мінімуму.

Факторний експеримент являє собою план, в якому всі рівні кожного фактора поєднуються з усіма рівнями інших факторів. Фактори можуть бути якісними та кількісними.

Застосування факторного плану замість класичної схеми, відповідно до якої кожного разу змінюється лише один фактор при фіксації всіх інших, має низку суттєвих переваг.

Формується більш повна та цілісна картина впливу кожного окремого фактора, оскільки їхнє дослідження відбувається в умовах різноманітних комбінацій, тобто за рахунок одночасної зміни інших факторів.

Велика кількість комбінацій факторів, що використовується в експерименті, значно полегшує прогнозування результатів, яких можна досягти при певному поєднанні умов.

Якщо ефекти, зумовлені дією кожного фактора, є статистично незалежними, то для кожного фактора можна отримати не менше інформації, ніж у класичних експериментах із варіюванням лише одного фактора при фіксації решти.

Якщо ж (що трапляється доволі часто) різні фактори не є незалежними, а створюють ефекти, які в більшій чи меншій мірі корелюють між собою, то лише факторний експеримент здатен дати достовірну інформацію про характер цих взаємодій. При наявності декількох взаємопов’язаних істотних факторів проведення дослідження без постановки факторного експерименту стає неможливим.

Для розв’язання ряду типових прикладних задач, що часто зустрічаються в експериментальній практиці, було розроблено значну кількість стандартних факторних планів.

Система GPSS World забезпечує два види автоматично виконуваних експериментів:

- розроблені для користувача,
- розроблені самим користувачем.

Автоматичні генератори відсіювальних (дисперсійний аналіз) та оптимізаційних (регресійний аналіз) експериментів, які вбудовані в GPSS World та призначені для користувача, не передбачають багаторазових прогонів моделі. Тобто ці експерименти виконуються з моделями, що функціонують у стаціонарному режимі. Для таких моделей оцінки точності результатів моделювання можуть бути отримані під час одного, достатньо тривалого прогона моделі. При цьому дані, що належать до перехідного

періоду, у таких експериментах можуть не враховуватися. Такі моделі не передбачають повторення експериментів для кожного рівня поєднання факторів.

Користувацькі експерименти можуть проводитися з моделями, що працюють у перехідному режимі, і надають можливість організувати повторення експериментів. У цьому випадку необхідно заздалегідь, ще до планування експериментів, визначити для кожної комбінації рівнів факторів кількість повторень експериментів. Наявність вбудованої мови програмування PLUS дає змогу розробляти користувацькі експерименти будь-якої складності.

7.1. Врахування перехідного режиму роботи моделі під час комп'ютерного моделювання

Оцінка точності результатів моделювання безпосередньо пов'язана з побудовою довірчих інтервалів для вихідних змінних (відгуків) моделі. Саме довірчі інтервали дозволяють визначити, наскільки близькими до істинних значень є отримані в процесі моделювання результати.

Точність результатів моделювання залежить від двох основних факторів:

- кількість реалізацій (прогонів) моделі – чим більше незалежних прогонів виконується, тим надійнішими будуть статистичні оцінки;
- тривалість прогона для кожної реалізації – чим довше триває кожен прогон, тим більший масив даних формується, а отже, підвищується точність оцінки середніх характеристик.

У випадку, коли модель детермінована, достатньо одного прогона для отримання точних результатів, оскільки за однакових початкових умов вихідні дані завжди будуть відтворюватися без відхилень.

У загальному випадку дані одного прогона моделі становлять одиничну вибірку або часовий ряд – кінцеву реалізацію випадкового процесу. Таким

чином, у результаті кожного прогона моделі формуються часові ряди для кожного значення відгуку моделі, що відповідають досліджуваним стохастичним процесам.

Як відомо, для стохастичних моделей виділяють два режими роботи:

- перехідний режим, коли модель ще не досягла сталого функціонування і результати відображають початкові флуктуації;
- стаціонарний режим, коли вихідні змінні моделі відповідають стаціонарному процесу, тобто мають сталі статистичні характеристики (середнє, дисперсія тощо).

Якщо модель функціонує в перехідному режимі, то необхідну кількість прогонів можна обчислити за спеціальними формулами, які застосовуються в методі статистичних випробувань Монте-Карло. Цей підхід дозволяє визначити мінімальну кількість прогонів, необхідних для отримання статистично достовірних результатів заданої точності.

Припустимо, що метою моделювання є оцінка ймовірності настання деякої події А, яка визначає певний стан модельованої системи. Відповідно, необхідно визначити кількість реалізацій для оцінки ймовірності настання події А.

У кожній з N реалізацій процес настання події А є випадковою величиною, яка може набувати значень:

- $x_1 = 1$ – у випадку, коли подія А настала, з ймовірністю p,
- $x_2 = 0$ – у випадку, коли подія А не настала, з ймовірністю $1 - p$.

Математичне сподівання і дисперсія випадкової величини:

$$M = x_1 p + x_2(1-p), \qquad D = (x_1 - M)^2 p + (x_2 - M)^2(1-p) = p(1-p).$$

В якості оцінки ймовірності p використовується частота настання події

$$\frac{m}{N} = \frac{1}{N}\sum_{i=1}^{N} m_i$$

де m – кількість виникнення подій, $m_i = \{0,1\}$ – настання події в реалізації, з властивостями:

- незміщеність – у середньому дорівнює істинній імовірності;
- спроможність – при збільшенні N наближається до p;
- ефективність – має мінімальну можливу дисперсію серед усіх незміщених оцінок.

Згідно з центральною граничною теоремою Лапласа, випадкова величина m/N буде мати розподіл, близький до нормального:

$$M\left[\frac{m}{N}\right] = p, \qquad D\left[\frac{m}{N}\right] = \frac{p(1-p)}{N-1}.$$

Для довірчої ймовірності α з таблиць нормального розподілу обирається відповідне значення аргументу функції Лапласа t_α і точність ε оцінки ймовірності дорівнюватиме:

$$\varepsilon = t_\alpha \sqrt{D\left[\frac{m}{N}\right]},$$

при $\alpha = 0.9$ $t_\alpha = 1.65$, $\alpha = 0.95$ $t_\alpha = 1.96$, при $\alpha = 0.997$ $t_\alpha = 3$.

Підставляючи дисперсію, можна отримати формулу для визначення кількості реалізацій:

$$\varepsilon = t_\alpha \sqrt{\frac{p(1-p)}{N-1}},$$

звідки

$$N = t_\alpha^2 \frac{p(1-p)}{\varepsilon^2} + 1.$$

Оскільки істинне значення p заздалегідь невідоме, на практиці використовують пробні випробування (наприклад, $N = 50 \ldots 100$), обчислюють емпіричну частоту m/N і підставляють цю оцінку замість p у формулу вище. Після цього обчислюють остаточне значення кількості випробувань, необхідне для заданої точності ε та довірчої ймовірності α. У формулі вище можна побачити, що кількість прогонів змінюється від 1 $(p = 0)$ до 1 $(p = 1)$ через максимум при $p = 0.5$ – найгірший випадок, тому таку ситуацію необхідно також враховувати. Якщо за результатами

моделювання з'ясується, що ймовірність p істотно відрізняється від 0.5 (у будь-який бік), то фактична точність моделювання буде вищою за задану.

Припустимо, що метою моделювання є оцінка середнього значення випадкової величини. Нехай випадкова величина має математичне сподівання M і дисперсію $D = \sigma^2$. У реалізації з номером i ця величина набуває значення $x_i,\ i = 1, \dots, N$. Для оцінювання математичного сподівання M використовується вибіркове середнє:

$$\overline{x} = \frac{1}{N}\sum_{i=1}^{N} x_i.$$

Згідно з центральною граничною теоремою, при достатньо великому N розподіл вибіркового середнього $\overline{x}$ буде наближено до нормального зі значенням математичного сподівання M і дисперсією $D = {\sigma^2}/{N-1}$. Отже,

$$\varepsilon = t_\alpha \frac{\sigma}{\sqrt{N-1}},$$

звідки

$$N = t_\alpha^2 \frac{\sigma^2}{\varepsilon^2} + 1.$$

Оскільки дисперсія випадкової величини невідома, необхідно провести пробні випробування ($N_0 = 50 - 100$ прогонів), з яких обчислюється вибіркова дисперсія

$$s^2 = \frac{1}{N_0 - 1}\sum_{i=1}^{N_0} (x_i - \overline{x})^2$$

та підставляється у формулу:

$$N \approx t_\alpha^2 \frac{s^2}{\varepsilon^2} + 1.$$

Необхідну точність ε можна задати як ±5 % від середнього значення тієї величини, для якої будується довірчий інтервал. Якщо модель реалізована мовою GPSS, після завершення останнього прогону (після

накопичення остаточної серії реалізацій) обов'язково слід виконати процедуру ANOVA.

