МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

О. В. Коломійцев, В. І. Панченко

МЕТОДИЧНІ ВКАЗІВКИ

до виконання практичних робіт
з навчальної дисципліни «Проблеми і методи математичного і комп'ютерного моделювання в наукових дослідженнях»
для студентів денної та заочної форми навчання
за спеціальністю «Комп'ютерна інженерія»

Затверджено
редакційно-видавничою
радою НТУ «ХПІ»,
протокол № 3 від 24.10.2024 р.

Харків
НТУ «ХПІ»
2024

Методичні вказівки до виконання практичних робіт з навчальної дисципліни «Проблеми і методи математичного і комп'ютерного моделювання в наукових дослідженнях» для студентів денної та заочної форми навчання за спеціальністю «Комп'ютерна інженерія» / О. В. Коломійцев, В. І. Панченко. – Харків: НТУ «ХПІ», 2024. – 89 с.

Автори: *Олексій КОЛОМІЙЦЕВ,* д-р техн. наук, професор, Заслужений винахідник України, проф. каф. комп'ютерної інженерії та програмування НТУ «ХПІ»,
Володимир ПАНЧЕНКО, ст. викл. каф. комп'ютерної інженерії та програмування НТУ «ХПІ»

Рецензент: *Андрій ПАШНЄВ*, канд. техн. наук, с.н.с., доц. каф. інформаційних систем та технологій НТУ «ХПІ»

Кафедра комп'ютерної інженерії та програмування

ЗМІСТ

ВСТУП

Методичні вказівки до виконання практичних робіт з навчальної дисципліни «Проблеми і методи математичного і комп'ютерного моделювання в наукових дослідженнях» призначені для формування практичних навичок з моделювання, необхідних для розв'язування складних завдань у різних сферах.

Матеріали практичних робіт охоплюють різні методи та техніки математичного і комп'ютерного моделювання, починаючи від основних принципів класифікації моделей і моделювання, до більш складних методів, таких як моделювання безперервних марковських процесів, статистичне моделювання, планування експериментів і оцінювання характеристик випадкових величин.

Окремо розглядаються питання роботи з програмним забезпеченням, зокрема з системою GPSS World, яка широко використовується для моделювання та симуляції складних систем.

Виконання практичних робіт дозволить студентам здобути необхідні навички для аналізу та оптимізації складних систем, а також для роботи з інструментами математичного та комп'ютерного моделювання, що є важливим для подальшої професійної діяльності в галузі комп'ютерної інженерії та суміжних областях.

1. КЛАСИФІКАЦІЯ МОДЕЛЕЙ І МОДЕЛЮВАННЯ

Мета заняття: набути практичних навичок виконувати класифікацію моделей та моделювання.

1.1. Короткі теоретичні відомості

Модель – це сутність чи об'єкт, який відображає процеси, що протікають у реальних системах, за допомогою подібних за своїми властивостями математичних або натурних засобів.

Моделювання – це засіб пізнання дійсності, який дозволяє досліджувати складні процеси і явища на основі експериментів не з реальною системою, а з її моделлю.

Найбільш актуальні ознаки класифікації моделей [2, 5]:

- за характером модельованої сторони об'єкта, що моделюється:
  - функціональні (кібернетичні);
  - 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.

Функціональна модель 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. При функціональному моделюванні експеримент полягає в спостереженні за виходом модельованого об'єкта при зміні вхідних впливів. За цими даними і 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.

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

Інформаційна модель – 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.

Детерміновані моделі відображають процеси, в яких відсутній випадковий вплив.

Стохастичні моделі відображають імовірнісні процеси і події.

Статичні моделі служать для опису стану об'єкта в будь-який момент часу.

Динамічні моделі відображають поведінку об'єкта в часі.

Дискретні моделі відображають поведінку систем з дискретними станами.

Безперервні моделі представляють системи з безперервними процесами.

Дискретно-безперервні моделі будуються тоді, коли дослідника цікавлять обидва ці типи процесів.

Абстрактні моделі являють собою певні конструкції із загальноприйнятих знаків на матеріальному носії або у вигляді комп'ютерної програми.

Математичне моделювання – це процес встановлення відповідності модельованого об'єкту деякої математичної конструкції, іменованою математичною моделлю, і дослідження цієї моделі, що дозволяє отримати характеристики об'єкта, що моделюється.

Аналітичні моделі – це функціональні співвідношення: системи алгебраїчних, диференціальних, інтегро-диференціальних рівнянь, логічних умов.

Імітаційне моделювання передбачає представлення моделі у вигляді деякого алгоритму (комп'ютерної програми), виконання якого імітує послідовність зміни станів у системі і таким чином являє собою поведінку модельованої системи. Імітаційна модель реалізує часову діаграму функціонування модельованої системи.

Процес створення і випробування таких моделей називається імітаційним моделюванням, а сам алгоритм – імітаційної моделлю.

Імітаційна модель – універсальний засіб дослідження складних систем, що представляє собою логіко-алгоритмічний опис поведінки окремих елементів системи та правил їх взаємодії, що відображають послідовність подій, які виникають у системі, що моделюється.

Статистичне моделювання – метод дослідження складних систем, заснований на описі процесів функціонування окремих елементів в їх взаємозв'язку з метою отримання безлічі часткових результатів, що підлягають обробці методами математичної статистики для отримання кінцевих результатів. В основі статистичного моделювання лежить метод статистичних випробувань Монте-Карло.

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

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

Приклад використання імітаційного моделювання системи, наведеної на рис. 1.1, яка складається з шести блоків, при чому є певна ймовірність виходу з ладу кожного з блоків (вказана всередині блоків на рис. 1.1). Необхідно за допомогою імітаційної моделі знайти ймовірність виходу з ладу всієї системи.

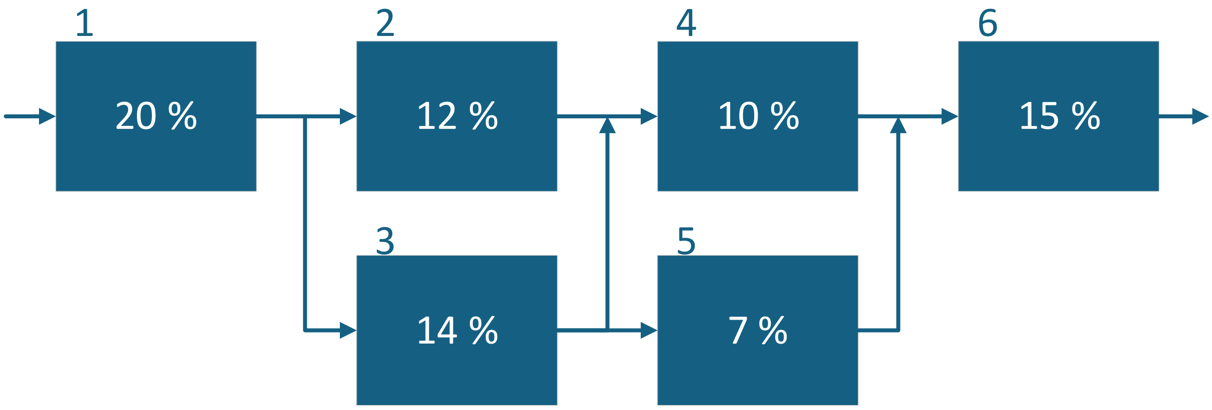

Рисунок 1.1 – Схема з’єднання блоків у системі

Вихід з ладу всієї системи відбудеться у випадку, якщо вийдуть з ладу як мінімум такі блоки:

- 1 або
- 6 або
- 2 та 3 одночасно або
- 4 та 5 одночасно або
- 3 та 4 одночасно.

Для виконання моделювання доцільно скористуватись методом Монте-Карло. Якщо вважати, що ймовірність виходу з ладу певного блоку підпорядковується рівномірному закону розподілу випадкової величини, то при розробці програми моделювання можна використати генератор псевдовипадкових чисел: якщо отримане число менше ніж вказана ймовірність, то вважається, що блок вийшов з ладу ($Б_i = 1$), інакше – $Б_i = 0$. Вихід всієї системи з ладу можна описати як: $Б_1 \vee Б_6 \vee (Б_2 \,\&\, Б_3) \vee (Б_4 \,\&\, Б_5) \vee (Б_3 \,\&\, Б_4)$. При достатній кількості експериментів (кілька тисяч) можна отримати з завданою точністю ймовірність виходу з ладу усієї системи.

1.2. Класифікація моделей (приклади для обговорення)

Класифікуйте модель, відповідь поясніть:

1. Модель системи управління базою даних для банку за ознакою «характер модельованої сторони».

2. Модель для прогнозування відвідуваності вебсайту за характером процесів, що протікають в об’єкті.

3. Модель поштового сервера за характером модельованої сторони об’єкта, що моделюється.

4. Модель багатокористувацької онлайн-гри за характером процесів, що протікають в об’єкті.

5. Моделювання взаємодії між користувачами в соціальній мережі. Який тип моделі?

6. Аналіз алгоритму пошуку в індексі вебсайтів. Який тип моделі?

7. Моделювання захисту від DDoS-атак на вебсайт. Який тип моделі?

8. Модель мережі передачі даних у великій компанії за характером процесів, що протікають в об’єкті.

9. Комп’ютерна модель для управління трафіком у великих містах за способом реалізації моделі.

10. Модель процесу обробки запитів у веб-сервері за характером модельованої сторони об’єкта, що моделюється.

11. Моделювання автоматичного перекладу текстів між мовами. Який тип моделі?

12. Модель поведінки користувача в інтерфейсі програми за характером модельованої сторони об’єкта, що моделюється.

13. Моделювання алгоритмів стискання даних у файлових системах. Який тип моделі?

14. Модель для аналізу продуктивності комп’ютерних мереж за характером процесів, що протікають в об’єкті.

15. Модель для виявлення шахрайства в онлайн-сплатах. Який тип моделі?

16. Моделювання алгоритмів машинного навчання для розпізнавання зображень. Який тип моделі?

17. Моделювання процесу зберігання даних у хмарному середовищі. Який тип моделі?

18. Модель автоматичного тестування програмного забезпечення за способом реалізації моделі.

19. Моделювання процесу оптимізації пошукових систем у мережі Інтернет. Який тип моделі?

20. Моделювання процесу комунікації між мобільними додатками в одній екосистемі. Який тип моделі?

1.3. Індивідуальні завдання

Написати програму для моделювання виходу з ладу системи згідно індивідуального варіанта. Імовірності виходу з ладу кожного блоку наведено на рисунках 1.2-1.9.