Якщо кількість прогонів моделі є невеликою (менш ніж 30), то при побудові довірчого інтервалу для середнього значення випадкової величини застосовують розподіл Стьюдента (t-розподіл). Це пояснюється тим, що при малих вибірках невідоме істинне значення дисперсії і її доводиться замінювати вибірковою оцінкою. Така заміна вносить додаткову невизначеність, яку і враховує t-розподіл. При більшій кількості прогонів ($N \geq 30$) вибіркове середнє підпорядковується нормальному закону розподілу в силу центральної граничної теореми, і тому для побудови довірчого інтервалу можна застосовувати нормальний розподіл.

Якщо критерієм оцінки є вартісна характеристика (наприклад, дохід, прибуток, витрати тощо), яка визначається для стаціонарного режиму роботи моделі, то довжина одного прогона може бути визначена на основі результатів спостереження за зміною величини, що дорівнює відношенню оцінюваного показника за весь період моделювання до тривалості цього моделювання (наприклад, витрати в розрахунку на одиницю часу).

Для цього в вікні PLOT будують графік зміни цієї величини в часі. По графіку визначають момент, коли система досягає стаціонарного режиму (лінія стабілізується, коливання не мають систематичної тенденції). Саме цей проміжок часу і визначає необхідну довжину прогона. Важливо, щоб результати аналізу враховували лише дані стаціонарного режиму функціонування моделі.

Статистичні дані, отримані в перехідному періоді (на початку моделювання, коли система ще не вийшла на стаціонарний режим), не повинні враховуватись, оскільки вони можуть спотворити оцінку. Для цього в GPSS застосовується команда RESET, яка «обнуляє» статистику, очищає накопичені дані і забезпечує початок збору статистичних показників з моменту настання стаціонарності (рис. 7.1).

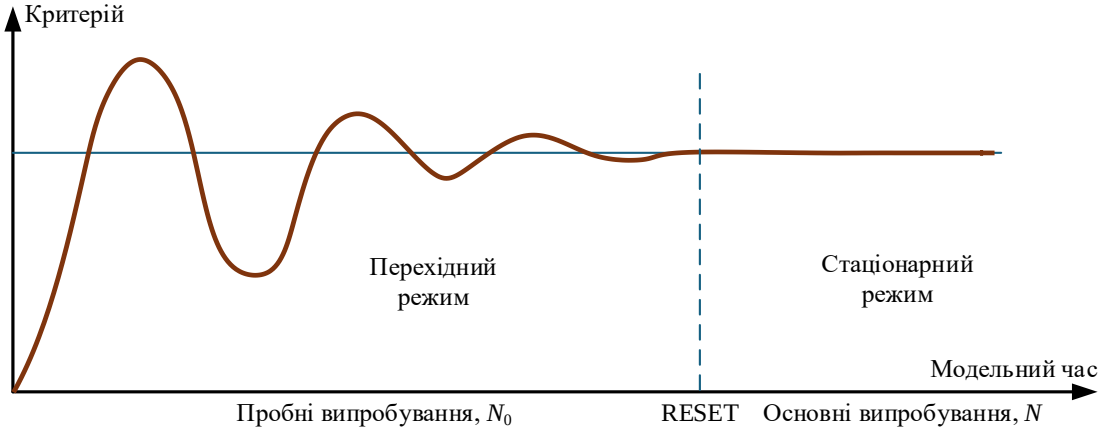


Рисунок 7.1 – Графічне визначення кількості пробних випробувань

Також стаціонарність процесу на виході моделі можна перевірити, спостерігаючи за його автоковаріаційною функцією. Якщо автоковаріаційна функція має тенденцію до згасання, це, як правило, свідчить, принаймні в першому наближенні, про те, що процес є стаціонарним. Оскільки автоковаріаційна функція випадкового процесу невідома, її можна лише оцінити. Враховуючи, що часові ряди, отримані з імітаційної моделі, зазвичай досить довгі, автоковаріаційну функцію cov_h можна оцінити за формулою:

$$\widehat{cov}_h = \frac{1}{N}\sum_{i=1}^{N-h}(x_i x_{i+h}) - \frac{1}{N-h}\left(\sum_{i=1}^{N-h} x_i\right)\left(\sum_{i=1}^{N-h} x_{i+h}\right),$$

де N – кількість точок часового ряду, h – зсув по ряду, x_i – i-те значення вимірюваної змінної.

Особливе значення при імітаційному моделюванні мають стаціонарні ергодичні процеси, властивості яких можна оцінити за результатами одного часового ряду. Послідовність вибіркових середніх значень $\{\overline{X}_N\}, N = 1,2,\dots,\infty$ вважається ергодичною, якщо дисперсія величини $\overline{X}_N$ прямує до нуля при необмеженому збільшенні N:

$$D[\overline{X}_N] \to 0 \text{ при } \overline{X}_N \to \infty.$$

У цьому випадку вибіркове середнє асимптотично прагне до математичного сподівання. Таким чином, встановлений стаціонарний режим роботи моделі не залежатиме від початкових умов моделювання.

Щоб перевірити на практиці, чи є процес ергодичним, необхідно різко змінити початкові умови моделювання. Наприклад, якщо модель спочатку запускалася при порожніх чергах та вільних пристроях обслуговування, другий прогін моделі слід провести при наявності транзактів у чергах та зайнятих пристроїв. Якщо на виході моделі будуть отримані близькі результати за таких різко відмінних початкових умов, це зазвичай свідчить про ергодичність процесу.

Для імітаційних стохастичних моделей, які працюють у перехідному (нестаціонарному) режимі, необхідно виконати кілька прогонів моделі, кожен з яких відрізняється своєю послідовністю псевдовипадкових чисел. Для цього зазвичай використовується послідовність команд такого типу:

RMULT 489
START 1
CLEAR
RMULT 268
START 1
CLEAR
….
RMULT 473
START 1

де команда **RMULT** – встановлює початкові значення множників для генераторів випадкових чисел (дозволяє через параметри одночасно встановити значення для семи генераторів, якщо становити значення 0 – відповідний генератор буде генерувати нулі), команда **CLEAR** – здійснює скидання (обнулення) усієї статистики, накопиченої під час попереднього прогона моделі, видаляє всі транзакти з поточного стану моделі), черги очищуються, прилади вивільняються, але не встановлює множники генераторів випадкових чисел в їх початкові значення: послідовність псевдовипадкових чисел продовжується далі з того місця, де завершився попередній прогін. На відміну від команди CLEAR, команда RESET не

видаляє активні транзакти з моделі, тому слід використовувати CLEAR для повного перезапуску моделі, а RESET для скидання статистики після переходу моделі в стаціонарний режим та продовження моделювання.

7.2. Проведення дисперсійного аналізу (відсіювальний експеримент) у GPSS World

Такий експеримент використовується для визначення факторів, що суттєво впливають на систему. Отримана інформація є важливою для проведення подальших експериментів з метою створення більш ефективних оптимізаційних експериментів.

Результати відсіювального експерименту показують, які фактори є менш значущими та повинні надалі отримати нижчий пріоритет під час розгляду. Для отримання достовірних результатів необхідно правильно добирати значення факторів, з якими буде проводитися експеримент. У випадках, коли область допустимих значень фактора є великою, може виникнути потреба в проведенні кількох експериментів для різних значень факторів з цієї області.

У загальному випадку значення кожного фактора бажано обирати рівновіддалено від меж області та одне від одного.

Щоб скористатися вбудованим генератором відсіювальних експериментів GPSS World, необхідно виконати пункт меню Edit/Insert Experiment/Screening, у результаті чого з’явиться діалогове вікно Screening Experiment Generator (рис. 7.2). Відсіювальний експеримент проводиться лише для двох значень кожного рівня факторів (Value 1 і Value 2 на рис. 7.2), оскільки в його основу покладено лінійну регресійну модель. Fraction (доля) визначає, яка частина від повного факторного експерименту буде виконана, тобто будується план 2^{k-p}. У цьому випадку прогони виконуються для кожного варіанта значень рівнів факторів, заданих у полях Value 1 і Value 2. Result Expression – це вираз для вихідної змінної, для якої проводиться

експеримент, у це поле можуть бути введені назви змінних користувача, системні числові атрибути або вирази.