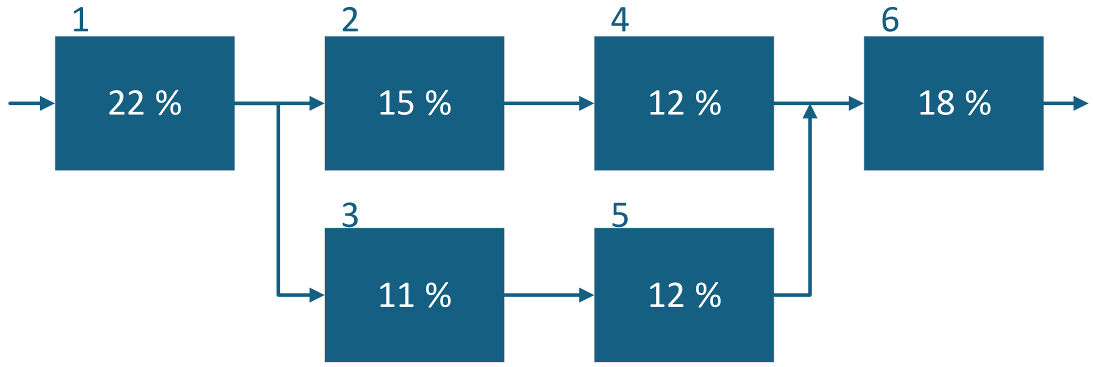

Рисунок 1.2 – Варіант 1

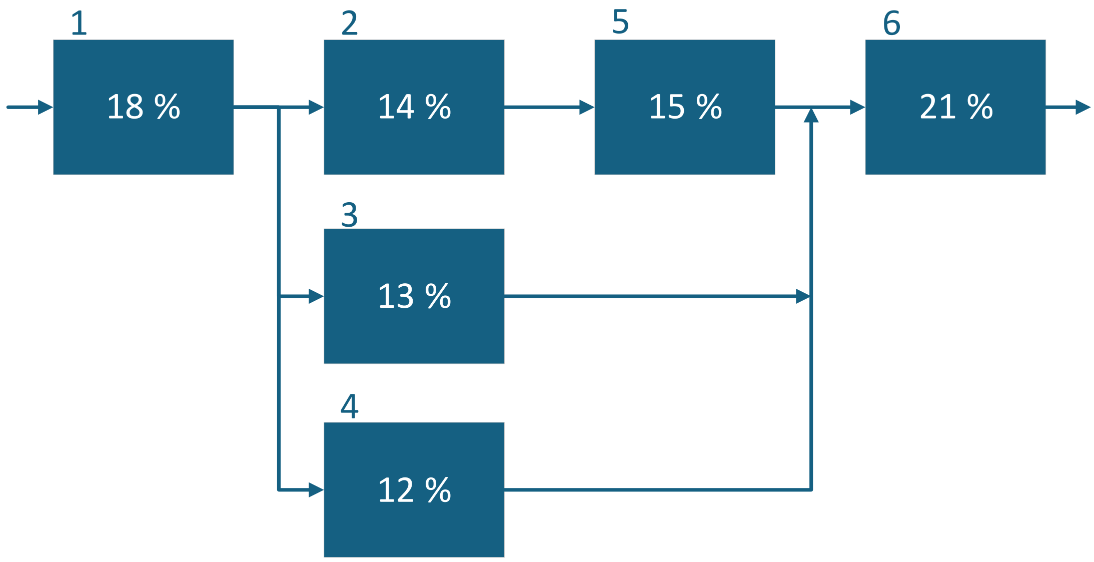

Рисунок 1.3 – Варіант 2

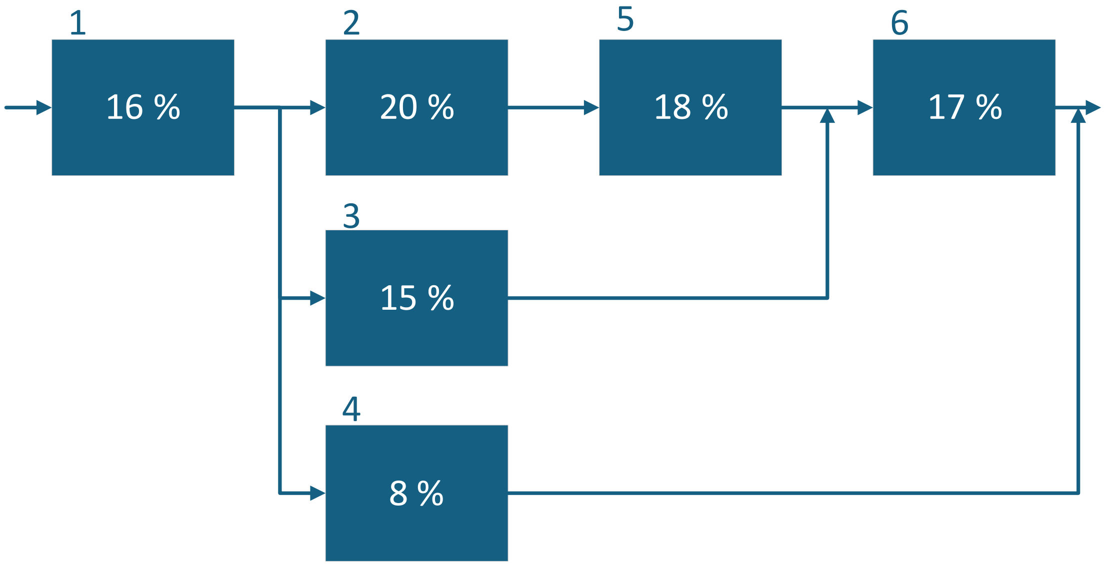

Рисунок 1.4 – Варіант 3

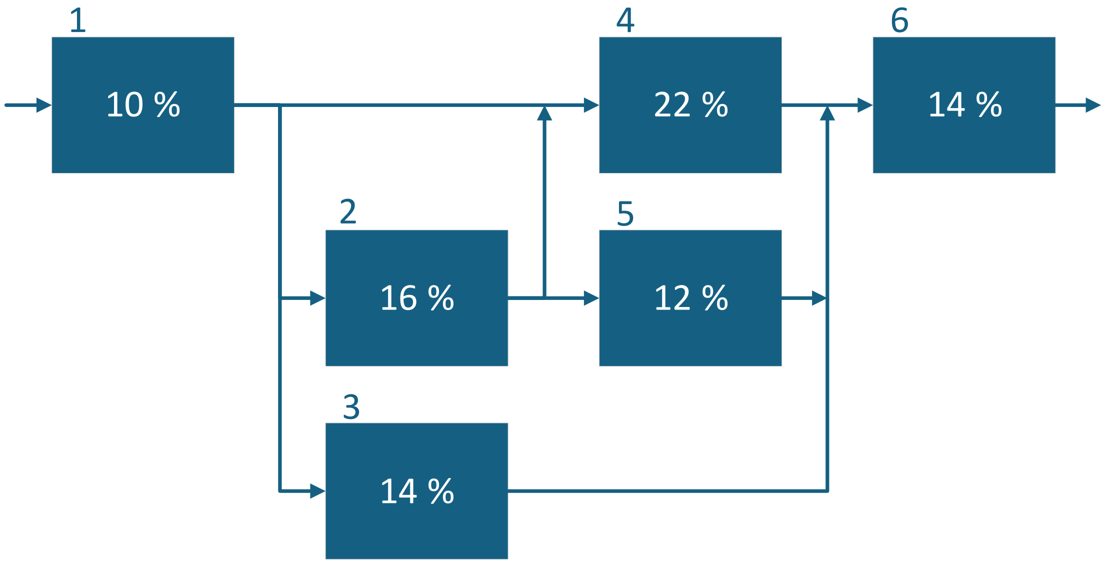

Рисунок 1.5 – Варіант 4

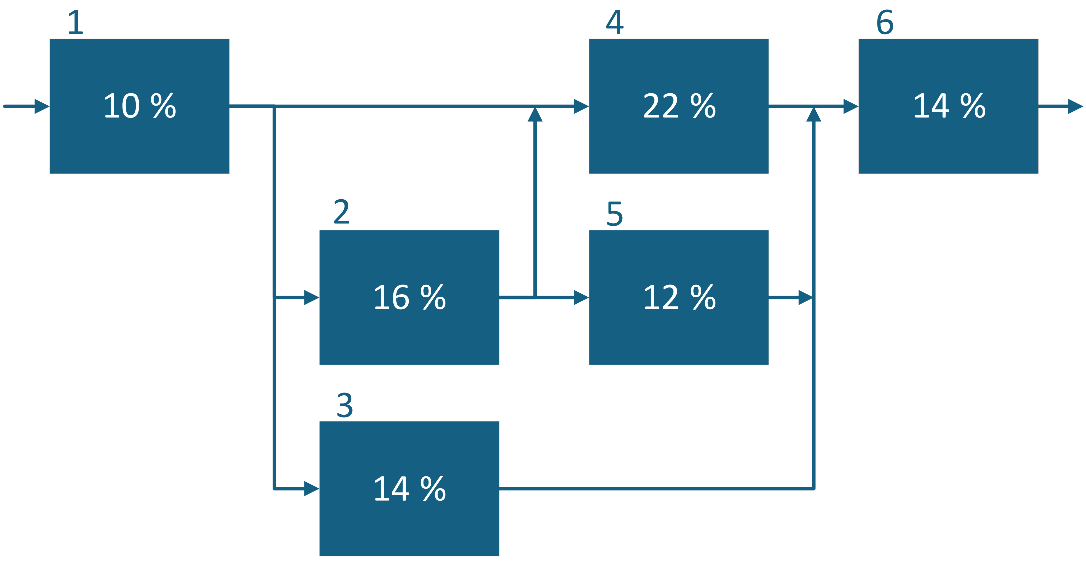

Рисунок 1.6 – Варіант 5

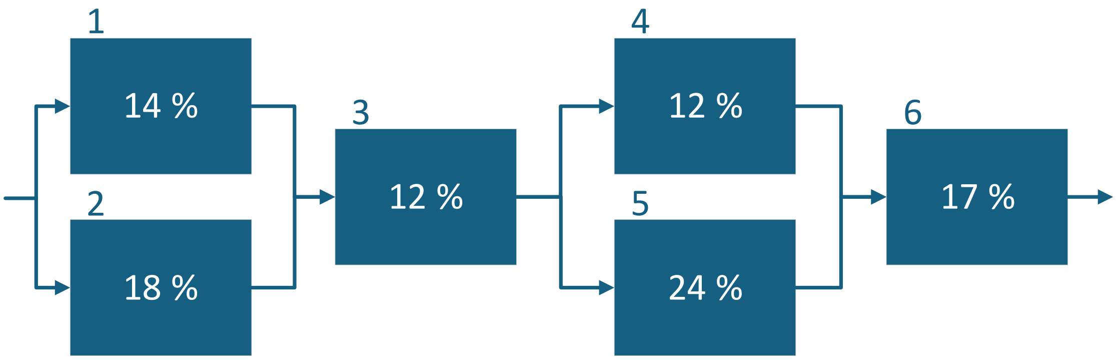

Рисунок 1.7 – Варіант 6

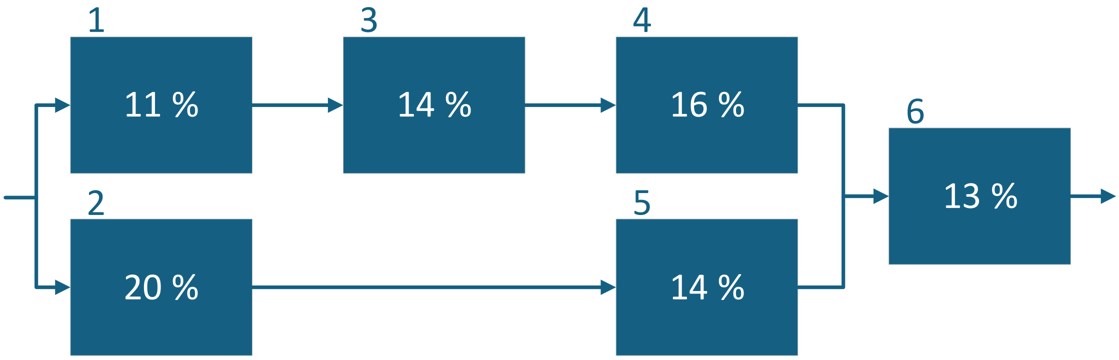

Рисунок 1.8 – Варіант 7

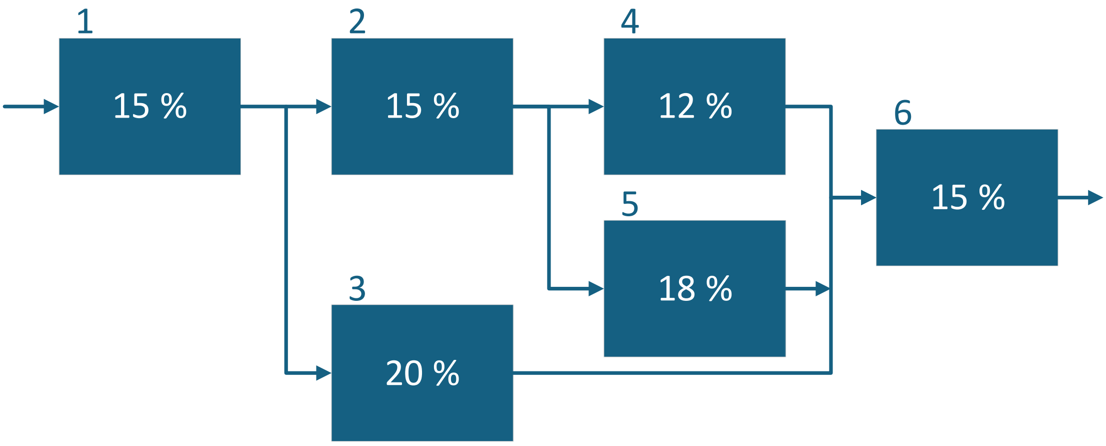

Рисунок 1.9 – Варіант 8

1.4. Контрольні запитання

1. Що таке модель та моделювання?

2. Назвіть ознаки класифікації моделей за характером модельованої сторони об’єкта, що моделюється.

3. Назвіть ознаки класифікації моделей за характером процесів, що протікають в об’єкті.

4. Назвіть ознаки класифікації моделей за способом реалізації моделі.

5. Що означає імітаційне моделювання?

2. МОДЕЛЮВАННЯ БЕЗПЕРЕРВНИХ МАРКОВСЬКИХ ПРОЦЕСІВ

Мета заняття: ознайомитись з принципами моделювання за схемою безперервних марковських процесів, навчитись виконувати моделювання системи масового обслуговування в класі безперервних марковських процесів.

2.1. Короткі теоретичні відомості

Марковським випадковим процесом називається такий випадковий процес, для якого в будь-який моменту часу t_0 ймовірнісні характеристики процесу в майбутньому $(t > t_0)$ залежать лише від його стану в даний момент $(t = t_0)$ і не залежать від того, коли і як система набула цього стану $(t < t_0)$ [1]. Тобто марковський процес є процесом без післядії.

Для аналізу марковських процесів використовуються графи станів системи.

Марковський випадковий процес називається процесом з дискретними станами, якщо всі n можливих станів системи S можна перелічити та пронумерувати $(S_1,\ S_2, \dots, S_n)$, а сам процес полягає в тому, що система S миттєво переходить з одного стану в інший.

Окрім процесів з дискретними станами існують процеси із неперервними станами, коли зміна станів системи S виконується поступово, плавно та неперервно.

За іншою ознакою марковські процеси можна поділити на процеси з неперервним часом – переходи із стану в стан можуть відбуватись у будь-які моменти часу та процеси з дискретним часом – переходи відбуваються тільки в дискретні моменти часу.

У випадку безперервних марківських процесів для визначення ймовірності стану системи для будь-якого моменту часу необхідно використовувати щільність ймовірностей переходів:

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

де $p_{ij}(\Delta t)$ – ймовірність того, що система, що знаходилася в момент часу t у стані S_i за час Δt перейде в стан S_j.

З точністю до нескінченно малих другого порядку:

$$p_{ij}(\Delta t) \approx \lambda_{ij} \cdot \Delta t.$$

Однорідний неперервний марковський процес – щільності ймовірностей переходів λ_{ij} не залежать від часу t (від початку проміжку Δt). Інакше неперервний марковський процес є неоднорідним.

Мета моделювання – визначення ймовірностей станів системи $p_i(t)$ за допомогою інтегрування системи диференціальних рівнянь Колмогорова за наступним алгоритмом:

1. Визначається стан системи та щільність ймовірностей переходів λ_{ij}.
2. Створюється граф станів.
3. Складається система диференціальних рівнянь Колмогорова, число рівнянь в якій дорівнює числу станів, а кожне рівняння формується в такий спосіб:
   - у лівій частині рівняння записується похідна ймовірності i-го стану $\frac{dp_i(t)}{dt}$;
   - у правій частині записується алгебраїчна сума творів $\lambda_{ij}p_j(t)$ і $-\lambda_{ij}p_i(t)$. Число творів визначається кількістю стрілок, пов'язаних з цим станом. Якщо стрілка графа спрямована на цей стан, то відповідний твір має знак плюс, якщо з цього стану – мінус.
4. Визначаються початкові умови та вирішується система диференціальних рівнянь.

Система диференціальних рівнянь Колмогорова має вигляд:

$$\frac{dp_i(t)}{dt} = \sum_{j=1,\, j\neq i}^{n} \lambda_{ji} p_j(t) - \left(\sum_{j=1,\, j\neq i}^{n} \lambda_{ij} \right) p_i(t)$$

Приклад системи диференціальних рівнянь Колмогорова для знаходження ймовірностей станів системи, розмічений граф станів якої наведено на рис. 2.1:

$$\begin{cases} \dfrac{dp_1(t)}{dt} = \lambda_{31}p_3(t) - \lambda_{13}p_1(t) - \lambda_{12}p_1(t) \\ \dfrac{dp_2(t)}{dt} = \lambda_{12}p_1(t) + \lambda_{32}p_3(t) - \lambda_{23}p_2(t) \\ \dfrac{dp_3(t)}{dt} = \lambda_{13}p_1(t) + \lambda_{23}p_2(t) - \lambda_{31}p_3(t) - \lambda_{32}p_3(t) \end{cases}$$

$$p_1(t) + p_2(t) + p_3(t) = 1 .$$

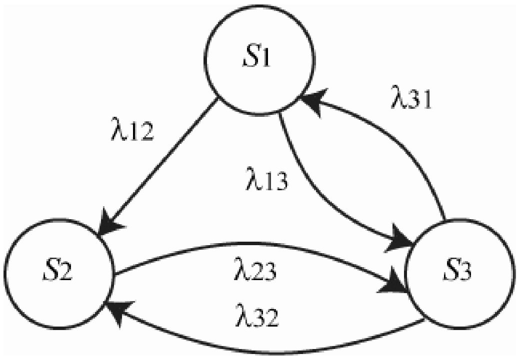

Рисунок 2.1 – Розмічений граф станів системи

Теорема Маркова: якщо для однорідного дискретного марковського процесу з кінцевим чи лічильним числом станів всі $p_{ij} > 0$, то граничні значення $p_j(k)$ існують та їх значення не залежать від обраного початкового стану системи.

Якщо безперервний марковський процес однорідний і з кожного стану можливий перехід за кінцевий час у будь-який інший стан і кількість станів кінцева або лічильна, то граничні значення $p_i(t)$ існують та їх значення не залежать від обраного початкового стану.

Неперервний ланцюг Маркова називається процесом «загибелі та розмноження», якщо її граф станів має вигляд, наведений на рис. 2.2, тобто всі стани можна витягнути в один ланцюг, в якому кожний із середніх станів $(S_2, \dots, S_{n-1})$ пов'язаний прямим та зворотнім зв'язком з кожним із сусідніх станів, а крайні стани (S_1, S_n) – тільки з одним сусіднім станом.

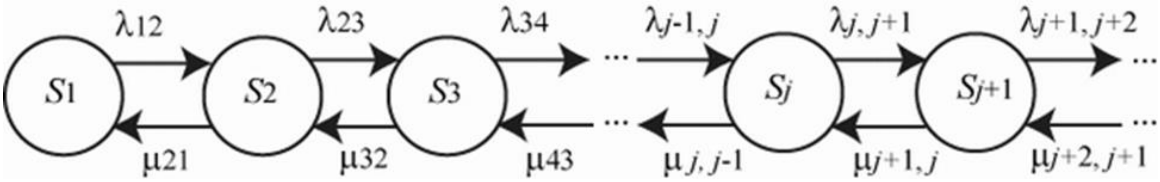

Рисунок 2.2 – Схема «загибелі та розмноження»

Деякі приклади марковських випадкових процесів «загибелі та розмноження»:

- поступлення та витрачання грошей у касі;
- поступлення та витрачання матеріалів на складі;
- обслуговування клієнтів і їхнє перебування;
- процес відмов і відновлення компонентів багатомодульних систем;
- робота технологічних ліній щодо виробництва та ремонту техніки;
- функціонування різних систем масового обслуговування та ін.

Складати рівняння Колмогорова для схеми «загибелі та розмноження» немає необхідності, оскільки структура регулярна (необхідні формули часто наводяться в довідковій літературі). Для схеми на рис. 2.2 формули такі:

$$P_1 = \frac{1}{1 + \frac{\lambda_{12}}{\mu_{21}} + \frac{\lambda_{12}\lambda_{23}}{\mu_{21}\mu_{32}} + \frac{\lambda_{12}\lambda_{23}\lambda_{34}}{\mu_{21}\mu_{32}\mu_{43}} + \frac{\lambda_{12}\lambda_{23}\lambda_{34}}{\mu_{21}\mu_{32}\mu_{43}} + \cdots + \frac{\lambda_{12}\lambda_{23}\ldots\lambda_{n-1,n}}{\mu_{21}\mu_{32}\cdots\mu_{n,n-1}}}$$

$$P_2 = \frac{\lambda_{12}\lambda_{23}}{\mu_{21}\mu_{32}} \cdot P_1; \ldots P_n = \frac{\lambda_{12}\lambda_{23}\ldots\lambda_{n-1,n}}{\mu_{21}\mu_{32}\cdots\mu_{n,n-1}} \cdot P_1.$$

Приклад розрахунку ймовірностей станів системи, яка складається з двох однакових блоків, які працюють паралельно.

У системі можливі три стани:

S_1 – обидва блоки справні;

S_2 – один блок справний, інший у ремонті;

S_3 – обидва блоки несправні і в ремонті.

Припускається, що процеси відмов і відновлення однорідні марковські та одночасний вихід з ладу обох блоків (і одночасне відновлення обох блоків) неможливо. Оскільки блоки однакові, то неважливо, який саме з них несправний у стані S_2. Схема «загибелі та розмноження» для даного прикладу наведена на рис. 2.3. Позначки:

$\lambda_{12}, \lambda_{23}$ – інтенсивності потоків відмов;

μ_{21}, μ_{32} – інтенсивності потоків відновлення.

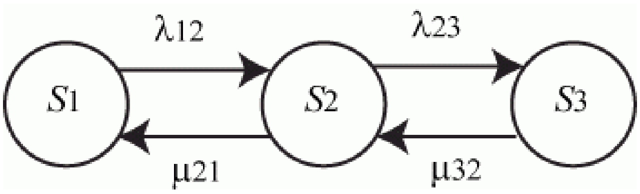

Рисунок 2.3 – Схема «загибелі та розмноження» для прикладу

Припустимо, що час безвідмовної роботи кожного блоку складає $\bar{t} =$ 10 діб, середній час відновлення одного блоку $\bar{t}_{\text{в}} = 0.1$ доби.

Інтенсивність відмов одного блоку становить $\lambda = \frac{1}{\bar{t}} = \frac{1}{10} = 0.1 \frac{1}{\text{доба}}$, інтенсивність відновлення одного блоку – $\mu = \frac{1}{\bar{t}_{\text{в}}} = \frac{1}{0.1} = 10 \frac{1}{\text{доба}}$.

У стані S_1 працюють обидва блоки, тобто:

$$\lambda_{12} = 2\lambda = 2 \cdot 0.1 = 0.2 \frac{1}{\text{доба}}.$$

У стані S_2 працює один блок, тобто:

$$\lambda_{23} = \lambda = 0.1 \frac{1}{\text{доба}}.$$

У стані S_2 відновлюється один блок, тобто:

$$\mu_{21} = \mu = 10 \frac{1}{\text{доба}}.$$

У стані S_3 відновлюються обидва блоки, тобто:

$$\mu_{32} = 2\mu = 2 \cdot 10 = 20 \frac{1}{\text{доба}}.$$

Використовуючи формули для схеми «загибелі та розмноження» можна отримати:

- ймовірність стану S_1, коли обидва блоки справні:

$$P_1 = \frac{1}{1 + \frac{\lambda_{12}}{\mu_{21}} + \frac{\lambda_{12}\lambda_{23}}{\mu_{21}\mu_{32}}} = \frac{1}{1 + \frac{0.2}{10} + \frac{0.2 \cdot 0.1}{10 \cdot 20}} = 0.98;$$

- ймовірність стану S_2 (працює один блок):

$$P_2 = \frac{\lambda_{12}}{\mu_{21}} P_1 = 0.02 \cdot 0.98 = 0.0196;$$

- ймовірність стану S_3 (останній стан, можна скористуватись формулою суми ймовірностей):

$$P_3 = 1 - (P_1 + P_2) = 1 - (0.98 + 0.196) = 0.0004.$$

Системами масового обслуговування (СМО) [7] називають системи, які складаються з будь-якої кількості каналів обслуговування, які призначені для обслуговування будь-якого потоку заявок, що надходять на систему у випадкові моменти часу (рис. 2.4).

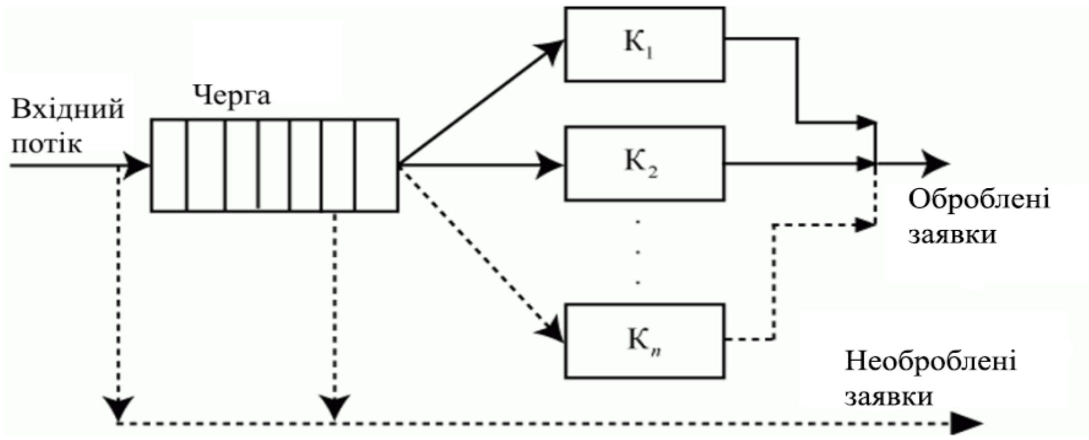

Рисунок 2.4 – Система масового обслуговування

Елементами СМО є:

- вхідний потік заявок на обслуговування;
- прилади (канали) обслуговування;
- черга заявок, які очікують на обслуговування;
- вихідний потік оброблених заявок;
- потік необроблених заявок;

- черга вільних каналів (для багатоканальних СМО).

Операція в СМО – комплекс заходів щодо обслуговування вхідного потоку заявок на інтервалі часу T. Залежно від типу системи показниками ефективності СМО є:

- Для СМО з відмовами:
  - абсолютна пропускна здатність Q – середнє число заявок, що обслуговується системою за час T;
  - відносна пропускна здатність q – середня доля заявок, що обслуговується системою (відношення середнього числа оброблених заявок до середнього числа отриманих за час T);
  - середнє число зайнятих каналів $\overline{n_з}$;
  - коефіцієнт зайнятості каналів $К_з = \frac{\overline{n_з}}{n}$, де n – число каналів у системі;
  - коефіцієнт простоювання каналів $К_п = 1 - К_з$.
- Для СМО з необмеженим очікуванням:
  - середнє число заявок у черзі $\overline{l_{оч}}$;
  - середнє число заявок у системі (у черзі і на обслуговуванні $\overline{l_с}$;
  - середній час очікування заявки в черзі $\overline{t_{оч}}$;
  - середній час перебування заявки в системі $\overline{t_с}$;
  - коефіцієнти простоювання і зайнятості каналів $К_п$ і $К_з$;
  - середнє число вільних і зайнятих каналів $\overline{n_в}$ і $\overline{n_з}$.
- Для СМО змішаного типу використовуються обидві групи показників.

Будь-який із наведених показників (або сукупність показників) може бути обраний як критерій ефективності.

Аналітичною моделлю СМО є сукупність рівнянь або формул, що дозволяють визначати ймовірності станів системи в процесі її функціонування та розраховувати показники ефективності за відомими характеристиками вхідного потоку та каналів обслуговування.

Для забезпечення універсального підходу в позначенні різних типів СМО розроблено класифікацію «Символіка Кендала–Башаріна».

Символіка складається з шести позицій, що розділяються слешами:

1/2/3/4/5/6

1. Тип потоку, що надходить:
   - M – найпростіший потік;
   - Mt – пуассонівський потік із змінним параметром (залежить від часу);
   - Mr – пуассонівський потік з умовним параметром;
   - Mi – примітивний потік;
   - D – детермінований (невипадковий) потік;
   - En – потік Ерланга n-ого порядку;
   - Ge – довільний потік.
2. Закон розподілу часу обслуговування виклику:
   - M – експоненціальний;
   - D – детермінований;
   - G – довільний.
3. Структура СМО:
   - V – число каналів;
   - G – неповнодоступні канали обслуговування (існує алгоритм, що визначає, які канали доступні яким заявкам). Якщо не вказано, то усі канали обслуговування доступні усім викликам;
   - LS – багатофазна система, якщо не вказано, то це – однофазна система, де заявка проходить тільки одну фазу обслуговування в каналі.
4. Дисципліна або спосіб обслуговування:
   - LL – без втрат;
   - L – з втратами;

- W – з очікуванням (чергою);
- R – з повторенням;
- WL – з умовними втратами (комбінований).

5. Тип черги:
   - I – індивідуальна, якщо не вказано – загальна черга до усіх каналів обслуговування;
   - SP – рівноімовірна;
   - FF – FIFO;
   - LF – стекова LIFO;
   - PR – з пріоритетом:
     - PRR – відносний – заявка чекає звільнення каналу,
     - PRA – абсолютний – заявка перериває обслуговування і займає канал.
6. Спосіб заняття каналу:
   - S – послідовний;
   - R – випадковий.

2.2. Індивідуальні завдання

Для вказаної ситуації визначити за класифікацією «Символіка Кендала–Башаріна» та побудувати модель СМО, навести формули розрахунку показників та написати програму моделювання обробки заявок. Формули та тип СМО визначити за допомогою довідкової літератури.

1. Моделювання обробки запитів в інформаційній службі: модель прогнозування часу обробки запитів користувачів за допомогою системи підтримки користувачів Help Desk.
2. Аналіз навантаження на сервери при масштабуванні хмарної інфраструктури: модель оцінки ефективності масштабування серверів у хмарних системах.

3. Моделювання черг у багатозадачних операційних системах: модель оцінки часу обробки багатозадачних запитів в операційних системах.

4. Оптимізація розподілу ресурсів між користувачами в багатокористувацьких системах: модель рівномірного розподілу ресурсів між користувачами на основі їхніх запитів.

5. Аналіз пропускної здатності мережі в умовах високого навантаження: модель прогнозування затримок і втрат пакетів у комп'ютерних мережах.

6. Оцінка продуктивності бази даних при збільшенні кількості запитів: модель прогнозування швидкості обробки запитів до бази даних.

7. Аналіз використання мережевих каналів в умовах трафіку з високими піковими навантаженнями: модель прогнозування поведінки мережі при високих пікових навантаженнях.

8. Аналіз ефективності роботи чат-ботів у системах підтримки користувачів: модель прогнозування часу відповіді та точності чат-ботів у різних сценаріях.

2.3. Контрольні запитання

1. Що таке марковський випадковий процес?
2. Поясніть використання системи диференціальних рівнянь Колмогорова.
3. Коли використовується схема «загибелі та розмноження»?
4. Що таке СМО?
5. Як можна класифікувати СМО?

3. СТАТИСТИЧНЕ МОДЕЛЮВАННЯ

Мета заняття: ознайомитись та набути практичні навички при розв’язанні детермінованих задач за допомогою моделювання рівномірно розподіленої випадкової величини, випадкової величини з довільним законом розподілу, одиничних подій, повної групи несумісних подій, спільних незалежних подій та спільних залежних подій.

3.1. Короткі теоретичні відомості

Імітаційне моделювання – це процес створення та тестування алгоритму, який імітує поведінку і взаємодію досліджуваної системи з урахуванням випадкових вхідних впливів і зовнішнього середовища. Імітаційна модель має одну з головних характеристик – вона може бути об’єктом експерименту, де експеримент проводиться з моделлю, представленою у вигляді комп’ютерної програми.

Імітаційна модель відображає стохастичний процес змін дискретних станів системи. Під час реалізації моделі на комп’ютері збираються статистичні дані за показниками моделі, які є предметом дослідження. Після завершення моделювання накопичена статистика обробляється, і результати моделювання отримують у вигляді вибіркових розподілів досліджуваних величин. Математичну основу імітаційного моделювання становлять математична статистика та теорія ймовірностей.

Рівномірно розподілена випадкова величина є необхідною для моделювання випадкових процесів та величин, випадкові величини з іншими законами розподілу часто формуються на її основі.

Неперервна випадкова величина γ має рівномірний розподіл у діапазоні $[a, b]$, якщо її щільність ймовірності $f(x)$ визначається як:

$$f(x) = \begin{cases} \dfrac{1}{b-a}, & x \in (a,b), \\ 0, & x \notin (a,b). \end{cases}$$

Графік щільності ймовірності неперервної випадкової величини наведено на рис. 3.1.

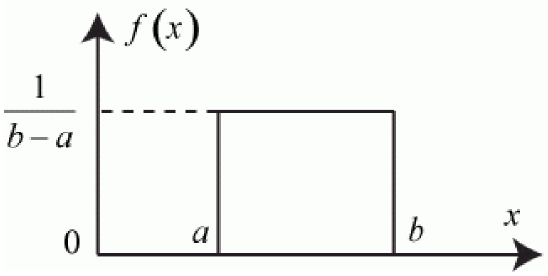

Рисунок 3.1 – Графік щільності ймовірності неперервної випадкової величини

Характеристики рівномірного закону розподілу:

$$M(X)=\frac{a+b}{2},\quad D(X)=\frac{(b-a)^2}{12};\ \ \sigma(X)=\frac{b-a}{2\sqrt{3}}.$$

Неперервні випадкові величини існують лише в теорії – усі випадкові величини дискретні і крок дискретності дорівнює найменшій одиниці виміру. Для формування послідовності випадкових чисел у комп’ютері може використовуватися один із трьох основних способів:

- апаратний (фізичний) – випадкові числа формуються спеціальним пристроєм. Джерелом випадкових чисел найчастіше є шуми в електронних приладах;
- табличний (файловий) – випадкові числа формуються заздалегідь чи беруться із відповідного довідника та поміщаються в оперативну або зовнішню пам’ять комп’ютера у вигляді таблиці (файлу);
- алгоритмічний (програмний) – випадкові числа формуються за допомогою спеціальних алгоритмів (формул) і програм, що їх реалізують, при кожному зверненні моделюючого алгоритму за випадковим числом.

На сьогодні майже скрізь використовуються алгоритмічні датчики випадкових чисел. Вони не забезпечують отримання теоретично «чистої» випадковості чисел, оскільки їх формування відбувається за допомогою формул. Внаслідок цього рано чи пізно послідовність випадкових чисел почне повторюватися. Тому алгоритмічні датчики називають датчиками псевдовипадкових чисел.

Основою моделювання випадкових величин з довільними законами розподілу ймовірностей, як правило, є метод оберненої функції.

Використовується теорема: якщо випадкова величина Y має щільність розподілу ймовірностей $f(y)$, то розподіл випадкової величини

$$F(y) = \int_{-\infty}^{y} f(y)dy$$

рівномірний в інтервалі $[0, 1]$. Звідки:

$$x_i = \int_{-\infty} y_i f(y)dy.$$

Залишається знайти невідоме y_i, яке знаходиться у верхній межі інтегрування.

У моделюванні важливу роль відіграє випадкова величина з нормальним розподілом. Метод оберненої функції тут не підходить, оскільки його інтеграл не має аналітичного розв’язку. Для генерації нормальних випадкових чисел застосовують метод оберненої функції з кусочно-лінійною апроксимацією та метод на основі центральної граничної теореми.

Випадкова величина

$$\eta = \frac{\sum_{i=1}^{N} x_i - NM[x]}{\sqrt{ND[x]}},$$

де $\sum_{i=1}^{N} x_i$ – сума N випадкових чисел одного розподілу з математичним очікуванням $M[x]$ і дисперсією $D[x]$ при $N \to \infty$ асимптотично прагне до нормального розподілу з $M[\eta] = 0$ і дисперсією $D[\eta] = 1$.

Добре наближення до нормального розподілу зазвичай досягається вже при числі $N = 6$, де кожне випадкове число генерується як:

$$y_i = \frac{\sum_{i=1}^{N} x_i - 3}{\sqrt{0.5}} = \sqrt{2}\left(\sum_{i=1}^{6} x_i - 3\right).$$

Мета моделювання одиничної події – імітувати зміни двох станів елемента. Перехід між станами – випадковий: у будь-який момент часу система знаходиться в одному стані з ймовірністю P, в іншому – з ймовірністю $1 - P$. Моделлю такої одиничної події A є потрапляння значення x_i випадкової величини γ, яка рівномірно розподілена в інтервалі $[0, 1]$, в інтервал $[0, P(A)]$.

При моделюванні повної групи несумісних подій елемент системи може знаходитись у багатьох (більше ніж два) несумісних станах з відомими ймовірностями знаходження системи в цих станах. Виконується імітація стану елемента за допомогою жеребкування: у повній групі несумісних подій моделлю настання події A_m, яка стається з ймовірністю P_m, є потрапляння значення x_i у відрізок, який дорівнює P_m, шкали $\sum_{m=1}^{n} P_m = 1$, де n – число несумісних подій, як наведено на рис. 3.2.

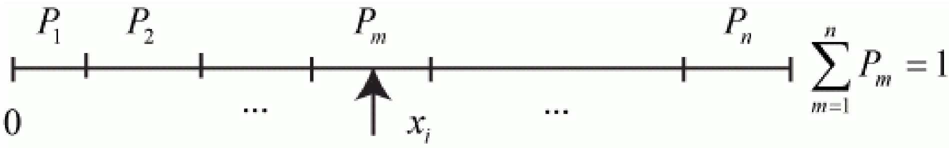

Рисунок 3.2 – Настання події A_m

Спосіб моделювання спільних незалежних подій полягає в тому, що ці події зводяться до однієї складної. Для кращого розуміння розглянемо моделювання двох подій A і B (збільшення кількості подій не вносить принципових змін у моделювання).

Нехай незалежні події A і B відбуваються з ймовірностями $P(A)$ та $P(B)$ відповідно.

Моделювання такої ситуації може бути виконано двома способами:

- визначення спільних результатів за допомогою жеребкування;
- послідовна перевірка результатів.

При жеребкуванні по ймовірностям $P(A)$ та $P(B)$ необхідно визначити ймовірності можливих сумісних незалежних подій.

Можливі результати спільної події визначаються комбінацією всіх можливих результатів кожної з подій. Якщо події A і B незалежні, то ймовірність спільного настання цих подій обчислюється як добуток ймовірностей кожної з подій:

1. Спільний результат (сумісний випадок) – відбуваються обидві події. Ймовірність цього результату: $P(A \cap B) = P(A) \cdot P(B)$. Межа ділянки $l_1 = P(A) \cdot P(B)$.

2. Відбувається лише подія A а B не відбувається. Ймовірність цього результату: $P(A \cap \bar{B}) = P(A) \cdot (1 - P(B))$. Межа ділянки $l_2 = l_1 + P(A) \cdot (1 - P(B))$.

3. Відбувається лише подія B, а A не відбувається. Ймовірність цього результату: $P(\bar{A} \cap B) = (1 - P(A)) \cdot P(B)$. Межа ділянки $l_3 = l_2 + (1 - P(A)) \cdot P(B)$.

4. Не відбувається жодна з подій. Ймовірність цього результату: $P(\bar{A} \cap \bar{B}) = (1 - P(A)) \cdot (1 - P(B))$. Межа ділянки $l_4 = l_3 + \left(1 - P(A)\right) \cdot \left(1 - P(B)\right) = 1$.

Відповідно, якщо випадкове число x_i потрапляє на ділянку $[0, l_1]$, то фіксується настання складної події AB (відбулись обидві події A і B) і т.д.

При послідовній перевірці результатів перевірку кожної з сумісних подій слід проводити за допомогою різних випадкових чисел, оскільки події незалежні.

При моделюванні спільних залежних подій потрібно враховувати умови їх взаємозв’язку:

1. Визначення ймовірностей умовних подій: Якщо події A і B є залежними, то ймовірність настання події B при умові, що подія A вже відбулася, позначається як умовна ймовірність $P(B/A)$. Так само можна

визначити ймовірність настання події A при умові, що подія B відбулася, тобто $P(A/B)$.

Формула для умовної ймовірності:

$$P(A \cap B) = P(A) \cdot P(B/A) = P(B) \cdot P(A/B).$$

2. Метод жеребкування. Для кожної події потрібно генерувати випадкові числа та враховувати, чи відбулася попередня подія, перед тим, як визначити ймовірність наступної. Наприклад: спочатку генерується випадкове число для події A. Якщо подія A сталася, то для події B генерується випадкове число з умовною ймовірністю $P(B/A)$.

3. Розрахунок ймовірностей для кожного з можливих варіантів.

4. Моделювання виконується через генерацію випадкових чисел: для першої події, далі для кожної наступної з урахуванням того, чи відбулася попередня подія, тобто з умовною ймовірністю.

3.2. Індивідуальні завдання

Написати програму для моделювання вказаної ситуації. Розрахувати ймовірності всіх можливих подій та спільних подій).

1. Імітувати 4 події: доступ до серверу, успішна передача даних, обробка запиту, збереження даних на сервері. Ймовірності для кожної події: 0.95, 0.9, 0.85, 0.8.

2. Моделювати 4 події для тестування ефективності API: отримання відповіді від сервера, обробка запиту, перевірка даних, збереження результатів. Ймовірності кожної події: 0.98, 0.97, 0.95, 0.92.

3. Імітувати 4 події в тестуванні системи: успішне з’єднання, отримання даних, обробка запиту, завершення операції. Ймовірності для подій: 0.99, 0.9, 0.85, 0.8.

4. Моделювати 4 події: вхід користувача на сайт, верифікація користувача, отримання даних з бази, генерація відповіді. Ймовірності цих подій: 0.95, 0.9, 0.85, 0.8.

5. Моделювати 4 події для перевірки транзакцій у платіжній системі: ініціація транзакції, верифікація картки, обробка транзакції, завершення операції. Ймовірності кожної події: 0.97, 0.95, 0.92, 0.9.

6. Імітувати 4 події в роботі серверу: підключення до сервера, отримання запиту, обробка запиту, відповідь на запит. Ймовірності для кожної події: 0.98, 0.97, 0.95, 0.9.

7. Створити модель для 4 подій при тестуванні сервісу: підключення користувача, аутентифікація, отримання доступу до даних, виведення результатів. Ймовірності: 0.98, 0.95, 0.9, 0.85.

8. Моделювати 4 події в роботі електронної пошти: отримання листа, перевірка на спам, обробка листа, збереження результату. Ймовірності: 0.99, 0.95, 0.9, 0.85.

3.3. Контрольні запитання

1. Що виконується під час імітаційного моделювання?

2. Як виконується моделювання рівномірно розподіленої випадкової величини?

3. Як виконується моделювання повної групи несумісних подій?

4. Як виконується моделювання спільних незалежних подій?

5. Як виконується моделювання спільних залежних подій?

4. ПЛАНУВАННЯ ЕКСПЕРИМЕНТІВ У МОДЕЛЮВАННІ

Мета заняття: набути практичні навички з використання стандартних планів, формального підходу до скорочення загальної кількості прогонів моделі, з визначення оцінки математичного очікування та дисперсії, розрахунку точності і кількості реалізацій моделі при визначенні ймовірностей результатів.

4.1. Короткі теоретичні відомості

Модель створюється для проведення експериментів. Експеримент складається зі спостережень, а кожне спостереження – з прогонів (реалізацій) моделі.