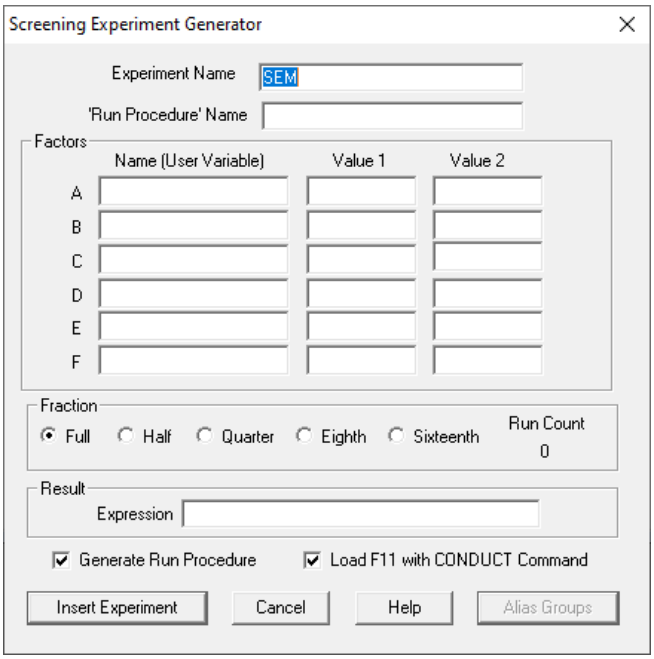


Рисунок 7.2 – Вікно GPSS World для налагодження відсіювального експерименту

Для організації експерименту необхідно мати налагоджену імітаційну модель, яка повинна бути перевірена при всіх можливих значеннях вхідних змінних. Потрібно гарантувати, що модель буде коректно працювати за будь-якої комбінації рівнів факторів.

У результаті заповнення полів діалогового вікна до цієї моделі автоматично буде додано код мовою PLUS. За вибором користувача для запуску експерименту може бути призначена функціональна клавіша (зазвичай F11, але її можна перевизначити).

Процедура дисперсійного аналізу ANOVA доцільна лише для кількісних факторів. За допомогою ANOVA визначаються числові відхилення результатів спостережень від середнього значення. Ця процедура дозволяє аналізувати експерименти з кількістю факторів від 1 до 6 (фактори A, B, C, D, E, F на рис.7.2), включаючи взаємодії 2-го та 3-го порядку між факторами.

Мета та результат роботи ANOVA:

- створення стандартної ANOVA-таблиці;
- визначення значущості факторів за F-критерієм Фішера;
- обчислення довірчих інтервалів для досліджуваних величин.

Якщо значення F-критерію не перевищує критичного, то фактор вважається незначущим, а вибірки належать до однієї генеральної сукупності.

Для роботи процедури ANOVA їй потрібно передати ім’я GPSS-матриці, в якій збережені результати всіх прогонів моделі. Можна мати кілька таких матриць результатів і викликати процедуру для кожної з них окремо. У GPSS World матриці результатів можуть мати максимальну розмірність 6, що і обмежує максимальну кількість факторів, які можуть бути проаналізовані. Оголошення в моделі матриці виконується таким чином:

NAME MATRIX A,B,C,[D],[E],[F],[G]

де NAME – ім’я матриці, A – не використовується (для сумісності зі старішими версіями GPSS), B – кількість рядків (рівні факторів, Treatment levels), C – кількість стовпців (кількість повторів для кожного рівня), D,E,F,G – максимальна кількість елементів у третьому, четвертому, п’ятому і шостому вимірах відповідно.

При виклику процедури ANOVA потрібно задати 3 параметри:

- ім’я матриці результатів (наприклад, MyResults);
- розмірність (індекс) матриці для повторів – якщо експеримент передбачає повтори з різними початковими послідовностями випадкових чисел, то ця розмірність використовується для збереження результатів кожного повтору, якщо повтори не використовуються, то другий параметр дорівнює нулю;
- рівень врахування взаємодій між факторами: 1 – без взаємодій, 2 – включаються взаємодії 2-го порядку, 3 – враховуються взаємодії 2-го і 3-го порядків. Взаємодії вищих порядків у GPSS World не підтримуються.

Перед початком експерименту необхідно ініціалізувати елементи матриці в стан UNSPECIFIED за допомогою оператора:

INITIAL A,B

де A – ім’я матриці результатів, B – задає початкове значення, що присвоюється, може бути назвою, числом, рядком або UNSPECIFIED. Якщо операнд B не використовується, то значення стає рівним 1.

Стан UNSPECIFIED вказує програмі ANOVA, що експеримент ще не завершений.

Кожний вимір матриці інтерпретується процедурою ANOVA як фактор. Розмірність матриці результатів має бути не меншою за кількість рівнів цього фактора або максимально необхідної кількості повторів.

Під час проведення дисперсійного аналізу для кожного рівня обробки досліджуваного фактора необхідно виконати кілька прогонів моделі. Кількість прогонів визначається залежно від необхідної точності результатів. Щоб зменшити кореляцію між результатами окремих прогонів, рекомендується, аби кожний прогін використовував власну послідовність псевдовипадкових чисел.

Після завершення кожного прогона моделі потрібно:

- зберегти отримані результати моделювання;
- обнулити статистику, що була накопичена в попередньому прогоні;
- видалити транзакти з моделі.

Для реалізації цих дій у GPSS World використовуються такі оператори:

- RMULT – для встановлення початкових значень множників для генераторів випадкових чисел;
- CLEAR – для здійснення обнулення накопиченої статистики за попередній прогін моделі та видалення транзактів з моделі. Операнд A = OFF забороняє обнулення матриці результатів;

- START – для ініціювання процесу моделювання. При вказівці в другому операнді NP статистика не буде виводитися на екран;
- MSAVEVALUE – для запису отриманих результатів моделювання в спеціальну матрицю результатів.

Фрагмент тексту моделі при проведенні дисперсійного аналізу в GPSS World:

```
ANOVA_Results MATRIX ,2,2,2
INITIAL ANOVA_Results,UNSPECIFIED
...
FACTOR EQU N1 ; N1 – значення першого рівня фактора
RMULT Z11 ; Z11 – випадкове число,
          ; початкове значення для генератора випадкових чисел
START 1,NP
MSAVEVALUE ANOVA_Results,1,1,X$Rez ; X$Rez – результат моделювання
CLEAR OFF
…
RMULT Z1N ; Z1N – випадкове число,
          ; початкове значення для генератора випадкових чисел
START 1,NP
MSAVEVALUE ANOVA_Results,1,N,X$Rez ; X$Rez – результат моделювання
CLEAR OFF
…
FACTOR EQU NI ; NI – значення i-го рівня фактора
RMULT ZI1 ; ZI1 – випадкове число,
          ; початкове значення для генератора випадкових чисел
START 1,NP
MSAVEVALUE ANOVA_Results,I,1,X$Rez ; X$Rez – результат моделювання
CLEAR OFF
…
RMULT ZIN ; ZIN – випадкове число,
          ; початкове значення для генератора випадкових чисел
START 1,NP
MSAVEVALUE ANOVA_Results,I,N,X$Rez ; X$Rez – результат моделювання
CLEAR OFF
…
FACTOR EQU NK ; NK – значення k-го рівня фактора
RMULT ZK1 ; ZK1 – випадкове число,
          ; початкове значення для генератора випадкових чисел
START 1,NP
MSAVEVALUE ANOVA_Results,K,1,X$Rez ; X$Rez – результат моделювання
CLEAR OFF
…
RMULT ZKN ; ZKN – випадкове число
START 1,NP
MSAVEVALUE ANOVA_Results,K,N,X$Rez ; X$Rez – результат моделювання
CLEAR OFF
```

Слід зауважити, що процес моделювання при виконанні дисперсійного аналізу може зайняти певний час залежно від кількості прогонів моделі (якщо система має перехідний режим роботи, то ще крім виклику START 1,NP необхідно додати для кожного рівня фактора мінімальну кількість прогонів, необхідних для отримання статистично достовірних результатів заданої точності).

7.3. Проведення регресійного аналізу (оптимізаційний експеримент) у GPSS World

Оптимізаційний експеримент призначений для побудови рівняння поверхні відгуку для заданих факторів моделі та пошуку числового значення оптимуму (поєднання таких значень факторів, за якого задана функція набуває екстремального значення – максимуму або мінімуму). Максимально можлива кількість змінюваних факторів – п’ять.