Комп'ютерний експеримент – це процес використання моделі для отримання та аналізу інформації про властивості досліджуваної системи.

Переваги комп'ютерного експерименту перед натурним:

- простота повторення умов експерименту;
- можливість управління експериментом, у тому числі його припинення та відновлення;
- легкість зміни умов експерименту (впливу зовнішнього середовища);
- виключення кореляції між даними, отриманими під час експерименту;
- визначення часового інтервалу дослідження моделі.

План експерименту визначає:

- обсяг обчислень;
- порядок проведення обчислень;
- способи накопичення і статистичної обробки результатів.

Мета планування експериментів:

- зменшення часу моделювання при збереженні точності і достовірності результатів;

- збільшення інформативності кожного спостереження;
- створення основи для дослідження.

Планування експериментів поділяється на стратегічне і тактичне:

- стратегічне планування – розробка умов і режимів експерименту для досягнення максимальної інформативності;
- тактичне планування – забезпечення точності та достовірності результатів.

Стратегічний план формується у факторному просторі – сукупності параметрів, які можна контролювати під час експерименту. Об'єкти стратегічного планування:

- вихідні змінні (реакції) $y = f(x) + \xi$, де ξ – помилка, перешкода, що викликається наявністю випадкових факторів; $f(x)$ – оператор, що моделює дію реальної системи та який визначає залежність вихідної змінної y від факторів x_i;
- вхідні змінні (фактори) $x_i, i = \overline{1, n}$;
- рівні факторів.

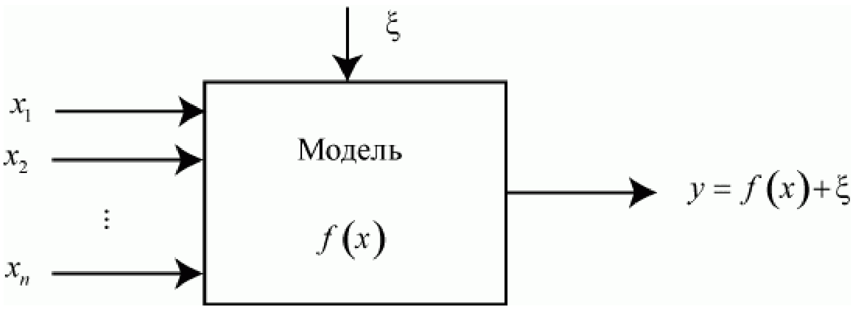

Рисунок 4.1 – Математична модель експерименту у вигляді «чорної скриньки»

Фактори можуть бути кількісними та (або) якісними. Кількісні фактори – це ті, значення яких виражені числами, наприклад: інтенсивність потоків, ємність буфера, кількість каналів у СМО, частка браку при виготовленні деталей тощо. Якісні фактори – це характеристики, які не можна виміряти числами, наприклад: дисципліни обслуговування (LIFO, FIFO), кваліфікація персоналу тощо.

Фактор має бути керованим, що означає можливість встановлення і підтримки його значення постійним або змінним згідно з планом експерименту. Можуть бути й некеровані фактори, наприклад, вплив зовнішнього середовища. До сукупності факторів висуваються дві основні вимоги:

- сумісність – усі можливі комбінації значень факторів мають бути здійсненними;
- незалежність – значення фактора можна встановлювати на будь-якому рівні незалежно від значень інших факторів.

Сукупність значень кожного фактора має назву – рівні фактора. Мінімальна кількість рівнів – два. З точки зору зручності планування експерименту доцільно встановлювати однакову кількість рівнів для всіх факторів. Таке планування називають симетричним. Аналіз даних експерименту значно спрощується, якщо рівні факторів будуть рівновіддаленими один від одного. Такий план називають ортогональним. Ортогональність досягається, коли два крайні рівні фактора вибираються як межі, а решта рівнів розташовуються так, щоб поділити відрізок навпіл.

Експеримент, в якому реалізуються всі поєднання рівнів усіх факторів, називається повним факторним експериментом (ПФЕ). Особливості побудови планів ПФЕ та проведення експериментів наведено в [8].

Інколи, якщо $f(x)$ – це імітаційна модель процесу, потрібно отримати так звану «вторинну модель» у вигляді аналітичної залежності. Надалі «вторинну модель» можна використовувати на практиці або в інших дослідженнях. У таких випадках математична модель формується за даними експерименту методом регресійного аналізу. Приклад регресійної залежності для трьох факторів:

$$y = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + b_{12}x_1x_2 + b_{13}x_1x_3 + b_{23}x_2x_3 + b_{123}x_1x_2x_3 + e.$$

При машинній реалізації ПФЕ у кожному спостереженні (інформаційній точці) необхідно виконати певну кількість прогонів (реалізацій) моделі, щоб забезпечити задану точність і достовірність значень відгуків. Визначення кількості прогонів моделі є частиною тактичного планування. Якщо p – кількість прогонів у кожному спостереженні, тоді для симетричного ПФЕ загальна кількість вимірювань дорівнює $N = pq^k$, де q – кількість рівнів факторів, k – кількість факторів.

Скоротити кількість прогонів можна за рахунок виключення якихось комбінацій рівнів факторів.

У багатьох випадках є певна свобода дій у виборі числа факторів k, кількості їх рівнів q і кількості прогонів p моделі в одному спостереженні. Оцінка впливу кожного з них на загальну кількість прогонів N за допомогою часткових похідних першого порядку від функції N за цими аргументами:

$$\frac{\partial N}{\partial k} = pq^k \ln q, \quad \frac{\partial N}{\partial q} = pkq^{k-1}, \quad \frac{\partial N}{\partial p} = q^k.$$

Попарне порівняння отриманих похідних:

$$\frac{\partial N}{\partial k} \bigg/ \frac{\partial N}{\partial q} = \frac{q \ln q}{k}, \quad \frac{\partial N}{\partial p} \bigg/ \frac{\partial N}{\partial q} = \frac{q}{kp}, \quad \frac{\partial N}{\partial p} \bigg/ \frac{\partial N}{\partial k} = \frac{1}{p \ln q}.$$

З чого можна зробити висновок:

- якщо $(kp > q)$ і $(k > q \ln q)$, то найбільший вплив на число N має зміна кількості рівнів q;
- якщо $(p \ln q > 1)$ і $(k < q \ln q)$, то найбільший вплив на число N має зміна кількості факторів k;
- якщо $(q > kp)$ і $(1 > p \ln q)$, то найбільший вплив на число N має зміна кількості реалізацій моделі на кожному рівні факторів.

Наприклад, на модель впливають чотири фактори по три рівня в кожному $(k = 4, q = 3)$. Для кожного спостереження виконується по вісім прогонів $(p = 8)$. ПФЕ потребує $N = pq^k = 8 \cdot 3^4 = 648$ спостережень (прогонів). Визначення, який з параметрів слід зменшити для досягнення суттєвого зменшення числа спостережень:

$$q \ln q = 3 \ln 3 = 3.3; \; kp = 4 \cdot 8 = 32; \; p \ln q = 8 \cdot 1.1 = 8.8.$$

Виконується умова $(kp > q)$ і $(k > q\ln q)$, тому що (32>3) і (4>3.3), тобто, найбільший вплив на зміну N дає зміна числа рівнів q. Якщо q зменшити на 1, то при ПФЕ буде потрібно виконати $N = pq^k = 8 \cdot 2^4 = 128$ прогонів.

В якості показників ефективності моделі використовується математичне очікування та дисперсія. Якщо при N прогонах було отримано незалежні значення $a_1, a_2, \dots, a_i, \dots, a_N$, то оцінка математичного очікування буде

$$\overline{a} = \frac{\sum_{i=1}^{N} a_i}{N}.$$

Зв'язок між точністю ε, достовірністю α та числом реалізацій моделі N:

$$\varepsilon = t_\alpha \frac{\sigma_\alpha}{\sqrt{N}},\ N = t_\alpha^2 \frac{\sigma_\alpha^2}{\varepsilon^2}.$$

Звідки:

- збільшення точності на порядок (зменшення помилки на порядок) вимагатиме збільшення числа реалізацій на два порядки;
- число необхідних реалізацій моделі не залежить від величини шуканого параметра, лише від дисперсії σ_α^2.

Для розрахунку використовується оцінка дисперсії, отримана на основі попереднього прогону моделі (зазвичай за $N^* = 1000$ реалізацій):

$$S_\alpha^2 = \frac{1}{N^* - 1}\sum_{i-1}^{N}(a_i - \overline{a})^2,\ S = \sqrt{\frac{1}{N^* - 1}\sum_{i-1}^{N}(a_i - \overline{a})^2},$$

де $\overline{a}$ – середньоарифметичне значення по N^* вимірюванням;

$$N = t_\alpha^2 \frac{S_a^2}{\varepsilon^2},\ \varepsilon = t_\alpha \frac{S_a}{\sqrt{N}}.$$

Якщо $N > N^*$, то моделювання слід продовжити до виконання N реалізацій, інакше моделювання закінчується – необхідну точність ε оцінки величини a при заданій достовірності α досягнуто.

При використанні оцінки дисперсії замість аргументу функції Лапласа t_α слід використовувати параметр розподілу Стьюдента $t^*_{\alpha(N-1)}$, значення якого залежать не тільки от рівня достовірності α, але й від числа ступенів волі $N-1$.

При визначенні оцінки дисперсії враховується такий зв'язок між точністю ε, достовірністю α та числом реалізацій моделі N:

$$N = t_\alpha^2 \frac{(\mu_4 - \sigma^4)}{\varepsilon^2};\ \varepsilon = t_\alpha \sqrt{\frac{(\mu_4 - \sigma^4)}{\varepsilon^2}},$$

де μ_4 – емпіричний центральний момент четвертого порядку:

$$\mu_4 = \frac{1}{N}\sum_{i=1}^{N}(a_i - \overline{a})^4.$$

Якщо випадкова величина має нормальний розподіл, то:

$$N = t_\alpha^2 \frac{2S^4}{\varepsilon^2};\ \varepsilon = t_\alpha \frac{S^2\sqrt{2}}{\sqrt{N}}.$$

Однакову точність визначення оцінок математичного очікування і дисперсії випадкового параметра при однаковій достовірності забезпечує різна кількість реалізацій моделі. Наприклад, якщо в результаті попередніх прогонів ($N^* = 1000$) було отримано оцінку дисперсії $S^2 = 10\,\text{ум.од}^2$, то кількість реалізацій моделі N_1 і N_2 для визначення оцінок математичного очікування і дисперсії випадкової величини a з точністю $\varepsilon = 0.1$ і достовірністю $\alpha = 0.9$ буде:

$$N_1 = t_\alpha^2 \frac{S^2}{\varepsilon^2} = 1.65^2 \frac{10}{0.1^2} \approx 2720$$
$$N_2 = t_\alpha^2 \frac{2S^4}{\varepsilon^2} = 1.65^2 \frac{200}{0.1^2} \approx 54450.$$

При розрахунку точності і кількості реалізацій моделі при визначенні ймовірностей результатів розглядається випадок, коли показником ефективності є ймовірність настання якоїсь події, наприклад, вихід з ладу техніки, завершення робіт у заданий час тощо. Оцінкою ймовірності P події $\bar{a}$ є частота її появи. Задача складається у визначенні такої кількості реалізацій N, щоб оцінка $\bar{P}$ відрізнялась від значення P менше, ніж на ε з заданою достовірністю.

Якщо апріорні відомості навіть про порядок ймовірності P невідомі, то використовується відносна погрішність d:

$$d = t_\alpha \sqrt{\frac{1-P}{PN}},\ N = t_\alpha^2 \frac{1-P}{Pd^2}.$$

При визначенні оцінок малих ймовірностей з високою точністю необхідно виконати дуже велику кількість реалізацій моделі. Наприклад для ймовірності $P = 0.1$ з відносною точністю $d = \frac{\varepsilon}{P} = 0.01$ при достовірності $\alpha = 0.9$ ($t_\alpha = 1.65$) число реалізацій моделі буде:

$$N = t_\alpha^2 \frac{1-P}{Pd^2} = 1.65^2 \frac{1-0.1}{0.1 \cdot 0.01^2} = 2.7252 \frac{0.9}{0.1 \cdot 0.01^2} = \frac{2.45}{10^{-5}} = 245000.$$

Обчислення потребують знання ймовірності P, тому що вона невідома, то виконується попередньо N^* прогонів моделі (зазвичай $N^* = 1000$) та обчислюється орієнтовне значення оцінки ймовірності $\bar{P}^*$, яку підставляють у формулу. Якщо $N > N^*$, то моделювання слід продовжити до виконання N реалізацій, інакше моделювання закінчується.

Розраховується кількість реалізацій для найгіршого випадку (P =0.5):

$$N_m = t_\alpha^2 \frac{P(1-P)}{\varepsilon^2} = t_\alpha^2 \frac{0.5(1-0.5)}{\varepsilon^2} = \frac{t_\alpha^2}{4\varepsilon^2}.$$

Якщо під час моделювання ймовірність буде значно відрізнятись від 0.5, то точність моделювання буде вище. Для визначення точності використовується формула:

$$\varepsilon = t_\alpha \sqrt{\frac{P(1-P)}{N_m}}.$$

При визначенні точності та кількості реалізацій моделі при залежному ряді даних виконується розбиття цього ряду на цикли, початок яких визначається однаковими умовами. Наприклад для одноканальної СМО канал може бути вільним – заявка одразу обробляється, або зайнятим – заявка очікує звільнення каналу; початок циклів – стан, коли канал вільний. Зрозуміло, що розміри циклів будуть різними.

Визначення необхідної кількості циклів моделювання:

$$n = t_\alpha^2 \frac{S^2}{\varepsilon^2 \overline{q}^2},$$

де $\bar{q} = \frac{\sum_{k=1}^{n^*} q_k}{n^*}$, q_k – кількість даних, що створюють k-й цикл; n^* – кількість реалізацій циклів по даним попередніх прогонів (зазвичай n^*=50-100); оцінка дисперсії $S^2 = S_\Theta^2 - 2\bar{t}_{\text{оч}} r_{\Theta,q} + t_{\text{оч}}^2 S_q^2$; $S_\Theta^2 = \frac{1}{n^*-1}\sum_{k=1}^{n^*}(\theta_k - \bar{\theta})^2$ – оцінка дисперсії θ; θ_k – сума часу очікування k-го циклу, $k = \overline{1,n}$; $t_{\text{оч}}$ – час очікування обробки попередніх даних; $S_q^2 = \frac{1}{n^*-1}\sum_{k=1}^{n^*}(q_k - \bar{q})^2$ – оцінка дисперсії $\bar{q}$; $r_{\theta,q} = \frac{1}{n^*-1}\sum_{k=1}^{n^*}(q_k - \bar{q})(\theta_k - \bar{\theta})$ – кореляційний момент випадкових величин θ і q; $\bar{\theta} = \frac{\sum_{k=1}^{n^*} \theta_k}{n^*}$; $\bar{t}_{\text{оч}} = \frac{\bar{\theta}}{\bar{q}}$.

Якщо $n > n^*$, то моделювання продовжується до досягнення n циклів, інакше моделювання закінчується.

Приклад побудови ПФЕ та розрахунку необхідної кількості прогонів моделі. Сервер виконує запити з інтервалами $T_1 = 2 \pm 1$ хв. Обчислювальна складність таких запитів з математичним очікуванням $M_1 = 6 \cdot 10^7 \pm 3 \cdot 10^7$оп і середньоквадратичним відхиленням $S_1 = 2 \cdot 10^5 \pm 1 \cdot 10^5$оп/с. Продуктивність сервера з обробки запитів $Q = 5 \cdot 10^5 \pm 2 \cdot 10^5$оп/с.

При виборі інтервалів варіювання рівнів факторів слід обрати крайні значення в діапазонах варіювання вхідних параметрів, тобто $T_{1\text{н}} = 1$ хв,

$T_{1в} = 3$ хв, $M_{1н}/S_{1н} = (3 \cdot 10^7 оп)/(1 \cdot 10^5 оп/с)$, $M_{1в}/S_{1в} = (9 \cdot 10^7 оп)/(3 \cdot 10^5 оп/с)$, $Q_н = 3 \cdot 10^5 оп/с$, $Q_в = 7 \cdot 10^5 оп/с$.

Складається план ПФЕ 2^3 (стовпці T_1, M_1/S_1, Q), наведений у табл. 4.1.

Таблиця 4.1 – План ПФЕ

№	T_1	M_1/S_1	Q	P_0	
				$N_0 = 1000$	$N = 9604$
1	-1	-1	-1	0.375	0.375
2	-1	-1	+1	0.584	0.583
3	-1	+1	-1	0.166	0.167
4	-1	+1	+1	0.32	0.319
5	+1	-1	-1	0.64	0.642
6	+1	-1	+1	0.809	0.808
7	+1	+1	-1	0.376	0.375

Проводиться експеримент. Виконуються попередні прогони моделі $N_0 = 1000$ для комбінації рівнів факторів $T_{1н}$, $M_{1н}/S_{1н}$, $Q_н$. Припустимо, що було отримано ймовірність обробки запитів $P_0 = 0.375$ (записано у відповідну комірку табл. 4.1). При заданій точності $\varepsilon = 0.01$ і достовірності $\alpha = 0.95$ ($t_\alpha = 1.96$) виконується розрахунок необхідної кількості прогонів моделі:

$$N = t_\alpha^2 \frac{P(1-P)}{\varepsilon^2} = 1.96^2 \frac{0.375(1-0.375)}{0.01^2} = 3.8416 \cdot \frac{0.375 \cdot 0.625}{0.01^2} = 9003.$$

Для «найгіршого» випадку (P =0.5):

$$N = 1.96^2 \frac{(0.5 \cdot 0.5)}{0.01^2} = 3.8416 \frac{0.25}{0.0001} = 9604.$$

Після цього виконується імітаційний експеримент для всіх комбінацій рівнів факторів у кількості 9604 прогони.

4.3. Індивідуальні завдання

Написати програму для проведення моделювання за допомогою ПФЕ. Визначити вихідні та вихідні дані, Сформувати фактори залежно від заданого закону розподілу випадкової величини. Побудувати план проведення експериментів. Задати точність і достовірність, розрахувати кількість прогонів. Під час моделювання врахувати закон, за яким розподіляються дані.

1. Оцінка продуктивності хмарних обчислень з урахуванням різних факторів: тип обробки (CPU, GPU), час затримки мережі (нормальний закон розподілу), розмір даних (експоненційний закон розподілу).