Експеримент проводиться з уже створеною моделлю. Фактори, які змінюватимуться, мають бути в моделі представлені у вигляді констант, визначених за допомогою оператора EQU, або змінних. У процесі експерименту здійснюється багаторазовий прогін моделі, фіксація результатів і використання їх для побудови поверхні відгуку та пошуку оптимуму.

Оптимізаційний експеримент додається до раніше створеної та налагодженої моделі. Для цього необхідно вибрати пункт меню Edit/Insert Experiment/Optimizing, де виконати задання необхідних параметрів для виконання експерименту (рис. 7.3).

У полях Experiment Name і Run Procedure Name задаються відповідно назва експерименту та назва Run-процедури прогонів моделі. За замовчуванням назва експерименту – RSM (від Response Surface Model). Це ім’я буде додано до всіх назв процедур і змінних, які автоматично генеруються для цього експерименту.

Користувачеві доступні для редагування всі процедури, що стосуються експерименту, крім однієї – RSM_FitSurfaceToData (її ім’я є фіксованим і не змінюється навіть при заданні іншої назви експерименту). Як назву Run-процедури рекомендується використовувати DoTheRun, хоча можна призначити й будь-яке інше ім’я.

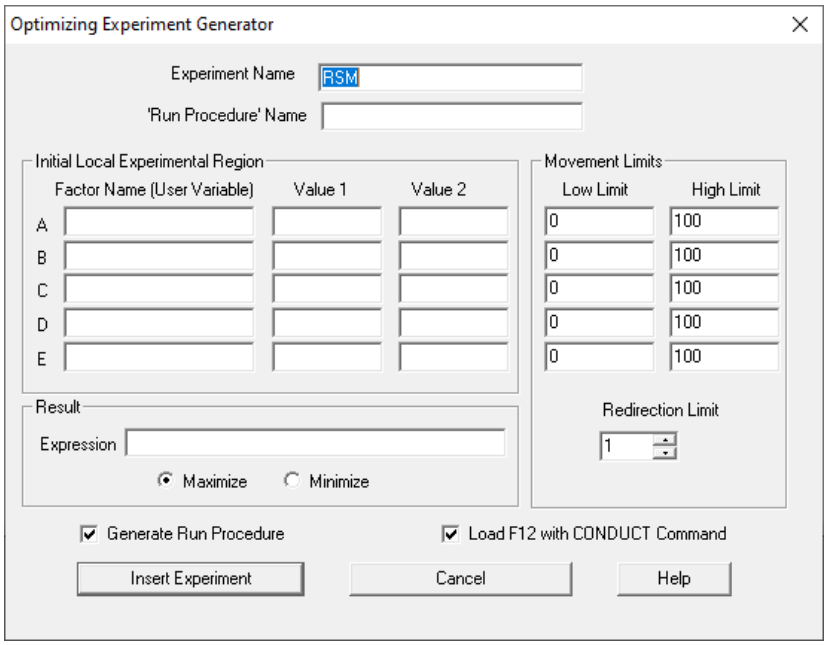


Рисунок 7.3 – Вікно GPSS World для налагодження оптимізаційного експерименту

Для кожного фактора обираються два рівні обробки, що формують локальну експериментальну область. Фактори задаються послідовно (A, B, C, D, E) та вводяться у полях Factor Name A, B, C, D, E, також задаються значення їхніх верхніх і нижніх рівнів (Value 1 і Value 2). Вибір рівнів факторів є критично важливим, оскільки він визначає якість пошуку. Поля Low Limit і High Limit у групі Movement Limits задають максимальні межі переміщення (іноді можуть допомогти при пошуку оптимуму, але в більшості випадків варто залишати значення за замовчуванням – 0 і 100). Значення поля Redirection Limit обмежує кількість змін напрямків при пошуку оптимуму. У деяких випадках достатньо 1, але зазвичай доцільно вибирати 2 або 3. Кожен фактор – це ім’я користувацької змінної, яке повинне відповідати правилам іменування в GPSS (починатися з літери, не збігатися з ключовими словами).

У полі Expression зазначається цільова функція – вираз мовою PLUS, укладений у круглі дужки, або змінна моделі, в якій задана ця функція. За допомогою перемикача Maximize–Minimize обирається напрям оптимізації.

При виборі опції Generate Run Procedure створюється процедура прогонів моделі, в іншому випадку її доведеться написати вручну. Опція Load F12... дозволяє запускати експеримент натисканням однієї клавіші F12 (еквівалент команди CONDUCT з меню Command).

GPSS World підтримує кілька методологій для пошуку оптимальних параметрів моделі. Основним поняттям є локальна експериментальна область, яку користувач визначає під час налаштування експерименту. Для кожного з обраних факторів (від 1 до 5) задаються два рівні значень (Value 1 і Value 2 на рис. 7.3), що визначають межі локальної області. У межах цієї області й відбувається пошук оптимуму. Визначення меж руху локальної області дозволяє уникнути марних запусків моделі. Інший спосіб – перенаправлення меж, яке задає максимальну кількість експериментів у часі та попереджає нескінченні цикли пошуку.

Метою експерименту є виявлення найкращих умов роботи системи та побудова математичної моделі, що їх описує. У більшості випадків прагнуть зробити так, щоб оптимум потрапив у межі локальної області.

Методика проведення експерименту:

1. Побудова моделей. Результати, отримані в ході симуляції, апроксимуються або лінійною регресійною моделлю, або моделлю другого порядку (з урахуванням взаємодій між факторами).

2. Пошук напряму руху. Початковою точкою є вибрана область факторного простору. Для пошуку використовується метод найшвидшого підйому. Якщо лінійна модель неадекватна, застосовується модель другого порядку.

3. Зсув локальної області. Після обчислення градієнта центр локальної області поступово переміщується в напрямку покращення показника до тих пір, поки не досягнуто межі руху або зменшення кроку. Якщо область

занадто вузька, алгоритм може витратити багато кроків на досягнення оптимуму.

4. Фіксація результатів. Кожне зміщення супроводжується запуском моделі, результати якого заносяться до журналу. Якщо оптимум не знайдено, експеримент може перейти в іншу частину факторного простору (ця можливість регулюється параметрами перенаправлення).

Під час проведення оптимізаційного експерименту слід дотримуватись таких зауважень:

- початкова точка повинна бути максимально близькою до очікуваного оптимуму;
- чим ширша експериментальна область, тим менше кроків знадобиться для пошуку, проте модель може виявитися менш точною;
- слід уважно переглядати згенерований PLUS-код і, за потреби, коригувати його для підвищення ефективності експерименту;
- доцільно проводити попередні експерименти, щоб краще визначити рівні факторів.

Завершення оптимізаційного експерименту відбувається, коли оптимальні умови знайдені або досягнуто межу перенаправлення або межу руху.

Як результат користувач отримує повний журнал результатів кожного підексперименту, що дозволяє простежити переміщення локальних експериментальних областей та зробити висновки про оптимальні налаштування системи.

7.4. Проведення експериментів користувача в GPSS World

GPSS World є дуже гнучким і потужним інструментом для проведення користувацьких експериментів, оскільки за допомогою вбудованої мови PLUS можна задати виконання різноманітних, досить складних дій. Крім

того, під час виконання експериментів дозволяється викликати будь-які команди GPSS із процедур, написаних мовою PLUS. Підпрограми на мові PLUS поділяються на процедури (у заголовку використовується ключове слово PROCEDURE) і експерименти (EXPERIMENT). Ці підпрограми повинні міститися у файлі моделі.

Різниця між цими двома видами підпрограм полягає в такому.

Процедури за допомогою оператора RETURN можуть повертати значення, яке можна переглянути за допомогою команди SHOW, або використати в інших підпрограмах. Для цього спочатку потрібно відтранслювати модель разом із процедурою або лише саму процедуру, якщо модель не використовується. При цьому процедура реєструється. Потім у меню потрібно обрати пункт Command/SHOW і вказати там назву процедури, зазначивши в дужках потрібні параметри, або нічого не вказуючи (якщо параметрів немає). Після цього процедура виконується і видає результат.

Експеримент відрізняється від процедури тим, що його може викликати (виконати) тільки користувач. Для цього потрібно обрати пункт меню Command/CONDUCT і написати там назву експерименту, вказавши в круглих дужках параметри (якщо вони є), або нічого не вказувати, якщо параметрів немає. Ключовим моментом є те, що за допомогою бібліотечної процедури DoCommand можна викликати будь-яку команду GPSS, що дозволяє багаторазово виконувати прогони моделі та в кожному прогоні отримувати доступ до поточних результатів.