2. Оцінка потужності серверу. Фактори: час обробки запитів (нормальний закон розподілу), розмір вхідних даних (рівномірний закон розподілу), пропускна здатність каналу (експоненційний закон розподілу).

3. Оцінка навантаження на веб-сервер. Фактори: кількість одночасних запитів (геометричний закон розподілу), тип запитів (статичні, динамічні), розмір вхідних даних (нормальний закон розподілу).

4. Оцінка надійності системи з реплікацією. Фактори: кількість реплік (нормальний закон розподілу), тип мережевого з'єднання (TCP/IP, UDP), час відмови вузлів (рівномірний закон розподілу).

5. Оцінка ефективності зберігання даних. Фактори: тип зберігання (SSD, HDD), розмір файлів (експоненційний закон розподілу), частота доступу до даних (рівномірний закон розподілу).

6. Оцінка ефективності алгоритмів маршрутизації. Фактори: тип маршрутизації (статична, динамічна), кількість вузлів (рівномірний закон розподілу), час затримки мережі (біноміальний закон розподілу).

7. Оцінка ефективності багатозадачності на сервері. Фактори: число активних процесів (експоненційний закон розподілу), тип завдань (I/O, обчислювальні), час відгуку (рівномірний закон розподілу).

8. Оцінка надійності системи з багатьма користувачами. Фактори: кількість одночасних користувачів (нормальний закон розподілу), тип запитів (читання/запис), час відгуку (рівномірний закон розподілу).

4.3. Контрольні запитання

1. Які існують плани експериментів?
2. Які є показники ефективності моделі?
3. Як можна скоротити кількість прогонів моделі?
4. Для чого виконуються попередні прогони моделі?
5. Як визначається необхідна кількість прогонів моделі?

5. ОЦІНЮВАННЯ ХАРАКТЕРИСТИК ВИПАДКОВИХ ВЕЛИЧИН І ПРОЦЕСІВ

Мета заняття: ознайомитись з методами оцінювання характеристик випадкових величин, побудовою гістограм, визначити критерії Фішера та Вількоксона. Набути практичних навичок у виконанні однофакторного дисперсійного аналізу, виявлення несуттєвих факторів та виконати обробку результатів експерименту на основі регресії.

5.1. Короткі теоретичні відомості

Характеристики випадкової величини a:

- характеристика розміру (математичне очікування, медіана, мода, середнє геометричне тощо);
- характеристика розсіювання: (дисперсія, середньоквадратичне відхилення, розмах);
- характеристика зв’язку (кореляція);
- характеристика закону розподілу ймовірностей випадкової величини (функція щільності ймовірності, функція розподілу).

Обмеження на кількість реалізацій моделі дозволяє отримати приблизні значення цих характеристик (отримати оцінки характеристик). Оцінка характеристики є випадковою величиною. Оцінка повинна мати такі властивості:

- незміщеність – математичне очікування оцінки збігається з дійсним значенням характеристики;
- обґрунтованість – оцінка збігається за імовірністю до оцінювального параметра при необмеженому зростанні числа досліджень;
- ефективність – оцінки, які мають властивості незміщеності і обґрунтованості, повинні мати мінімальну дисперсію.

У табл. 5.1 наведені основні формули оцінок характеристик досліджуваної величини, що відповідають нормальному закону розподілу ймовірностей.

Таблиця 5.1 – Оцінки характеристик випадкової величини

Характеристика	Оцінка	Середньоквадратичне відхилення оцінки
Математичне очікування $M[x] = \int_{-\infty}^{\infty} xf(x)dx$	$\bar{x} = \frac{\sum_{i=1}^{N} x_i}{N}$	$\sigma_{\bar{x}} = \frac{S}{\sqrt{N}}$
Дисперсія $D[x] = M[x^2] - (M[x])^2$	$S^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2$	$\sigma_{S^2} = \sqrt{\frac{2}{N}}S^2$
Середньоквадратичне відхилення $\sigma_x = \sqrt{D[x]}$	$S = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2}$	$\sigma_S = \frac{S}{\sqrt{2N}}$
Ймовірність події P	$\bar{P} = \frac{m}{N}$	$\sigma_{\bar{P}} = \sqrt{\frac{P(1-P)}{N}}$
Коефіцієнт кореляції $Q_{x,y} = \frac{cov(x,y)}{\sigma_x \sigma_y}$	$\bar{r}_{xy} = \frac{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})\,(y_i - \bar{y})}{S_x S_y}$	$\sigma_{\bar{r}} = \frac{1 - \bar{r}_{xy}^2}{\sqrt{N}}$

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 $x_i(t_j)$, 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 очікуваннями та дисперсіями:

$$\bar{x}(t_j) = \frac{\sum_{i=1}^{N} x_i(t_j)}{N}, S_{x(t_j)}^2 = \frac{1}{N}\sum_{i=1}^{N}\left(x_i(t_j) - \overline{x(t_j)}\right)^2.$$

Ступінь залежності між перетинами визначає автокореляційна функція, її оцінка:

$$\bar{K}_{x(t_k,t_s)} = \frac{1}{N}\sum_{i=1}^{N}\left(x_i(t_k) - \bar{x}(t_k)\right) \cdot \left(x_i(t_s) - \bar{x}(t_s)\right),$$

де $x_i(t_k)$ і $x_i(t_s)$ – значення перетинів у точках t_k і t_s i-ї реалізації; $\bar{x}(t_k)$ і $\bar{x}(t_s)$ – оцінки математичних очікувань сукупності перетинів у точках t_k і t_s.

Для визначення закону розподілу ймовірностей і кількісних значень його характеристик використовується гістограма. На основі її вигляду робиться припущення щодо отриманого емпіричного розподілу ймовірностей. Далі перевіряється гіпотеза за допомогою критеріїв згоди (критерій Колмогорова, критерій Смірнова, критерій хі-квадрат – критерій Пірсона).

Оцінки математичного очікування і дисперсії по даним гістограми:

$$\bar{a} = \sum_{j=1}^{l} \bar{a}_j \bar{P}_j \, , S^2 = \sum_{j=1}^{l} \bar{a}_j^2 \bar{P}_j^2 - \frac{\Delta}{12},$$

де $\bar{a}_j$ – середнє значення кожного інтервалу; $\bar{P}_j$ – оцінка по кожному інтервалу; $\frac{\Delta}{12}$ – поправка Шеппарда.

У математичній статистиці використовують термін – гіпотеза. Гіпотеза – це припущення щодо:

- законів розподілу ймовірностей випадкових величин;
- значень характеристик випадкових величин;
- співпадіння законів розподілу двох або більше випадкових величин тощо.

Вихідна гіпотеза – нульова H_0. Протилежне твердження – конкуруюча (альтернативна) гіпотеза H_1. Перевірка гіпотези полягає в її прийнятті або відхиленні з допустимим мінімальним ризиком. При цьому можливі два види помилок:

- помилка першого роду – відхилення основної гіпотези, якщо вона вірна;
- помилка другого роду – прийняття основної гіпотези, якщо вона хибна.

Критерій перевірки (згоди) – правило, по якому приймається судження про істинність чи хибність основної гіпотези H_0.

Чим менше рівень значущості, тим менше ймовірність відхилити гіпотезу H_0, коли вона вірна, тобто здійснити помилку першого роду. Зі зменшенням рівня значущості розширюється область допустимих помилок, що призводить до збільшення ймовірності прийняття невірного рішення, тобто здійснення помилки другого роду.

Один з видів гіпотез – це гіпотеза про приналежність двох або більше вибірок до однієї генеральної сукупності. Ознаки, за якими виконується оцінка, часто недетерміновані, мають розсіювання. Перевірка гіпотези про тотожність вибіркових дисперсій до однієї генеральної дисперсії виконується за допомогою дисперсійного аналізу.

Для перевірки гіпотези про рівність дисперсій використовується незалежна функція Фішера, яка обчислюється за даними експерименту:

$$F = \frac{S_1^2}{S_2^2} = \frac{\frac{U}{k_1}}{\frac{V}{k_2}},$$

де U і V – випадкові величини, що мають розподіл χ^2; k_1 і k_2 – ступені волі випадкових величин U і V, $k_1 = N_1 - 1$, $k_2 = N_2 - 1$; N_1 і і N_2 – кількість іспитів (обсяги вибірок). При обчисленні F чисельник відношення $\frac{S_1^2}{S_2^2}$ повинен бути більше знаменника.

Графік щільності розподілу F наведено на рис. 5.1.

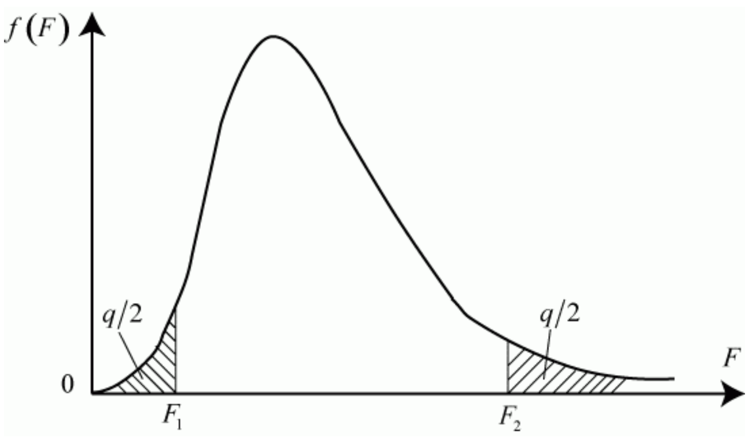

Рисунок 5.1 – Графік щільності розподілу F

Якщо обчислене по даним експерименту значення F потрапляє в область між точками F_1 і F_2, то гіпотеза H_0 не спростовується.

Критерій Вількоксона використовується також для перевірки гіпотези щодо тотожності функцій розподілів $F(x)$ та $F(y)$:

$$H_0: F(x) = F(y),$$

$$H_1: F(x) < F(y).$$

Для перевірки гіпотез необхідно розрахувати інверсії: виміряні значення x_i, $i = \overline{1, m}$ і y_j, $j = \overline{1, n}$ розташовуються в спільній послідовності за зростанням їх значень. Якщо гіпотеза H_0 вірна, то, числа x_i і y_j у спільній послідовності будуть добре перемішані – це визначається числом інверсій елементів першої послідовності відносно другої. Одна інверсія – якщо в спільній послідовності деякому x передує одне значення y. Якщо деякому x передує k значень y, то це значення x має k інверсій. Загальна кількість інверсій u має близький до нормального закон розподілу (при $m + n > 20$, $\min(m, n) \geq 3$) з математичним очікуванням та дисперсією:

$$M[u] = \frac{m - n}{2}, \;\; D[u] = \frac{m \cdot n}{12}(m + n + 1),$$

і при рівні значущості q межі критичної області, де приймається гіпотеза H_0:

$$u_1 = M[u] - t_\alpha \sigma_u, \;\; u_2 = M[u] + t_\alpha \sigma_u,$$

де t_α – аргумент функції Лапласа:

$$t_\alpha = \Phi^{-1}\left(\frac{1 - q}{2}\right).$$

Найчастіше використовують $q = 2$ ($t_\alpha = 2.33$), $q = 5$ ($t_\alpha = 1.96$), $q = 10$ ($t_\alpha = 1.65$).

Приклад перевірки гіпотези щодо тотожності функцій розподілів по критерію Вількоксона. Є два набори виміряних даних:

x:	0,8	1,9	3,0	3,5	3,8	2,5	1,7	0,9	1,0	2,3
y	1,4	2,1	3,1	3,6	2,7	1,8	1,1	0,2	1,6	2,8

Спільна послідовність:

$y_1(0.2)$, $x_1(0.8)$, $x_2(0.9)$, $x_3(1.0)$, $y_2(1.1)$, $y_3(1.4)$, $y_4(1.6)$, $x_4(1.7)$, $y_5(1.8)$, $x_5(1.9)$, $y_6(2.1)$, $x_6(2.3)$, $x_7(2.5)$, $y_7(2.7)$, $y_8(2.8)$, $x_8(3.0)$, $y_9(3.1)$, $x_9(3.5)$, $y_{10}(3.6)$, $x_{10}(3.8)$.

Кількість інверсій:

$$\bar{u} = 1 + 1 + 1 + 4 + 5 + 6 + 6 + 8 + 9 + 10 = 51.$$

$$M[u] = \frac{10 \cdot 10}{2} = 50,$$

$$D[u] = \frac{10 \cdot 10}{12}(10 + 10 + 1) = 175,$$

$$\sigma_u = \sqrt{175} = 13.23.$$

При рівні значущості $q = 5$ $t_\alpha = 1.96$.

Критичні точки:

$$u_1 = M[u] - t_\alpha \sigma_u = 50 - 1.96 \cdot 13.23 \approx 24,$$

$$u_2 = M[u] + t_\alpha \sigma_u = 50 + 1.96 \cdot 13.23 \approx 74.$$

Перевірка гіпотези H_0:

$$24 < 51 < 74,$$

тобто гіпотеза про ідентичність розподілів двох наборів виміряних даних не спростовується.

Дисперсійний аналіз застосовується для дослідження впливу однієї або кількох якісних змінних (факторів) на одну залежну кількісну змінну (відгук). Основною метою дисперсійного аналізу є дослідження значущості відмінностей між середніми за допомогою порівняння (аналізу) дисперсій.

При виконанні однофакторного дисперсійного аналізу накопичують результати n вимірювань контрольованого параметра (відгуку) a_{ij} при кожному з m варіантів фактора, $i = \overline{1, m}$ – номер варіанта фактора, $j = \overline{1, n}$ – номер вимірювання.

Визначається, чи є результати $m \cdot n$ вимірювань вибіркою однієї генеральної сукупності (варіанти фактора несуттєві).

Для кожного варіанта фактора проводиться n вимірювань контрольованого параметра (відгуку) a_{ij} та розраховується середнє значення:

$$\overline{a}_i = \frac{1}{n}\sum_{j=1}^{N} a_{ij}.$$

Спільне середнє арифметичне по всім $m \cdot n$ вимірюванням:

$$\overline{a} = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}}{m \cdot n}.$$

Сума квадратів відхилень по всім $m \cdot n$ вимірюванням:

$$Q = \sum_{i=1}^{m}\sum_{j=1}^{n} (a_{ij} - \overline{a})^2.$$

Або:

$$Q = n\sum_{i=1}^{m} (\overline{a}_i - \overline{a})^2 + \sum_{i=1}^{m}\sum_{j=1}^{n} (a_{ij} - \overline{a})^2.$$

Складові частини можна позначити як:

$$Q_1 = n\sum_{i=1}^{m} (\overline{a}_i - \overline{a})^2, \quad Q_2 = \sum_{i=1}^{m}\sum_{j=1}^{n} (a_{ij} - \overline{a})^2,$$

де Q_1 – сума квадратів відхилень між варіантами фактора; Q_1 – характеризує відхилення всередині кожного варіанта.

Для перевірки гіпотези про однорідність варіантів фактора використовується критерій:

$$F = \frac{\frac{1}{m-1}Q_1}{\frac{1}{m(n-1)}Q_2},$$

який має F-розподіл з $m-1$ та $m(n-1)$ ступенями волі. При порівнянні його значення з табличним F_2(з величиною рівня значущості $\frac{q}{2}$), якщо $F > F_2$, то є потрапляння в область малоймовірних значень F і гіпотеза не підтверджується – варіанти фактора не однотипні. Інакше – гіпотеза про однорідність варіантів фактора підтверджується.

Під час проведення експерименту велика кількість факторів ускладнює і знижує його ефективність. Слід виключити несуттєві фактори з експерименту. Для цього виконуються експерименти з N спостережень з урахуванням фактора, який аналізується на несуттєвість, та з N спостережень – без нього. Відгуки, які отримані, відповідно:

$$y_1', y_2', y_3', \ldots, y_i', \ldots, y_N';$$
$$y_1'', y_2'', y_3'', \ldots, y_i'' ;, \ldots, y_N''.$$

Якщо фактор несуттєвий, то ці послідовності мають однакові математичні очікування $M[y]$ та дисперсії σ_y^2. Z – випадкова величина, реалізації якої – послідовність випадкових чисел

$$z_i = y_i' - y_i'', i = \overline{1, N}.$$

Згідно центральній граничній теоремі:

$$M[\overline{z}] = M[z] = 0, \ \ \sigma_{\overline{z}} = \frac{\sigma_z}{\sqrt{N}}.$$

Обчислюється величина:

$$\overline{z} = t_\alpha \frac{\sigma_z}{\sqrt{N}}.$$

Якщо $\bar{z} > t_\alpha \frac{\sigma_y \sqrt{2}}{\sqrt{N}}$, прийнята гіпотеза про несуттєвість фактора не підтверджується.

Якщо $\bar{z} \leq t_\alpha \frac{\sigma_y \sqrt{2}}{\sqrt{N}}$, прийнята гіпотеза не спростовується.

Зазвичай величина σ_y невідома, тому слід використовувати її оцінку S_y:

$$S_y = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} (\overline{y} - y_i)^2}.$$

Оцінку $\bar{y}$ і ряд значень y_i можна отримати з даних першого або другого експериментів, тому що за гіпотезою вони однакові. Якщо $N < 100$, замість аргументу функції Лапласа t_α слід використовувати аргумент функції Стьюдента t_α'.

Часто під час досліджень визначається функціональна залежність між факторами і відгуком. Для цього виконується регресійний аналіз – сукупність методів побудови і дослідження регресійної залежності між величинами по статистичним даним, які накопичуються під час експерименту [8].

За причиною обмеженої кількості спостережень точні значення коефіцієнтів регресійної залежності отримати неможливо, тому будуть знайдені їх оцінки b_i, і рівняння регресії матиме вигляд:

$$\overline{y} = b_0 + b_1 x_1 + b_2 x_2 + \ldots + b_n x_n = \sum_{i=0}^{n} b_i x_i, \;\; x_0 = 1 .$$

Вибір рівняння регресії зазвичай починають з лінійної моделі. Наприклад, при двофакторному експерименті:

$$\overline{y} = b_0 + b_1 x_1 + b_2 x_2 .$$

Якщо така апроксимація дає несприятливі відхилення при порівнянні з експериментальними точками відгуку y, то модель ускладняється.

Коефіцієнти регресії b_i розраховуються при вирішенні системи рівнянь:

$$\sum_{l=1}^{N} y_l x_{jl} = \sum_{l=1}^{N} b_i \sum_{i=0}^{n} x_{il} x_{jl} ,$$

де x_{il} – значення i-го фактора в спостереженні номер l; y_l – значення відгуку в спостереженні номер l.