З підпрограми експерименту можуть викликатися процедури, які в свою чергу можуть викликати інші процедури і т.д. Причому з будь-якої процедури, на якій би глибині вкладеності вона не була, можна викликати процедуру DoCommand, наприклад:

DoCommand("START 1000");

Тому структура виконуваного коду експерименту може бути будь-якої складності. Особливість функції DoCommand полягає в тому, що її аргумент компілюється в глобальній області. Це означає, що локальні змінні не можуть

бути використані безпосередньо. Натомість, якщо необхідно передати локальні значення, слід попередньо сформувати рядок команди засобами роботи з рядками PLUS. Таким чином, користувацький експеримент поєднує в собі набір процедур мовою PLUS, де одна з процедур має в заголовку ключове слово EXPERIMENT. Виконати такий експеримент можна за допомогою команди CONDUCT. Різні експерименти можуть мати спільні викликані процедури. У багатократних експериментах зручно використовувати окрему процедуру PLUS для підготовки кожного прогона. Вона ініціалізує генератори випадкових чисел та формує керуючі команди симуляції. Далі високорівневий експеримент PLUS викликає цю процедуру для кожного окремого прогона.

Користувач може створювати власні списки команд у Include-файлах або вводити їх вручну. Головне – забезпечити занесення результатів кожного прогона експерименту до однієї чи кількох матриць результатів.

У користувацькому експерименті для задання послідовності команд і вихідних даних кожного прогона моделі зручно використовувати PLUS-процедуру. Така процедура може ініціалізувати генератори випадкових чисел, виконувати команди і керувати імітацією.

Контрольні запитання

1. Які види експериментів можна проводити за допомогою GPSS World?
2. Опишіть особливості проведення відсіювальних експериментів у GPSS World.
3. Опишіть особливості проведення оптимізаційних експериментів у GPSS World.
4. Для чого і яким чином організуються користувацькі експерименти в GPSS World?
5. Які основні проблеми виникають при проведенні імітаційних експериментів у GPSS World?

6. Як у системі GPSS World відбувається визначення експерименту?

7. Для чого необхідно враховувати перехідний режим роботи моделі?

8. Як визначити кількість попередніх прогонів моделі, якщо метою моделювання є оцінка ймовірності настання деякої події?

9. Як визначити кількість попередніх прогонів моделі, якщо метою моделювання є оцінка середнього значення випадкової величини?

10. Як визначити кількість попередніх прогонів моделі, якщо критерієм оцінки є вартісна характеристика?

11. Опишіть головні дії під час проведення відсіювальних експериментів у GPSS World.

12. Прокоментуйте призначення дій, які виконуються в наведеному в даному розділі фрагменті тексту моделі при проведенні дисперсійного аналізу в GPSS World.

13. Опишіть головні дії під час проведення оптимізаційних експериментів у GPSS World.

8. РОЗРОБКА ТА МОДЕЛЮВАННЯ ІМІТАЦІЙНИХ МОДЕЛЕЙ У СИСТЕМІ GPSS WORLD

8.1. Застосування текстових об’єктів і потоків даних

На відміну від середовищ моделювання Powersim та AnyLogic, система GPSS World не підтримує безпосередню графічну реалізацію математичної моделі. Тобто вона не містить вбудованих архетипів, які б дозволяли відображати зв’язки між змінними у вигляді графічних діаграм. Це ускладнює процес відлагодження моделей у порівнянні з іншими імітаційними системами, що мають розвинуті засоби візуалізації. Водночас GPSS World має середовище моделювання (GPSS World Simulation Environment), яке дає змогу реалізовувати моделі програмно, а потім візуалізувати результати експериментів.

Одними з важливих інструментів GPSS є блоки для роботи з потоками даних (Data Streams, DS). Потоки даних можуть використовуватись:

- для читання і запису текстових файлів на диску;
- для тимчасового збереження даних у пам’яті під час роботи моделі.

Кожен потік даних ідентифікується числовим номером (ID), який задається користувачем. Цей номер є додатним цілим числом. GPSS дозволяє працювати одночасно з кількома DS.

Основні блоки роботи з потоками даних

- OPEN – створює новий потік даних (DS). Синтаксис:

OPEN A,B,C

де A – ім’я файлу (якщо файл не існує, він створюється, якщо вказано порожній рядок – створюється DS у пам’яті), B – числовий ідентифікатор DS, C – блок для перенаправлення транзакта у випадку помилки (необов’язковий параметр).

- CLOSE – закриває DS і знищує його. Синтаксис:

CLOSE A,B,C

де A – атрибут транзакта для запису коду помилки (необов'язковий параметр), B – ідентифікатор DS, C – блок, куди перенаправляється транзакт у випадку помилки.

- READ – зчитує текстову строку з потоку. Синтаксис:

READ A,B,C

де A – атрибут транзакта для збереження прочитаного рядка, B – ідентифікатор DS, C – блок для помилок або кінця файлу.

- WRITE – записує рядок у DS. Синтаксис:

WRITE A,B,C,D

де A – значення для запису (число, рядок або системний числовий атрибут – всі не рядкові значення автоматично перетворюються в рядок), B – ідентифікатор DS, C – блок для обробки помилок, D – режим запису (ON – вставка, OFF – заміна; за замовчуванням – вставка).

- SEEK – встановлює вказівник на нову позицію у DS. Синтаксис:

SEEK A,B

де A – номер нового рядка (позиція), B – ідентифікатор DS.

Коди помилок роботи з потоками даних:

- 0 – помилок немає;
- 10 – помилка OPEN (занадто довге ім'я файлу, >200 символів);
- 11 – помилка OPEN (неможливо прочитати зовнішній файл у пам'ять);
- 12 – помилка OPEN (недостатньо пам'яті для завантаження файлу);
- 21 – помилка READ (недостатньо пам'яті);
- 22 – помилка READ (потік не відкрито);
- 31 – помилка WRITE (недостатньо пам'яті);
- 32 – помилка WRITE (потік не відкрито);
- 41 – помилка CLOSE (не вдалося записати файл на диск);

- 43 – помилка CLOSE (потік не відкрито);
- 51 – помилка SEEK (потік не відкрито).

Будь-яка модель у GPSS World розглядається як єдиний об’єкт «Модель», що складається з набору операторів і може містити будь-яку кількість додаткових текстових об’єктів. Текстові об’єкти, у свою чергу, є файлами, які зберігають фрагменти операторів або процедур моделі. Важливою перевагою GPSS World є те, що текстові об’єкти можуть використовуватися разом із потоками даних. Це дозволяє:

- зчитувати дані з файлів для використання в процесі моделювання;
- записувати результати експериментів у файли для подальшої обробки;
- створювати окремі файли з результатами, які можуть бути використані іншими системами аналізу (наприклад, Excel, MATLAB, Python).

Для підключення текстових об’єктів використовується команда INCLUDE. Синтаксис:

INCLUDE A

де A – специфікація файла у вигляді рядка (String), що може містити як повний шлях до файла, так і лише його назву.

Використання команди INCLUDE дає змогу:

- підключати стандартні бібліотеки блоків;
- використовувати готові модулі для повторного застосування в різних моделях;
- спрощувати структуру великих імітаційних проектів шляхом поділу їх на окремі частини.

Команда INCLUDE є термінальною командою. Під час трансляції вона буквально замінюється вмістом зазначеного файлу. Саме тому її слід

розташовувати в тій частині моделі, де передбачається використання операторів, описаних у зовнішньому файлі.

Кожен файл, підключений командою INCLUDE, отримує унікальний номер (ціле число), який призначає транслятор. Номер 0 завжди закріплений за основним об'єктом «Модель». Якщо один і той самий файл підключається кілька разів, транслятор усе одно створює окремі, незалежні набори блоків з різними номерами.

Система дозволяє мати до п'яти рівнів вкладеності (тобто файл може підключати інший файл, і так далі, але не більше ніж у 5 рівнів). Але існує обмеження – команда INCLUDE не може використовуватись усередині Plus-процедур.

Для зручності оператор INCLUDE можна закріпити за функціональною клавішею. У такому випадку натисканням клавіші користувач передає об'єкту «Процес моделювання» цілий набір команд або Plus-операторів із зовнішнього файлу.

Таким чином, текстові об'єкти в поєднанні з потоками даних забезпечують гнучкість та модульність у побудові імітаційних моделей. Вони дозволяють організувати моделі у вигляді набору керованих компонентів, що значно спрощує повторне використання коду та інтеграцію з іншими програмними засобами.

Приклад використання потоків даних. У серверному центрі клієнтські запити надходять у середньому кожні 6 хвилин. Системний адміністратор опрацьовує кожен запит у середньому протягом 3 хвилин. Час надходження запитів і час їх обслуговування описується експоненційним законом розподілу. Необхідно промоделювати роботу сервісної системи протягом 8-годинного робочого дня (480 хвилин).

Час очікування запитів у черзі записується у файл на диск для подальшого аналізу та оптимізації роботи інфраструктури.

Для отримання часу очікування використовується системний числовий атрибут M1 – час транзиту, він повертає різницю між абсолютним системним

часом (значенням глобального годинника моделі) та моментом «позначки» (mark time), який був зафіксований у транзакті раніше. M1 дозволяє визначити, скільки часу транзакт перебуває в системі від моменту його створення або від спеціально заданого часу позначки, його зручно використовувати для:

- вимірювання часу очікування в черзі;
- визначення загального часу перебування транзакта в системі;
- збору статистики для подальшого аналізу продуктивності моделі.

Текст моделі:

```
      ; створення файлу і потоку даних
GENERATE „„1
OPEN ("MY.TXT"),1,ERROR
TERMINATE
      ; прихід і обслуговування відвідувачів
GENERATE (EXPONENTIAL(1,0,6))
QUEUE OCH
SEIZE CAN
DEPART OCH
WRITE M1,1,ERROR  ; запис часу перебування в черзі у файл
ADVANCE (EXPONENTIAL(1,0,3))
RELEASE CAN
TERMINATE
      ; таймер
GENERATE 480
ERROR CLOSE OSH,1
SAVEVALUE 1,P$OSH ; збереження коду помилки роботи з DS
TERMINATE 1
START 1
```

Фрагмент змісту згенерованого файла MY.TXT (за замовчанням знаходиться в каталозі з моделлю):

```
...
5.205973301987172
14.27608565852194
0
0
1.564876020716369
4.595756966475307
0
0
...
```

8.2. Розробка моделі в GPSS World

Ефективне використання системи GPSS World передбачає виконання ряду етапів:

1. Постановка задачі.

2. Виявлення основних особливостей.

3. Створення імітаційної моделі процесу.

4. Подання імітаційної моделі в системі GPSSW.

5. Моделювання системи.

Розглянемо розробку моделі на прикладі.

Постановка задачі.

Система обробки пакетів мережевого трафіку забезпечує прийом і маршрутизацію даних, що надходять з частотою 100 тис. пакетів/с, і складається з чотирьох серверів, організованих у конвеєр.

Пакети з рівною ймовірністю спрямовуються у вхідний буфер одного з двох серверів першого сегмента конвеєра. Ємність буфера кожного сервера становить 10 пакетів. Час обробки одного пакета на сервері першого сегмента дорівнює 20 ± 4 мкс. Оброблені пакети з рівною ймовірністю передаються у вхідний буфер одного з двох серверів другого сегмента конвеєра ємністю 5 пакетів, де їхня обробка триває 12 ± 3 мкс.

У процесі обробки виникають збої серверів:

- у серверах першого сегмента з імовірністю 3 %,
- у серверах другого сегмента з імовірністю 7 %.

У разі збою пакет повторно розміщується в буфер сервера і обробка повторюється з початку.

Виконати моделювання роботи системи маршрутизації трафіку протягом 10 мс.

Виявлення основних особливостей.

Задано систему обробки мережевих запитів від віддалених клієнтів, що складається з:

- вхідного буфера даних системи, в який надходять запити;
- чотирьох серверів, об’єднаних у конвеєрну схему.

Конвеєр, у свою чергу, розбитий на два сегменти, кожен з яких містить два сервери:

- у першому сегменті виконується попередня обробка запитів (наприклад, розбір протоколу та автентифікація користувача);
- у другому сегменті – глибока обробка (наприклад, виконання запиту до бази даних або генерація відповіді).

Згідно завдання структурна схема має вигляд, який наведено на рис. 8.1.

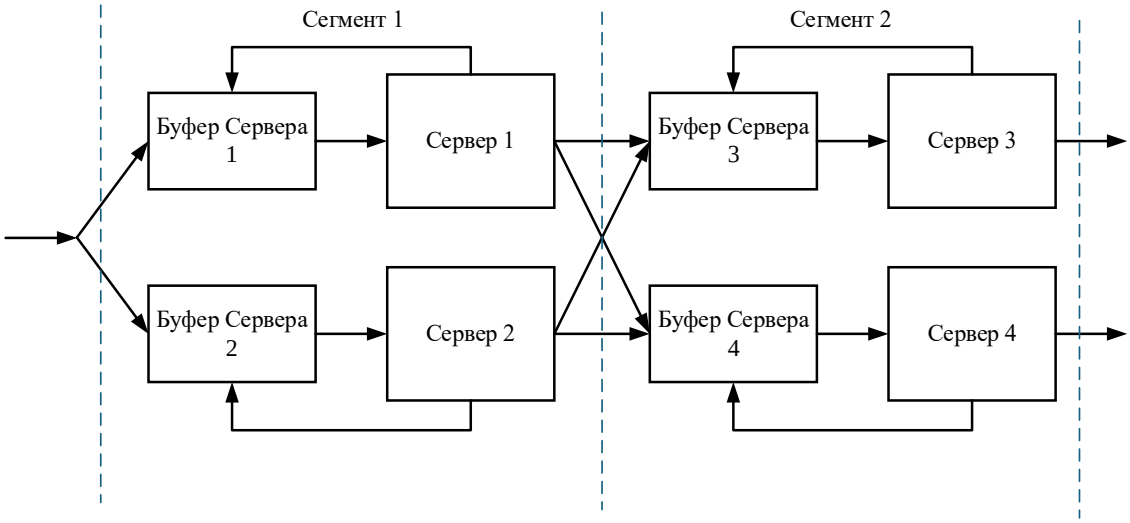


Рис. 8.1 – Структурна схема моделі

Відповідно до наведеної структурної схеми виконується побудова системи у вигляді СМО. Під час розв’язання задач моделювання за допомогою СМО аналіз пов’язується з дослідженням процесу проходження заявок через елементи системи. Заявки розглядаються як випадкові процеси, для яких зазвичай відомі лише закони розподілу та їх числові характеристики – СМО має імовірнісно-статистичний характер.

Пристрої, які здійснюють обслуговування заявок – канали, вони є статичними об’єктами моделі, тоді як заявки виступають динамічними об’єктами. У процесі функціонування СМО можуть виникати черги.

Виконується конкретизація умов та будується СМО, що описує структурну схему.

Вхідні пакети:

- надходять із частотою 100 000 пакетів/с (тобто один пакет кожні 10 мкс);
- усі пакети мають однаковий пріоритет;
- пакети надходять у модель з рівною ймовірністю.

Структура системи:

- перший сегмент конвеєра – два сервери (SERV1, SERV2) з буферами QUE1 та QUE2 (ємність 10 пакетів кожен);
- другий сегмент конвеєра – два сервери (SERV3, SERV4) з буферами QUE3 та QUE4 (ємність 5 пакетів кожен).

Параметри обслуговування:

- час обробки пакета в серверах першого сегмента: 20 ± 4 мкс;
- час обробки пакета в серверах другого сегмента: 12 ± 3 мкс.

Ймовірності збоїв:

- у серверах першого сегмента – 3 %;
- у серверах другого сегмента – 7 %.

Правила маршрутизації:

- пакети з рівною ймовірністю надходять у буфер одного з двох серверів першого сегмента (QUE1 або QUE2);
- після обробки пакети з рівною ймовірністю розподіляються між серверами другого сегмента (QUE3 або QUE4);
- при збої пакет залишається в буфері відповідного сервера, а його обробка повторюється з самого початку.

Моделювання:

- тривалість – 10 мс;
- одиниця модельного часу – 1 мкс.

Q-схему СМО наведено на рис. 8.2.

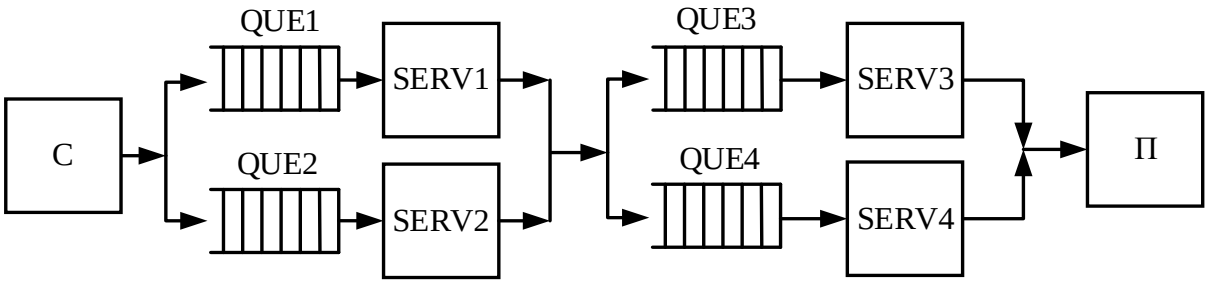


Рисунок 8.2 – Q-схема СМО

У ролі «серверів» системи (SERV1, SERV2, SERV3, SERV4) використовуються одноканальні пристрої. У ролі буферів системи використовуються об’єкти типу черга: QUE1, QUE2, QUE3, QUE4.