Довірчі границі для y приймають різні значення залежно від значень факторів. Часто обмежуються узагальненими оцінками адекватності збудованої моделі:

- величиною середнього абсолютного відхилення:

$$\varepsilon = \frac{\sum\limits_{l=1}^{N} (|y_l - \overline{y}_l|)}{N},$$

- величиною середньоквадратичної помилки на одиницю ваги:

$$\varepsilon_0 = \sqrt{\frac{\sum\limits_{l=1}^{N} (|y_l - \overline{y}_l|)}{N - (n+1)}}, \;\; N > (n+1) ,$$

де $\overline{y_l}$ – значення відгуку, що визначено по збудованому рівнянню регресії і даним x_{il}.

Вагою або ступенем волі експерименту називають різницю $N-(n+1)$ між числом спостережень N і числом коефіцієнтів регресії $(n+1)$.

5.2. Індивідуальні завдання

Для індивідуальної задачі з теми 4 виконати побудову регресійної моделі, перевірити наявність несуттєвих факторів, за критерієм Фішера та Вількоксона перевірити, чи належать до однієї генеральної сукупності вихідні дані (отримані в програмі для вирішення задачі з 4 теми) та розрахункові дані для побудованої регресійної моделі.

Всі розрахунки реалізувати в розробленій програмі.

5.3. Контрольні запитання

1. Які є основні характеристики випадкової величини?
2. Що таке статистична гіпотеза?
3. Для чого використовується дисперсійний аналіз?
4. Як можна визначити, що дві вибірки мають однаковий закон розподілу?
5. Як розраховуються коефіцієнти регресійної моделі?

6. ПОБУДОВА МОДЕЛЕЙ ІЗ ПРИСТРОЯМИ В СИСТЕМІ GPSS WORLD

Мета заняття: ознайомитись з системою моделювання GPSS World. Отримати практичні навички по моделюванню простих моделей у системі GPSS World.

6.1. Короткі теоретичні відомості

GPSS (General Purpose Simulation System) – система імітаційного моделювання загального призначення, мова моделювання системи GPSS використовується для імітаційного моделювання різних систем, зокрема систем масового обслуговування. Абстрактні елементи системи мають назву – об'єкти, кількість яких обмежена сьома категоріями: динамічна (транзакти), операційна (блоки), апаратна (одноканальні пристрої, пам'ять, логічні ключі), обчислювальна (змінні, функції, генератори випадкових чисел), статистична (черги, таблиці), запам'ятовуюча (комірки, матриці комірок), групуюча (числові групи, групи транзактів, списки) [2-4, 6, 9].

Динамічними об'єктами є транзакти, які створюються в певних точках моделі, просуваються планувальником через блоки, а потім знищуються (наприклад, замовлення в СМО). З кожним транзактом може бути пов'язана певна кількість параметрів, які містять необхідну інформацію про нього. Параметри нумерують або їм дають імена. Номери параметрів та імена використовують для посилань на значення, присвоєні параметрам. Транзакти можуть мати різні пріоритети. Пріоритет визначає перевагу, яку отримує транзакт, коли він та інші транзакти претендують на один і той самий ресурс.

Операційні об'єкти, тобто блоки, задають логіку функціонування моделі системи і визначають шлях руху транзактів між об'єктами апаратної категорії. Модель системи виглядає як сукупність блоків, об'єднаних відповідно до логіки роботи реальної системи в блок-схему, яку можна зобразити графічно.

Об’єкти апаратної категорії – це абстрактні елементи, на які може бути декомпозована реальна система. Транзакти взаємодіють з цими блоками: змінюють їхній стан і впливають на рух інших транзактів.

Обчислювальна категорія служить для опису таких ситуацій у процесі моделювання, коли зв’язки між компонентами модельованої системи через процеси найпростіше і компактно виражаються у вигляді математичних (аналітичних і логічних) співвідношень.

Об’єкти запам’ятовуючої категорії забезпечують доступ до збережених значень. Будь-яка активна транзакція може здійснити запис інформації в ці об’єкти та прочитати її. Матриці можуть мати до шести вимірів.

До статистичних об’єктів відносяться черги та таблиці. У будь-якій системі рух потоку транзакцій може бути затриманий через недоступність пристроїв. У цьому випадку затримані транзакти ставляться в чергу.

Об’єкти групуючої категорії типу дають змогу користувачеві звертатися до атрибутів транзактів даної групи для одночасної зміни деяких атрибутів усіх транзактів цієї множини в одному з режимів: у режимі транзакта або числовому. У режимі транзакта величини, які надходять до групи, зображають номерами за чергою транзактів даної групи. У другому режимі величини, які надходять, зображають групою числових значень.

Для здійснення обслуговування використовують об’єкти апаратної категорії. До них належать одноканальні і багатоканальні пристрої та логічні ключі (перемикачі). Потоки, які існують у реальних системах, у моделях імітують за допомогою транзактів.

GENERATE – це блок, через який транзакти заходять до моделі. На кількість різних блоків GENERATE в одній моделі обмежень не існує:

GENERATE [A],[B],[C],[D],[E]

Інтервал часу між послідовними появами транзактів блоку GENERATE називають інтервалом надходження. Інформація про ймовірнісний розподіл інтервалу надходження задається за допомогою операндів A і B. Усі можливі розподіли інтервалів часу надходження в GPSS класифікують на рівномірно розподілені й усі інші розподіли.

Операнд A – середній інтервал часу між послідовними надходженнями транзактів до моделі.

Операнд B задає модифікатор, який змінює значення інтервалу генерування транзактів порівняно з інтервалом, зазначеним операндом A. За допомогою модифікатора-інтервалу задають рівномірний закон розподілу часу між створенням транзактів.

Різниця A–B значень, заданих операндами A і B, дає нижню межу інтервалу, а сума A+B – його верхню межу.

Операнд C задає зміщення інтервалів (початкова затримка) – це момент часу появи першого транзакта в блоці GENERATE. Після цієї першої появи всі наступні появи транзактів виникають згідно з розподілом інтервалів часу, які задають операнди A і B.

Операнд D задає граничне значення загальної кількості транзактів, які можуть увійти до моделі через цей блок GENERATE протягом часу моделювання. Коли ця кількість досягнута, даний блок GENERATE перестає бути активним.

Операнд E встановлює клас пріоритету кожного з транзактів, які потрапляють до моделі. Для задання пріоритетів рекомендується використовувати послідовність цілих чисел 0, 1, 2, 3,... Чим більше число, тим вищий пріоритет.

У початковий момент часу в кожному блоці GENERATE здійснюється підготовка до виходу одного транзакта. У моделі перед блоком GENERATE мають стояти команди визначення EQU, INITIAL, FUNCTION, VARIABLE, FVARIABLE.

Транзакти видаляються з моделі через блок TERMINATE:

TERMINATE   [A]

Операнд А – число одиниць, на яке блок TERMINATE зменшує лічильник завершення, який визначає момент закінчення моделювання. Час моделювання задається в операторі START.

Для задання значень параметрів транзактів використовується блок ASSIGN:

ASSIGN A,B,[C]

Операнд А – номер параметра, якому присвоюється значення.

Операнд В – значення, яке слід додати, відняти або замінити значення в параметрі.

Операнд С задає номер модифікатора-функції, на яке значення операнду В помножується.

Елементами, які потребують обслуговування є транзакти. Якщо транзакт займає одноканальний пристрій, то для цього він входить у блок, який описує цей пристрій. Блок повинен бути блоком SEIZE. Звільнення пристрою є вхід транзакту в блок RELEASE:

SEIZE A

RELEASE A

Операнд А – це ім'я одноканального пристрою.

Якщо пристрій зайнято, транзакт розміщується в списку затримки даного пристрою згідно пріоритету.

Для імітації обслуговування використовується затримка в часі за допомогою блоку ADVANCE:

ADVANCE A,[B]

Операнд А – середній час обслуговування, В – спосіб модифікації операнду А: якщо В не функція, то він задає максимальне відхилення для діапазону випадкового часу затримки; якщо функція, то він помножується на значення операнду А.

Для визначення стану пристроїв використовується блок GATE, який працює в двох режимах (відмова у вході, коли не пропускає транзакти, якщо об'єкт не знаходиться в потрібному стані, та дозвіл у вході і альтернативному виході):

GATE X A,[B]

X – умовний оператор, операнд A – ім'я одноканального пристрою, B – номер наступного блока для транзакта. Умова X: «NU» – пристрій вільний, «U» – пристрій зайнятий.

Для отримання статистики з черг використовуються реєстратори черг QUEUE (стати в чергу) і DEPART (вийти з черги).

Блок QUEUE встановлює довжину черги:

QUEUE A,[B]

Операнд A задає ім'я черги, к довжині якої додаються одиниці. Операнд B визначає кількість таких одиниць.

Блок DEPART використовується для зменшення довжини черги:

DEPART A,[B]

Операнд A задає ім'я черги. Операнд B задає кількість одиниць, на яку зменшується довжина черги.

Для отримання щільності розподілу, її інтегральних відносних частот, середнього значення і стандартного відхилення деяких аргументів використовуються статистичні таблиці TABLE і QTABLE.

Команда опису таблиці TABLE:

Name TABLE A,B,C,D

Мітка Name – ім'я таблиці, операнд A – аргумент таблиці, операнд B – верхній ліміт першого частотного інтервалу, операнд C – ширина частотного інтервалу, операнд D – число частотних інтервалі.

Для збору даних транзакт повинен зайти в блок TABULATE з ім'ям таблиці:

TABULATE A,[B]

Операнд A – ім'я таблиці, операнд B – ваговий коефіцієнт, що задає число одиниць, які повинні бути занесені в той частотний інтервал, в який потрапило значення аргументу.

Q-таблиці є засобом отримання розподілу тільки часу перебування транзакту в черзі. Формат команди QTABLE такий самий як у TABLE, за винятком того, що операнд А задає ім’я черги. Для її створення виконується спочатку визначення:

Name QTABLE A,B,C,D

При використанні QTABLE інформація в таблицю заноситься автоматично при вході транзакта в блоки QUEUE і DEPART, блок TABULATE при цьому не потрібен.

Для змін маршрутів руху транзактів у моделі використовуються блоки DISPLACE, LOOP, GATE, TEST і TRANSFER.

Блок TRANSFER призначений для передачі транзакта в будь-який інший блок моделі:

TRANSFER [A],[B],[C],[D]

Операнд А задає режим вибору (один з дев’яти: «,» – безумовний, «-» – випадковий вибір одного з двох блоків, «BOTH» – послідовний вибір одного з двох блоків, «ALL» – послідовний вибір одного з кількох блоків, «PICK» – вибір випадковим чином одного з кількох блоків, «FN» – функціональний, «P» – параметричний, «SBR» – підпрограмний; «SIM» – одночасний).

Операнди В і С задають можливі значення номерів наступних блоків. Операнд D задає значення індексу тільки для режиму «ALL».

Блок DISPLACE призначений для пошуку будь-якого транзакта і переміщення його до нового блока:

DISPLACE A,B,[C],[D]

Операнд А – номер транзакта, який необхідно перемістити, операнд В – мітка блока, до якого переміщується транзакт, операнд С – номер параметра транзакта, в який записується час, що залишився до кінця його обслуговування, якщо він знаходився в списку майбутніх подій, операнд D – мітка альтернативного блока для транзакта.

Якщо необхідно перервати обслуговування транзакта на одноканальному пристрої, виконується захоплення пристрою черговим транзактом за допомогою блока PREEMPT:

PREEMPT A,[B],[C],[D],[E]

Операнд А – ім'я захоплюваного пристрою. Режим переривання задається операндом В («PR» – пріоритетний).

У пріоритетному режимі використовуються операнди C, D і E. Операнд C – ім'я блока, куди повинен бути переправлений перерваний транзакт. Якщо його значення дорівнює «RE», то операнд E визначає режим видалення перерваного транзакта. Операнд D задає номер параметра перерваного транзакта, в який записується час, що залишився до завершення обслуговування.

Транзакт, який захопив одноканальний пристрій, звільняє його за допомогою блока RETURN:

RETURN A

Операнд А – ім'я блока, який звільняється.

Для моделювання несправностей одноканальних пристроїв та інших ситуацій використовуються блоки, які реалізують їх недоступність та доступність**.**

Недоступність моделюється блоком FUNAVAIL:

FUNAVAIL A,[B],[C],[D],[E],[F],[G],[H]

Блок робить недоступним пристрій з іменем (операнд А). За транзакт, який займає пристрій відповідають операнди B, C, D; за перервані транзакти – операнди E, F; за транзакти, що знаходяться в списку відкладених і в списку затримки – операнди G, H.

Операнд В задає режими обробки транзакта («СО» – продовжити обробку транзакта під час недоступності; «RE» – видалити і направити транзакт до блока, мітка якого вказана операндом С; за замовчанням – після відновлення доступності транзакт може зайняти пристрій).

Операнд D задає ім'я параметра транзакта, що займає пристрій під час переводу його в недоступний стан.

Операнд E задає режими обробки транзактів, що знаходяться в списку переривань (транзактів, обслуговування яких було перервано): «CO» – продовжити роботу пристрою під час недоступності; «RE» – видалити і направити транзакти до блока, мітка якого вказана операндом F; за замовчанням – залишити перервані транзакти в списку переривань і заборонити їм займати пристрій під час недоступності.

Операнд G задає режими обробки транзактів зі списку відкладених переривань та списку затримки («CO» – продовжити роботу пристрою під час недоступності; «RE» – видалити і направити транзакти до блока, мітка якого вказана операндом H; за замовчанням – залишити транзакти в списках і заборонити їм займати пристрій під час недоступності).

Недоступність пристрою зберігається до тих пір, поки транзакт, який викликав перехід у недоступний стан, не увійде в блок FAVAIL, який змінює стан пристрою на доступний:

FAVAIL A

Всі транзакти, що очікують доступний стан пристрою, вказаний операндом A, активуються і можуть пробувати його зайняти.

Якщо заблоковані транзакти знаходяться в списку поточних подій, то при великій їх кількості витрачається багато часу на перегляд цього списку. Для економії машинного часу заблоковані транзакти слід розміщувати в списки користувача. Список користувача – деякий буфер, в який можуть тимчасово розміщуватись транзакти. Для виведення транзактів у список користувача використовується блок LINK.

У безумовному режимі, блок LINK має формат:

[Name] LINK A,B

Операнд A задає ім'я списку користувача, операнд B визначає, в яке місце списку слід розмістити транзакт («FIFO» – у кінець списку; «LIFO» – на початок списку; «PR» – за зменшенням пріоритету; «P» – після транзактів, значення відповідного параметра в яких менше; «M1» – у порядку зростання відносного часу перебування в моделі).

Для виведення одного або кількох транзактів зі списку користувача і повернення їх у список поточних подій використовується блок UNLINK:

[Name] UNLINK X A,B,C,[D],[E],[F]

Операнд A – ім'я списку користувача, операнд B – мітка блока, куди переходять виведені транзакти, операнд C – число транзактів (або «ALL» для виведення всіх).

Операнди D і E з умовним оператором X визначають спосіб і умови виведення транзактів зі списку користувача.

Операнд F вказує ім'я блока, куди переходить транзакт з блоку UNLINK, якщо зі списку користувача не виведено жодного транзакту.

Багатоканальний пристрій визначається командою STORAGE:

Name STORAGE A

Name – ім'я пристрою, операнд A задає ємність пристрою.

Зайняття і звільнення багатоканального пристрою імітується блоками ENTER і LEAVE:

ENTER A,[B]

LEAVE A,[B]

Операнд A використовується для вказівки імені пристрою. Операнд B задає число пристроїв (елементів пам'яті), яке повинно бути зайнятим у блоці ENTER або звільнено в блоці LEAVE.

Недоступність багатоканального пристрою моделюється блоком SUNAVAIL:

SUNAVAIL A

Операнд A задає ім'я пристрою.

Транзакти при спробі зайняти недоступний пристрій розміщуються в списку затримки багатоканального пристрою.

Недоступний стан продовжується до входу транзакту в блок SAVAIL:

SAVAIL A

Операнд A задає ім’я пристрою.

Стан багатоканального пристрою, як і для одноканального пристрою, перевіряється блоком GATE. Відмінність – в операторі X: «SE» – пристрій порожній; «SF» – пристрій заповнений; «SNE» – пристрій не порожній; «SNF» – пристрій не заповнений; «SNV» – пристрій не доступний; «SV» – пристрій доступний. Блок GATE також працює в режимах відмови у вході та дозволу у вході і альтернативному виході.

Моделювання перемикачів виконується за допомогою логічних ключів – блоків LOGIC:

LOGIC X A

Операнд A – ім’я логічного ключа. Логічний оператор X – стан логічного ключа: «S» – ключ вмикається; «R» – ключ вимикається; «I» – ключ інвертується.

Стан логічного ключа перевіряється блоком GATE:

GATE X A,[B]

Операнд A – ім’я логічного ключа. Операнд B – мітка блока, на який буде направлений транзакт при невиконанні умови X: «LS» дорівнює 1, якщо ключ увімкнуто та 0, якщо вимкнуто; «LR» дорівнює 1, якщо ключ вимкнуто та 0, якщо увімкнуто.

При моделюванні, зазвичай вирішуються дві основні задачі, які можна назвати прямою та зворотною:

- пряма задача полягає в обчисленні оцінки математичного очікування певного параметра модельованої системи за умови, що відомий час її роботи;
- зворотна задача включає в себе визначення оцінки математичного очікування часу роботи системи, при якому який-небудь її показник досягає заданого значення.

Приклад вирішення прямої та зворотної задачі за допомогою GPSS.

Вихідні дані: сервер обробляє запити від клієнтів з інтервалами, що розподілені за показним законом, з середнім значенням $T_1 = 120$ сек. Обчислювальна складність запитів розподілена за нормальним законом з математичним очікуванням $S_1 = 6 \cdot 10^7$ операцій і середньоквадратичним відхиленням $S_2 = 2 \cdot 10^5$. Продуктивність сервера $Q = 6 \cdot 10^5$ операцій за секунду. У випадку зайнятості сервера запит губиться. Таким чином, сервер є однофазною системою масового обслуговування розімкнутого типу з відмовами.

Пряма задача: побудувати модель для імітаційного моделювання, яка дозволяє визначити оцінку математичного очікування кількості запитів (далі – кількості запитів), оброблених сервером за час $T = 1$ година, а також оцінити математичне очікування ймовірності обробки запитів (далі – ймовірності обробки запитів).

Зворотна задача: побудувати модель для імітаційного моделювання, яка дозволяє визначити оцінку математичного очікування часу (далі – часу обробки), необхідного для обробки сервером N запитів, а також оцінити математичне очікування ймовірності обробки запитів.

Рішення прямої задачі.

Розрахунок кількості прогонів: припускається що, результати моделювання (ймовірність обробки запитів) необхідно отримати з довірчою ймовірністю $\alpha = 0.95$ і точністю $\varepsilon = 0.01$. Розрахунок ведеться для найгіршого випадку ($P = 0.5$):

$$N = t_\alpha^2 \frac{P(1-P)}{\varepsilon^2} 1.96^2 \frac{0.5(1-0.5)}{0.01^2} = 3.8416 \frac{0.25}{0.0001} = 9604.$$

Для імітації джерела запитів слід використовувати блок GENERATE, для імітації сервера як одноканального пристрою – блоки SEIZE і RELEASE, для імітації обробки запитів – блок ADVANCE.

Модель повинна складатися з таких елементів:

- задання вихідних даних;

- опис арифметичних виразів;
- сегмент імітації надходження і обробки запитів;
- сегмент вказівки часу моделювання і розрахунку результатів моделювання.

Ім’я серверу – Server. Виведення транзактів з моделі виконується за допомогою блоків TERMINATE (по одному блоку для оброблених та загублених транзактів), відповідні мітки: OkZap і LostZap. Для підрахунку кількості всіх запитів використовується мітка NZap.

Вихідні дані задаються змінними користувача за допомогою команди EQU. Час моделювання – TMod, арифметична змінна для розрахунку часу за нормальним законом розподілу обробки запиту на сервері – TObr.

Ймовірність обробки запитів на сервері Pobr визначається як відношення кількості оброблених запитів N$OkZap до загальної кількості запитів N$NZap (N – системний числовий атрибут, який визначає кількість транзактів, що потрапили в блок з мітками OkZap та NZap відповідно).

Текст програми:

```
T1_     EQU      120
S1_     EQU      6#10^7
S2_     EQU      2#10^5
Q_      EQU      6#10^5
Koef    EQU      1
TObr    VARIABLE (Normal(2,S1_#Koef,S2_#Koef)/Q_)
PObr    VARIABLE N$OkZap/N$NZap
TMod    EQU 3600    ; час моделювання - 1 година
        GENERATE (Exponential(1,0,T1_))     ; створити запит
           ; за експоненційним законом
NZap    GATE NU  Server,LostZap  ; визначення вільного стану сервера,
           ; якщо ні - перехід на LostZap
        SEIZE    Server     ; зайняти сервер
        ADVANCE  V$TObr     ; імітація обслуговування запиту
        RELEASE  Server     ; звільнити сервер
OkZap   TERMINATE     ; видалити оброблений запит
```

```
LostZap TERMINATE     ; видалити загублений запит

; сегмент генерації часу моделювання і розрахунку результатів
        GENERATE TMod
        TEST L   X$Prog,TG1,Met1
        SAVEVALUE     Prog,TG1
Met1    TEST E   TG1,1,Met2
        SAVEVALUE     PObr,(V$PObr)
        SAVEVALUE     Res,(INT(N$OkZap/X$Prog))
           ; X$Prog – кількість прогонів
Met2    TERMINATE     1
; сегмент запуску моделювання
        START    1000,NP    ; тестові прогони без виведення статистики
        RESET
        START    9604  ; час моделювання - кількість прогонів
        SAVEVALUE     Res,(INT(N$OkZap/X$Prog))
```

Результат моделювання:

```
SAVEVALUE              RETRY       VALUE
 POBR                    0           0.546
 PROG                    0        9604.000
 RES                     0          16.000
```

За одну годину сервером буде оброблено 16 запитів, ймовірність обробки дорівнює 0.546.

Рішення зворотної задачі.

Визначається кількість запитів, очікуваний час обробки яких необхідно визначити (N =16 – з результату рішення прямої задачі).

Текст програми:

```
T1_     EQU      120
S1_     EQU      6#10^7
S2_     EQU      2#10^5
Q_      EQU      6#10^5
N_      EQU      16
```

```
        GENERATE (Exponential(1,0,T1_))
KolZap  GATE NU  Server,LostZap
        SEIZE    Server     ; зайняти сервер
        ADVANCE  ((Normal(2,S1_,S2_))/Q_) ; імітація обробки запиту
        RELEASE  Server     ; звільнити сервер
        TRANSFER ,ObrZap ; передача транзакта на мітку ObrZap
LostZap TERMINATE
; сегмент завершення моделювання і розрахунку результатів
ObrZap  TEST L   X$Prog,TG1,Met1
        SAVEVALUE     Prog,TG1
        SAVEVALUE     NZap,0
Met1    SAVEVALUE     NZap+,1
        TEST E   X$NZap,N_,Ter1
        TEST E   TG1,1,Met2
        SAVEVALUE     PObr,(N$ObrZap/N$KolZap)
        SAVEVALUE     TimeNZap,((AC1-X$AC2)/X$Prog)
        SAVEVALUE     AC2,AC1
Met2    SAVEVALUE     NZap,0
        TERMINATE     1
Ter1    TERMINATE
        START    1000,NP    ; тестові прогони без виведення статистики
        RESET
        START    9604  ; час моделювання - кількість прогонів
```

У даному випадку один прогін визначається кількістю запитів. Лічильник оброблених запитів – NZap. Як тільки X$NZap = N_, лічильник завершень зменшується на одиницю. Розрахунок часу обробки: (AC1-X$AC2)/X$Prog, де AC1 – абсолютний модельний час, X$Prog – кількість прогонів, X$AC2 – абсолютний модельний час тестових прогонів.

Результат моделювання:

```
SAVEVALUE               RETRY         VALUE
 PROG                     0        9604.000
 NZAP                     0               0
 POBR                     0           0.546
 TIMENZAP                 0        3523.658
 AC2                      0    37369292.321
```

Число 3523.658 отримано тому, що під час моделювання не була врахована дробова частина кількості запитів, отримана в першій програмі.

Коротка інструкція по моделюванню в GPSS World:

1. Введення тексту виконується у вікні «Model».

2. Як розділювач між елементами використовується знак табуляції, між параметрами ставиться кома без інших розділювачів.

3. Після введення тексту програми моделі необхідно створити об’єкт «Процес моделювання» за допомогою команді «Create Simulation».

4. При наявності помилок у вікні «JOURNAL» виводиться список повідомлень, інакше виконується моделювання.

5. При виправленні помилок слід використовувати команду «Search / Next Error».

6. Для запуску моделі використовується команда «Command / Start», в якості параметра вказується кількість прогонів (якщо немає сегменту запуску моделі в самій програмі).

7. При відсутності помилок після моделювання створюється звіт у вікні «REPORT».

6.2. Індивідуальні завдання

Виконати моделювання в системі GPSS World. Вихідні дані самостійно сформувати згідно опису ситуації, обрати їх значення, закони розподілу.

1. Моделювання процесу обробки запитів на веб-сервері. Моделювати сервер, що обробляє вхідні HTTP-запити з різними інтервалами. Потрібно визначити середнє число запитів, оброблених сервером за годину.

2. Система обробки транзакцій у банкоматах. Побудувати модель для визначення часу обробки транзакцій у банкоматах з різними швидкостями відповіді, враховуючи можливість відмови.

3. Моделювання мережі доставки повідомлень. Створити модель для перевірки часу доставки повідомлень через мережу з випадковими затримками, обчислити ймовірність втрати повідомлення через затримки.

4. Моделювання роботи комп'ютерної мережі з пакетами даних. Моделювати мережу передачі даних і визначити середній час затримки доставки пакету та ймовірність втрати пакету.

5. Моделювання черги обробки клієнтів у системі підтримки. Моделювати чергу в системі підтримки клієнтів, де кожен запит має різний час обробки в залежності від типу проблеми.

6. Моделювання процесу оновлення програмного забезпечення через інтернет. Створити модель для оцінки часу завантаження і оновлення програм через інтернет-з'єднання з різними швидкостями.

7. Моделювання роботи системи резервного копіювання даних. Моделювати систему резервного копіювання даних, де час на виконання копії залежить від розміру файлів.

8. Моделювання роботи інтернет-магазину з обробкою замовлень. Побудувати модель для визначення кількості замовлень, оброблених в онлайн-магазині за певний період.

6.3. Контрольні запитання

1. Які складові системи GPSS World?
2. Що таке транзакти?
3. Як виглядає синтаксис мови GPSS?
4. Які є основні блоки в мові GPSS?
5. Як вирішуються пряма та зворотна задачі в GPSS World?

7. ДИСПЕРСІЙНИЙ АНАЛІЗ ТА РЕГРЕСІЙНИЙ АНАЛІЗ У СИСТЕМІ GPSS WORLD

Мета заняття: ознайомитись з можливостями проведення дисперсійного та регресійного аналізу в системі GPSS World, отримати практичні навички по виконанню відсіювального та оптимізаційного експерименту.

7.1. Короткі теоретичні відомості

Сутність дисперсійного аналізу полягає в проведенні багатофакторного дисперсійного аналізу з метою виявлення ступеня впливу різних факторів та їх комбінацій (взаємодій) на значення цільової функції (функції відгуку, представленої у вигляді рівняння регресії).

Неявно в дисперсійному аналізі використовується адитивна математична модель, що визначає компоненти змін у спостереженнях Таку модель називають статистичною моделлю.

Наприклад, при двофакторному дисперсійному аналізі модель має вигляд:

$$y_{ijk} = \mu + \alpha_i^A + \alpha_j^B + \alpha_{ij}^{AB} + \varepsilon_{ijk},$$

де i – номер рівня фактора A; j – номер рівня фактора B; k – номер спостереження; y_{ijk} – відгук; μ – спільне середнє по всім спостереженням; α_i^A – головний ефект фактора A на рівні i; α_j^B – головний ефект фактора B на рівні j; α_{ij}^{AB} – взаємодія фактора A на рівні i і фактора B на рівні j; ε_{ijk} – випадкова помилка спостереження.

Головний ефект фактора визначає частку впливу цього фактора на значення функції відгуку під час переходу фактора з нижчого рівня на вищий. Дисперсійний аналіз завершується побудовою таблиці, в якій аналізується вплив факторів A і B, взаємодія між факторами та випадкові помилки спостережень. За допомогою дисперсійного аналізу перевіряється гіпотеза про відсутність впливу фактора. Для перевірки гіпотези використовується F-розподіл Фішера. Критерій Фішера визначає відношення двох вибіркових дисперсій. Якщо фактор суттєво впливає на відгук, то F-розподіл набуває великих значень, і F-статистика стає значущою. Таким чином, відповідний фактор вважається значущим.

Вирішення прямої задачі дисперсійного аналізу на прикладі прямої задачі з теми 6: дослідження залежності ймовірності обробки запитів від трьох факторів: $T_1 = 120 \pm 60$ сек., $Q = 6 \cdot 10^5 \pm 1 \cdot 10^5$ операцій за секунду та $Koef = 1 \pm 0.5$ (коефіцієнт впливу на параметри нормального розподілу обчислювальної складності обробки запитів) при їх мінімальних і максимальних значеннях. З тексту програми моделі необхідно видалити сегмент запуску моделювання.

Для проведення аналізу необхідно додати відсіювальний експеримент через команду «Edit / Insert Experiment / Screening», як наведено на рис. 7.1.

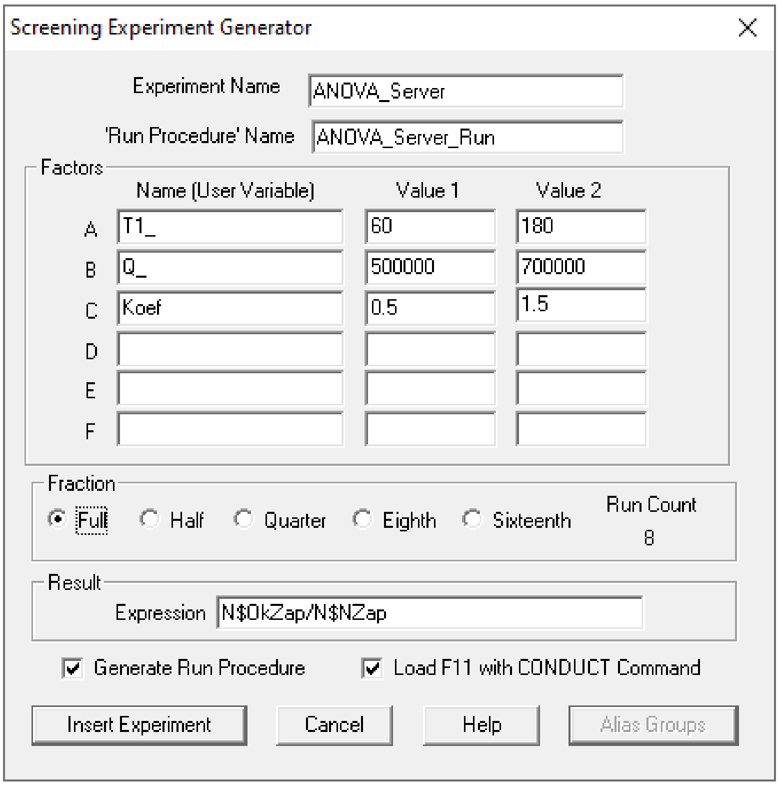

Рисунок 7.1 – Вікно додання відсіювального експерименту

У полі «Expression» необхідно ввести вираз, за яким обчислюється ймовірність обробки запитів. Після натискання на кнопку «Insert Experiment» у вікні «Run Procedure Generation» виконується корегування кількості прогонів, як наведено на рис. 7.2.

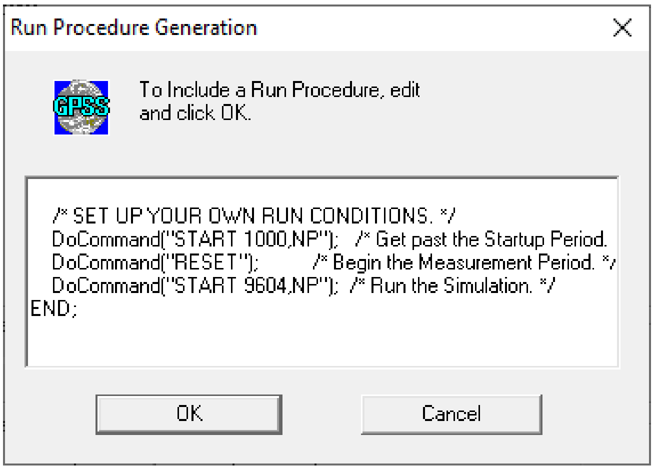

Рисунок 7.2 – Вікно «Run Procedure Generation»

Текст програми моделі:

```
T1_     EQU      120
S1_     EQU      6#10^7
S2_     EQU      2#10^5
Q_      EQU      6#10^5
Koef    EQU      1
TObr    VARIABLE (Normal(2,S1_#Koef,S2_#Koef)/Q_)
PObr    VARIABLE N$OkZap/N$NZap
TMod    EQU 3600    ; час моделювання - 1 година
        GENERATE (Exponential(1,0,T1_))      ; створити запит
            ; за експоненційним законом
NZap    GATE NU  Server,LostZap  ; визначення вільного стану сервера,
            ; якщо ні - перехід на LostZap
        SEIZE    Server      ; зайняти сервер
        ADVANCE  V$TObr      ; імітація обслуговування запиту
        RELEASE  Server      ; звільнити сервер
OkZap   TERMINATE      ; видалити оброблений запит
LostZap TERMINATE      ; видалити загублений запит
```

```
; сегмент генерації часу моделювання і розрахунку результатів
        GENERATE TMod
        TEST L   X$Prog,TG1,Met1
        SAVEVALUE     Prog,TG1
Met1    TEST E   TG1,1,Met2
        SAVEVALUE     PObr,(V$PObr)
        SAVEVALUE     Res,(INT(N$OkZap/X$Prog))
           ; X$Prog - кількість прогонів
Met2    TERMINATE     1

*******************************************************
*                                                     *
*                  ANOVA_Server                       *
*          Factorial Screening Experiment             *
*                                                     *
*******************************************************
ANOVA_Server_Results MATRIX ,2,2,2
INITIAL ANOVA_Server_Results,UNSPECIFIED
ANOVA_Server_NextRunNumber EQU  0

EXPERIMENT ANOVA_Server() BEGIN

        /*  Run 1  */
        T1_ = 60;
        Q_ = 500000;
        Koef = 0.5;
        IF
(StringCompare(DataType(ANOVA_Server_Results[1,1,1]),"UNSPECIFIED")'E'
0)
        THEN BEGIN
            /*  Set the Run Number Variable at the beginning. */

            ANOVA_Server_NextRunNumber = 1;
            /*  Log the Run and Execute the Simulation */
            ANOVA_Server_GetResult();
            ANOVA_Server_Results[1,1,1] = N$OkZap/N$NZap;
        END;
```

```
        /*  Run 2  */
        T1_ = 60;
        Q_ = 500000;
        Koef = 1.5;
        IF
(StringCompare(DataType(ANOVA_Server_Results[1,1,2]),"UNSPECIFIED")'E'
0)
        THEN BEGIN
            /*  Log the Run and Execute the Simulation */
            ANOVA_Server_GetResult();
            ANOVA_Server_Results[1,1,2] = N$OkZap/N$NZap;
        END;

        /*  Run 3  */
        T1_ = 60;
        Q_ = 700000;
        Koef = 0.5;
        IF
(StringCompare(DataType(ANOVA_Server_Results[1,2,1]),"UNSPECIFIED")'E'
0)
        THEN BEGIN
            /*  Log the Run and Execute the Simulation */
            ANOVA_Server_GetResult();
            ANOVA_Server_Results[1,2,1] = N$OkZap/N$NZap;
        END;

        /*  Run 4  */
        T1_ = 60;
        Q_ = 700000;
        Koef = 1.5;
        IF
(StringCompare(DataType(ANOVA_Server_Results[1,2,2]),"UNSPECIFIED")'E'
0)
        THEN BEGIN
            /*  Log the Run and Execute the Simulation */
            ANOVA_Server_GetResult();
            ANOVA_Server_Results[1,2,2] = N$OkZap/N$NZap;
        END;
```

```
        /*  Run 5  */
        T1_ = 180;
        Q_ = 500000;
        Koef = 0.5;
        IF
(StringCompare(DataType(ANOVA_Server_Results[2,1,1]),"UNSPECIFIED")'E'
0)
        THEN BEGIN
            /*  Log the Run and Execute the Simulation */
            ANOVA_Server_GetResult();
            ANOVA_Server_Results[2,1,1] = N$OkZap/N$NZap;
        END;

        /*  Run 6  */
        T1_ = 180;
        Q_ = 500000;
        Koef = 1.5;
        IF
(StringCompare(DataType(ANOVA_Server_Results[2,1,2]),"UNSPECIFIED")'E'
0)
        THEN BEGIN
            /*  Log the Run and Execute the Simulation */
            ANOVA_Server_GetResult();
            ANOVA_Server_Results[2,1,2] = N$OkZap/N$NZap;
        END;

        /*  Run 7  */
        T1_ = 180;
        Q_ = 700000;
        Koef = 0.5;
        IF
(StringCompare(DataType(ANOVA_Server_Results[2,2,1]),"UNSPECIFIED")'E'
0)
        THEN BEGIN
            /*  Log the Run and Execute the Simulation */
            ANOVA_Server_GetResult();
            ANOVA_Server_Results[2,2,1] = N$OkZap/N$NZap;
```

```
        END;

        /*  Run 8  */
        T1_ = 180;
        Q_ = 700000;
        Koef = 1.5;
        IF
(StringCompare(DataType(ANOVA_Server_Results[2,2,2]),"UNSPECIFIED")'E'
0)
        THEN BEGIN
            /*  Log the Run and Execute the Simulation */
            ANOVA_Server_GetResult();
            ANOVA_Server_Results[2,2,2] = N$OkZap/N$NZap;
        END;

        /*  Aliased Effects in Fractional Factorial Experiment */
        SE_Effects(ANOVA_Server_Results,"I");

END;

*******************************************************
*          The Run Execution Procedure                *
*******************************************************

PROCEDURE ANOVA_Server_GetResult() BEGIN

    /*    Run Simulation and Log Results.  */
    /*    Treatments have already been set for this run.  */

    TEMPORARY CurrentYield,ShowString,CommandString;

    /*  Run Procedure Call */

    ANOVA_Server_Run(ANOVA_Server_NextRunNumber);
    CurrentYield = N$OkZap/N$NZap;
    ShowString                  =                  PolyCatenate("Run
",String(ANOVA_Server_NextRunNumber),". ", "" );
```

```
    ShowString             =               PolyCatenate(ShowString,"
Yield=",String(CurrentYield),". ");
    ShowString = PolyCatenate(ShowString," T1_=",String(T1_), ";" );
    ShowString = PolyCatenate(ShowString," Q_=",String(Q_), ";" );
    ShowString = PolyCatenate(ShowString," Koef=",String(Koef), ";" );
    CommandString = PolyCatenate("SHOW """,ShowString,"""", "" );
    DoCommand(CommandString);
    ANOVA_Server_NextRunNumber = ANOVA_Server_NextRunNumber + 1;
    RETURN CurrentYield;

END;

******************************************************
*                Run Procedure                       *
******************************************************
PROCEDURE ANOVA_Server_Run(Run_Number) BEGIN
    DoCommand("CLEAR OFF");    /* Must use OFF to preserve results. */

    /* EXPAND THIS RMULT IF YOU HAVE MORE RNGs. */
    /* All Random Number Streams must have new seeds. */
    TEMPORARY CommandString;
    /* Evaluate before passing to DoCommand. */
    CommandString = Catenate("RMULT ",Run_Number#111);
    /* DoCommand compiles the string in Global Context. */
    DoCommand(CommandString);

    /* SET UP YOUR OWN RUN CONDITIONS. */
    DoCommand("START 1000,NP");   /* Get past the Startup Period. */
    DoCommand("RESET");          /* Begin the Measurement Period. */
    DoCommand("START 9604,NP");  /* Run the Simulation. */
END;
******************************************************
```