Алгоритм моделювання:

1) надходження заявки в модель;

2) рівноймовірний перехід у чергу одного з каналів QUE1 або QUE2;

3) вхід у чергу QUE1 (QUE2);

4) якщо черга заповнена – заявка втрачається;

5) вихід з черги QUE1 (QUE2);

6) займання каналу SERV1 (SERV2) для обробки;

7) обробка за час 20 ± 4 мкс;

8) звільнення каналу SERV1 (SERV2);

9) перевірка умови виникнення збою з ймовірністю 3 %, якщо збій мав місце – перехід на п.3;

10) рівноймовірний перехід у чергу одного з каналів QUE3 або QUE4;

11) вхід у чергу QUE3 (QUE4);

12) якщо черга заповнена – заявка втрачається;

13) вихід з черги QUE3 (QUE4);

14) займання каналу SERV3 (SERV4) для обробки;

15) обробка за час 12 ± 3 мкс;

16) звільнення каналу SERV3 (SERV4);

17) перевірка умови виникнення збою з ймовірністю 7 %, якщо збій мав місце – перехід на п.11;

18) видалення заявки з СМО.

Текст розробленої програми моделі:

```
QUE1 EQU 1 ; символьне і числове ім'я буфера сервера 1
QUE2 EQU 2 ; символьне і числове ім'я буфера сервера 2
QUE3 EQU 3 ; символьне і числове ім'я буфера сервера 3
QUE4 EQU 4 ; символьне і числове ім'я буфера сервера 4
SERV1 EQU 5 ; символьне і числове ім'я сервера 1
SERV2 EQU 6 ; символьне і числове ім'я сервера 2
SERV3 EQU 7 ; символьне і числове ім'я сервера 3
SERV4 EQU 8 ; символьне і числове ім'я сервера 4

GENERATE 10 ; генерація транзактів кожні 10 мкс
TRANSFER .5,LQUE1,LQUE2 ; передача транзакта з ймовірністю 0.5
                        ; на мітки LQUE1, LQUE2
LQUE1 TEST L Q$QUE1,10,LERROR ; перевірка на заповнення черги,
                              ; якщо заповнена – перехід транзакта на мітку LERROR
QUEUE QUE1 ; вхід у чергу 1
SEIZE SERV1 ; займання сервера 1
DEPART QUE1 ; вихід з черги 1
ADVANCE 20,4 ; затримка на 16 - 24 мкс - імітація обробки
RELEASE SERV1 ; звільнення сервера 1
SAVEVALUE SUM1+,1 ; підрахунок оброблених транзактів сервером 1
TRANSFER .03,LABEL12, LERR1 ; перехід з ймовірністю 0.03 на мітку
                            ; LERR1, інакше на LABEL12
LERR1 SAVEVALUE SUMAG1+,1 ; підрахунок повторно оброблених транзактів
                          ; сервером 1
TRANSFER ,LQUE1 ; безумовний перехід на LQUE1

LQUE2 TEST L Q$QUE2,10,LERROR
QUEUE QUE2 ; вхід у чергу 2
SEIZE SERV2 ; займання сервера 2
DEPART QUE2 ; вихід з черги 2
ADVANCE 20,4 ; затримка на 16 - 24 мкс - імітація обробки
RELEASE SERV2 ; звільнення сервера 2
SAVEVALUE SUM2+,1 ; підрахунок оброблених транзактів сервером 2
TRANSFER .03,LABEL12, LERR2 ; перехід з ймовірністю 0.03 на мітку
                            ; LERR2, інакше на LABEL12
LERR2 SAVEVALUE SUMAG2+,1 ; підрахунок повторно оброблених транзактів
                          ; сервером 2
TRANSFER ,LQUE2 ; безумовний перехід на LQUE2

LABEL12 TRANSFER .5,LQUE3,LQUE4 ; передача транзакта з ймовірністю 0.5
                                ; на мітки LQUE3,LQUE4

LQUE3 TEST L Q$QUE3,5,LERROR
QUEUE QUE3 ; вхід у чергу 3
SEIZE SERV3 ; займання сервера 3
DEPART QUE3 ; вихід з черги 3
ADVANCE 12,3 ; затримка на 9 - 15 мкс - імітація обробки
RELEASE SERV3 ; звільнення сервера 3
```

```
        SAVEVALUE SUM3+,1 ; підрахунок оброблених транзактів сервером 3
        TRANSFER .07,LAST, LERR3 ; перехід з ймовірністю 0.07 на мітку
                                 ; LERR3, інакше на LAST
LERR3 SAVEVALUE SUMAG3+,1 ; підрахунок повторно оброблених транзактів
                          ; сервером 3
        TRANSFER ,LQUE3 ; безумовний перехід на мітку LQUE3

LQUE4 TEST L Q$QUE4,5,LERROR
        QUEUE QUE4 ; вхід у чергу 4
        SEIZE SERV4 ; займання сервера 4
        DEPART QUE4 ; вихід з черги 4
        ADVANCE 12,3 ; затримка на 9 - 15 мкс - імітація обробки
        RELEASE SERV4 ; звільнення сервера 4
        SAVEVALUE SUM4+,1 ; підрахунок оброблених транзактів сервером 4
        TRANSFER .07,LAST, LERR4 ; перехід з ймовірністю 0.07 на мітку
                                 ; LERR4, інакше на LAST
LERR4 SAVEVALUE SUMAG4+,1 ; підрахунок повторно оброблених транзактів
                          ; сервером 4
        TRANSFER ,LQUE4 ; безумовний перехід на мітку LQUE4

LERROR SAVEVALUE SUMERR+,1 ; підрахунок необроблених транзактів
LAST TERMINATE ; видалення транзактів із системи

        GENERATE 10000 ; генерування транзактів кожні 10000 мкс (10 мс)
        TERMINATE 1 ; видалення транзактів і зменшення лічильника на 1
        START 1 ; встановлення початкового значення лічильника моделювання
```

Фрагмент звіту з результатами моделювання:

```
FACILITY          ENTRIES  UTIL.   AVE. TIME AVAIL. OWNER PEND INTER RETRY DELAY
 SERV1              498    0.991      19.893  1      981    0     0      0      10
 SERV2              484    0.968      19.998  1        0    0     0      0       0
 SERV3              557    0.668      11.985  1      978    0     0      0       0
 SERV4              485    0.581      11.980  1      976    0     0      0       1


QUEUE              MAX CONT. ENTRY ENTRY(0) AVE.CONT. AVE.TIME   AVE.(-0) RETRY
 QUE1               10   10    508      9     5.279    103.926    105.801   0
 QUE2               10    0    484     21     4.920    101.648    106.258   0
 QUE3                3    0    557    271     0.264      4.745      9.242   0
 QUE4                3    1    486    290     0.173      3.555      8.816   0


SAVEVALUE              RETRY       VALUE
 SUM1                    0        497.000
 SUM2                    0        484.000
 SUM3                    0        556.000
 SUM4                    0        484.000
 SUMAG1                  0          9.000
 SUMAG3                  0         51.000
 SUMAG4                  0         34.000
 SUMAG2                  0         14.000
 SUMERR                  0         31.000
```

Згідно отриманого звіту можна побачити, що 31 пакет було відкинуто за причиною заповнення буферів до серверів, повторно довелось обробляти

пакети в зв’язку зі збоями – 9 на першому сервері, 14 – на другому, 51 на третьому і 34 на четвертому. Максимальна кількість пакетів у буферах третього і четвертого серверів дорівнює 3 (середній зміст 0.263 і 0.173 відповідно). Середній зміст буферів першого та другого серверів 5.279 і 4.92 відповідно. Середній час обробки пакетів на серверах 1-4 19.893, 19.998, 11.985 і 11.980 відповідно.

Інший невеликий приклад демонструє використання користувацьких процедур з інтегруванням (у системі GPSS World інтегрування виконується модифікованим методом Рунґе-Кутта-Фехлберга п’ятого порядку зі змінною довжиною кроку):

```
SIMULATE
F_ EQU 0.0001
X_ EQU 50000
Y_ EQU 100
X_ INTEGRATE (Proc1())
Y_ INTEGRATE (Proc2())
GENERATE 30
TERMINATE 1

PROCEDURE Proc1() BEGIN
TEMPORARY T1_;
IF(X_ < 0)THEN X_ = 0;
IF(X_ > 10e50)THEN X_ = 10e50;
T1_ = (0 - F_) # Y_ # X_;
RETURN T1_;
END;

PROCEDURE Proc2() BEGIN
TEMPORARY T2_;
IF(Y_ < 0)THEN Y_ = 0;
IF(Y_ > 10e50)THEN Y_ = 10e50;
T2_=(Y_ # (F_ # X_ - 1)) + 1000;
RETURN T2_;
END;
```

Результат інтегрування наведено на графіку, який можна отримати в системі за допомогою вікна Plot (рис. 8.3).

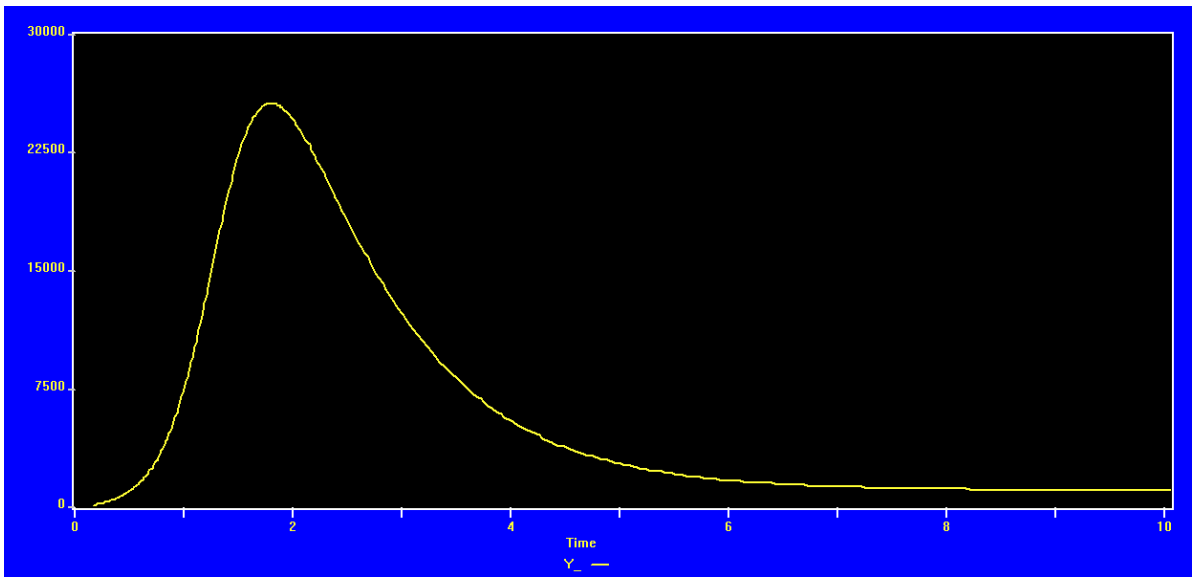


Рисунок 8.3 – Вікно Plot з побудованим графіком

Контрольні запитання

1. Для чого призначені блоки OPEN, CLOSE, READ, WRITE та SEEK?
2. Що є текстовим об’єктом у GPSS World?
3. Опишіть етапи моделювання в системі GPSS World?
4. Яким чином побудувати графіки в GPSS World?
5. Який метод використовується для операції інтегрування в GPSS World?

СПИСОК ЛІТЕРАТУРИ

1. Auping W. L., d'Hont F., Kubli M. D., et al. The Delft Method for System Dynamics. TU Delft OPEN Publishing, 2024. 161 p.

2. Барабаш О. В., Свинчук О. В., Мусієнко А. П. Математичне моделювання та оптимізація процесів і систем. Частина 1 : навч. посіб. Київ : КПІ ім. Ігоря Сікорського, 2023. 160 с.

3. Бахрушин В. Є. Методи аналізу даних: навчальний посібник для студентів. Запоріжжя : КПУ, 2011. 268 с.

4. Великодний С. С. Моделювання систем : конспект лекцій. Одеса : Одеський державний екологічний університет, 2018. 186 с.

5. Гниленко О. Б. Практикум з імітаційного моделювання комп'ютерних систем в середовищі GPSS World. Дніпро : «Ліра», 2023. 120 с.

6. Дриньов Д., Кулинич І., Тимошенко Р. Сучасні програмні засоби для імітаційного моделювання. *Управління змінами та інновації*, (9), 2024. С. 22-24.

7. Жерновий Ю. В. Імітаційне моделювання систем масового обслуговування : Практикум. Львів : Видавничий центр ЛНУ імені Івана Франка, 2007. 307 с.

8. Жерновий Ю. В. Марковські моделі масового обслуговування : Тексти лекцій. Львів : Видавничий центр ЛНУ імені Івана Франка, 2004. 154 с.

9. Задачин В. М., Конюшенко І. Г. Моделювання систем : конспект лекцій. Харків : Вид. ХНЕУ, 2010. 268 с.

10. Зінов'єва О. Г., Шаров С. В., Паламарчук І. П. Імітаційне моделювання та моделювання систем : навчальний посібник. Запоріжжя : ФОП Однорог Т. В., 2025. 203 с.

11. Кравець І. О. Імітаційне моделювання : навчальний посібник до виконання практичних робіт із дисциплін «Моделювання систем» та «Ситуаційні моделі». Миколаїв : Вид-во ЧДУ ім. Петра Могили, 2010. 108 с.

12. Кравченко І. В., Микитенко В. І., Тимчик Г. С. Комп’ютерне моделювання : системи і процеси : підручник. Київ : КПІ ім. Ігоря Сікорського, 2022. 215 с.

13. Махней О. В. Лабораторний практикум з імітаційного моделювання у GPSS. Ч. 1 : методичні рекомендації до проведення лабораторних занять. Івано-Франківськ : Голіней, 2020. 40 с.

14. Моделювання систем в середовищі GPSS World : навч. посіб. / Я. І. Соколовський [та ін.]. Львів : «Новий Світ – 2000», 2020. 288 с.

15. Павленко П. М., Філоненко С. Ф., Чередніков О. М., Трейтяк В. В. Математичне моделювання систем і процесів : навч. посіб. Київ : НАУ, 2017. 392 с.

16. Поворознюк А. І., Поворознюк О. А., Панченко В. І., Філатова Г. Є. Основи наукових досліджень : навч. посібник. Харків : НТУ "ХПІ", 2024. 199 с.

17. Ракович Г. М. Застосування засобу імітаційного моделювання GPSS World для рішення задач планування та обробки експерименту. *Сучасні проблеми моделювання*. 2014. Вип. 2. С. 108-118.

18. Теорія планування експериментів : Виконання розрахунково-графічної роботи : навч. посіб. / С. М. Лапач; КПІ ім. Ігоря Сікорського. Київ : КПІ ім. Ігоря Сікорського, 2020. 86 с.

19. Уривський Л. О., Мошинська А. В., Осипчук С. О. Імітаційне моделювання систем і процесів у телекомунікаціях. Навчальний посібник. Київ : КПІ ім. Ігоря Сікорського, 2022. 202 с.

Навчальне видання

КОЛОМІЙЦЕВ Олексій Володимирович
ПАНЧЕНКО Володимир Іванович
НОСКОВ Валентин Іванович
ЛИСИЦЯ Дмитро Олександрович

ПРОБЛЕМИ І МЕТОДИ МАТЕМАТИЧНОГО І КОМП'ЮТЕРНОГО МОДЕЛЮВАННЯ В НАУКОВИХ ДОСЛІДЖЕННЯХ

Навчальний посібник
для студентів спеціальності F7 «Комп'ютерна інженерія»
денної та заочної форм навчання

Відповідальний за випуск проф. Заковоротний О. Ю.

Роботу до видання рекомендував проф. Заполовський М. Й.

В авторській редакції

План 2026 р., поз. 16

Гарнітура Times New Roman. Ум. друк. арк. 11,5.

Видавничий центр НТУ «ХПІ».
Свідоцтво про державну реєстрацію ДК № 5478 від 21.08.2017 р.
61002, м. Харків, вул. Кирпичова, 2.

Електронне видання