Для запуску моделювання необхідно обрати команду «Create Simulation», і у випадку відсутності помилок після формування звіту натиснути клавішу F11 для проведення експериментів. Після їх проведення можна отримати результати у вікні «JOURNAL» (рис. 7.3).

Alias Group	Effect	Sum of Squares	Degrees of Freedom	F - for Only Main Effects	Critical Value of F (p=.05)
A	0.247	0.122	1	1023.114	7.71
B	0.073	0.011	1	89.953	7.71
AB	-0.003				
C	-0.246	0.121	1	1019.104	7.71
AC	0.011				
BC	0.002				
ABC	0.010				
Error		0.000	4		
Total		0.255	7		
Grand Mean	0.536				

Рисунок 7.3 – Результати проведення експериментів

У таблиці наведені фактори і їх взаємодія. Для всіх ефектів вказані коефіцієнти «Effect», з якими вони входять до статистичної моделі, для головних ефектів вказані суми квадратів відхилень «Sum of Squares». У стовпці «Degrees of Freedom» наведені ступені волі відповідних вимірювань. У стовпці «F-for Only Main Effects» – обчислені значення F-статистик для головних ефектів. Чим більше значення F-статистики, тим сильніше ефект, фактор вважається значущим, якщо це значення перевищує критичне значення «Critical Value of F (p=0,5)» F-розподілу для рівня значущості 0.05. За даними експерименту в прикладі найбільший ефект на ймовірність обробки запитів дають фактори A і C.

Рядок «Error» відображає залишкову складову дисперсії і відповідний ступінь волі. Рядок «Total» – загальна сума квадратів помилок. Рядок «Grand Mean» – середнє значення результату дослідження по даним всього експерименту.

Проведення оптимізаційного експерименту складається з визначення таких значень рівнів факторів, при яких показник ефективності процесу сягає максимального/мінімального значення залежно від сутності показника ефективності, який описується як рівняння регресії:

$$y = f(x_1, x_2, \dots, x_n),$$

де x_i – значення рівнів факторів процесу; n – кількість факторів.

Метод поверхень є найбільш ефективним методом комп’ютерного моделювання. Рівняння регресії вважається як рівняння поверхні в багатофакторному просторі і оптимальне рішення – це координати (значення факторів) вершини/впадини на поверхні. Пошук оптимуму – послідовні вимірювання значень рівнів факторів у напрямку покращення показника ефективності.

Під час моделювання в GPSS виконується спроба підібрати лінійну модель або модель другого порядку.

Вирішення прямої задачі регресійного аналізу на прикладі задачі з теми 6 за умови, що обчислювальна складність обробки запитів сервером розподілена по експоненційному закону – блок імітації обслуговування запиту:

```
ADVANCE    ((Exponential(32,0,S2_))/Q_)
```

Для проведення аналізу необхідно додати оптимізаційний експеримент через команду «Edit / Insert Experiment / Optimising», як наведено на рис. 7.4.

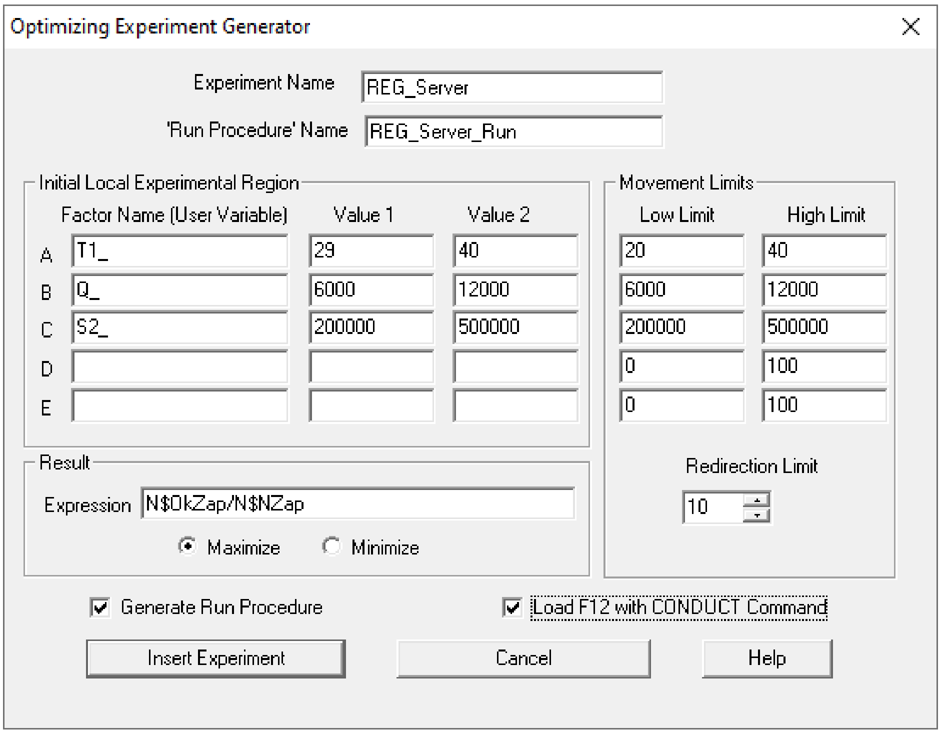

Рисунок 7.4 – Вікно додання оптимізаційного експерименту

Після натискання на кнопку «Insert Experiment» у вікні «Run Procedure Generation» виконується корегування кількості прогонів так само, як наведено на рис. 7.2.

Текст програми моделі не наводиться: модель залишилась не змінною, текст для проведення оптимізаційного експерименту, який додається програмою GPSS є стандартним.

Для запуску моделювання необхідно обрати команду «Create Simulation», і у випадку відсутності помилок після формування звіту натиснути клавішу F12 для проведення експериментів. Після їх проведення можна отримати результати у вікні «JOURNAL» (рис. 7.5).

Під час проведення оптимізаційного експерименту важливим є визначення локальної експериментальної області. Значення факторів, зазначені дослідником, формують початкову точку експерименту. Це впливає на досягнення мети експерименту, яка полягає в наступному:

- отриманні оптимального значення в межах локальної області;
- виконанні підтверджувального прогону;
- виведенні моделі, яка була використана.

Варто враховувати, що досягнення мети у вигляді математичної моделі в оптимізаційному експерименті не завжди є можливим.

```
Simulation in Progress.
Ą Simulation in an Experiment has ended.  Clock is 38174400.000000.
"Run 209.   Yield=.9460666964217284.  T1_=39; S_=20001; Q_=8999;"
       Second order model fails. Saddle point detected.
        Using Model:
           Y = 0.91632 +0.00106414  A -2.15504e-006  B +3.48748e-006  C
        Optimum could not be predicted.
        Saddle point detected. Critical point is not optimal.
 RSM_FitSurfaceToData() returns 2.
"ERROR: Response Surface Fails. Optimum not found."
Experiment ended.
```

Рисунок 7.5 – Результати оптимізаційного експерименту

Як видно, було досягнуто сідлову точку при використанні регресійного рівняння, яке не є оптимальним:

$$Y = 0.91632 + 0.00106414 \cdot A - 2.15504e - 006 \cdot B + 3.48748e - 006 \cdot C.$$

7.2. Індивідуальні завдання

Для варіантів з теми 6 виконати відсіювальний та оптимізаційний експеримент за допомогою системи GPSS World.

7.3. Контрольні запитання

1. У чому полягає сутність дисперсійного аналізу?
2. Як виконується відсіювальний експеримент у системі GPSS World?
3. У чому полягає сутність оптимізаційного експерименту?
4. Як виконується оптимізаційний експеримент у системі GPSS World?
5. Чому не завжди можна отримати математичну модель при проведенні оптимізаційного експерименту?

8. РОБОТА З ЗОВНІШНІМИ ПОТОКАМИ ДАНИХ ТА PLUS-ПРОЦЕДУРАМИ В GPSS WORLD

Мета заняття: ознайомитись з можливостями використання зовнішніх потоків даних та PLUS-процедур у GPSS World.

8.1. Короткі теоретичні відомості

Для роботи з зовнішніми файлами в системі GPSS World використовуються спеціальні блоки. Організація підтримки файлових операцій виконується за допомогою блоків: OPEN, READ, SEEK, WRITE і CLOSE. Блоки READ, WRITE і SEEK виконують операції лише з одним рядком тексту.

Блок OPEN призначений для ініціалізації потоку даних:

OPEN   A,[B],[C]

Операнд А – дескриптор потоку даних, який визначає тип потоку. Якщо це порожній рядок, то потік створюється в пам’яті. Якщо це ім’я каналу (наприклад, «\pipe\mypipe»), то створюється канал. Інакше створюється потік введення/виведення для файлу з ім’ям А. Якщо файл з ім’ям А не знайдено, то він створюється. Зміни у файлі виконуються тільки після блоку CLOSE. Операнд В задає номер потоку даних, за замовчанням дорівнює 1. Операнд С – це мітка блока, до якого спрямовується транзакт у випадку помилки. Можливі коди помилок:

- 0 – немає помилки;
- 10 – дуже довге ім’я файла;
- 11 – помилка читання файла;
- 12 – під час спроби відкрити файл немає доступу до пам’яті.

Блок CLOSE призначений для закриття потоку даних і запису даних з віртуальної пам’яті в зовнішню:

CLOSE   A,[B],[C]

Операнд A задає ім'я параметру транзакта, в який записується код помилки закриття потоку даних. Операнд B задає номер потоку даних для закриття, за замовчанням – 1. Операнд C – це мітка блоку, на який спрямовується транзакт при помилці закриття потоку даних. Можливі коди помилок:

- 0 – немає помилки;
- 41 – запис файла в зовнішню пам'ять не виконано по причині помилки введення/виведення;
- 42 – файл не був відкритий.

Блок READ призначений для читання з потоку даних текстового рядка:

READ A,[B],[C]

Операнд A – ім'я параметра транзакта, в який записується прочитаний з потоку рядок. Після зчитування позиція поточного рядка збільшується на одиницю. Операнд B задає номер потоку даних, з якого виконується зчитування, за замовчанням – 1. Операнд C визначає мітку блока, ня який переходить транзакт у випадку помилки зчитування. Для отримання коду помилки використовується блок CLOSE. Можливі коди помилок:

- 21 – під час спроби читання немає доступу до пам'яті;
- 22 – файл не був відкритий.

Блок WRITE призначений для передачі текстового рядка потоку даних:

WRITE A,[B],[C],[D]

Операнд A – текстовий рядок, який необхідно передати потоку даних. Операнд B задає номер потоку даних, за замовчанням – 1. Операнд C визначає мітку блока, ня який переходить транзакт у випадку помилки запису. Можливі коди помилок:

- 0 – немає помилки;
- 31 – під час спроби запису немає доступу до пам'яті;
- 32 – файл не був відкритий.

Операнд D задає режим роботи. Якщо операнд D не задано або він дорівнює «ON», то блок WRITE працює в режимі вставки. Якщо «OFF» – у режимі заміни.

У режимі вставки всі текстові рядки, що знаходяться на або за позицією поточного рядка, зсуваються на одну позицію. Якщо позиція поточного рядка за останнім текстовим рядком, то вона встановлюється після останнього рядка. Далі копія нового рядка розміщується на позиції поточного рядка, після чого позиція поточного рядка збільшується на одиницю.

У режимі заміни, якщо позиція поточного рядка за останнім рядком, то всі позиції між останнім рядком та поточним рядком заповнюються нульовими текстовими рядками. Рядок, що знаходиться на позиції поточного рядка, видаляється. Копія нового рядка розміщується на позиції поточного рядка і позиція поточного рядка збільшується на одиницю

Блок SEEK встановлює позицію поточного рядка потоку даних:

SEEK A,[B]

Операнд A визначає нову позицію поточного рядка. Операнд B визначає номер потоку даних, за замовчанням – 1. Можливий код помилки:

- 31 – файл не був відкритий.

Приклад зчитування даних і запису значення в матриці розмірністю десять рядків на один стовбець. Текст програми моделі:

```
Total   MATRIX   ,10,1
GENERATE   ,,,1
OPEN    ("MODEL.TXT"),1,Flag ; Відкриття потоку
Again   READ     Num,1,Fin ; Зчитування одного рядка
SAVEVALUE Nrow+,1
ASSIGN Numrow,X$Nrow
MSAVEVALUE Total,P$Numrow,1,P$Num
TRANSFER ,Again
Fin     CLOSE Errn,1,Flagl
TERMINATE 1 ; Нормальне завершення
Flag TERMINATE 1 ; Завершення при помилці відкриття потоку
Flagl TERMINATE 1 ; Завершення при помилці закриття потоку
```

Зміст файла даних «MODEL.TXT»: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10.

Перевірка слушності роботи моделі виконується за наступними діями: команда «Create Simulation», команда «START», відкриття вікна «Matrix Window». Результат наведено на рис. 8.1.

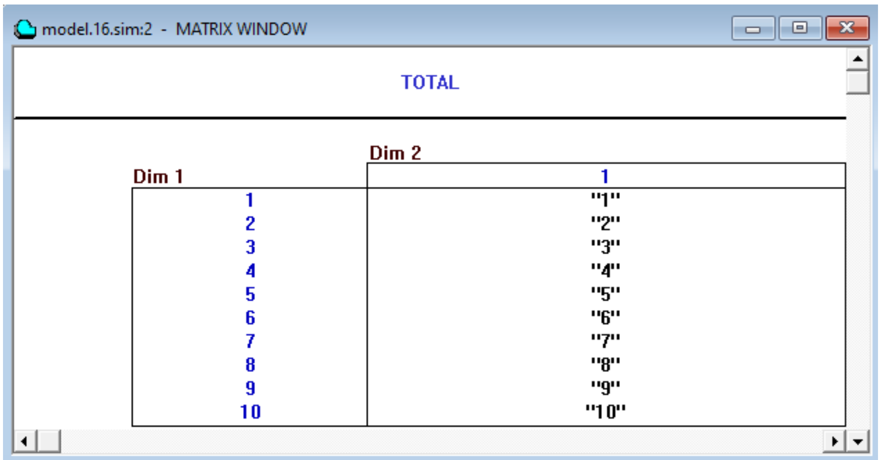

Рисунок 8.1 – Результат виконання моделі з прикладу

У GPSS World є можливість створення PLUS-процедур для використання в арифметичних виразах у блоках.

Ключове слово для опису PLUS-процедури – PROCEDURE, за яким ставиться ім’я процедури, далі – в круглих дужках задаються параметри. Після заголовку процедури розміщується тіло, обмежене ключовими словами BEGIN і END, після END ставиться крапка з комою. Кожен оператор у тілі процедури також закінчується крапкою з комою.

У тілі процедури можна використовувати змінні користувача, параметри процедури, системні числові атрибути та локальні змінні процедури TEMPORARY.

Форма оператора присвоювання:

змінна=вираз;

Оператор розгалуження:

IF (логічний вираз) THEN оператор [ELSE оператор];

Логічний вираз будується за правилами побудови логічних виразів у GPSS. Частина ELSE – не обов’язкова.

Оператор циклу:

WHILE (логічний вираз) DO оператор;

Оператор переходу на мітку всередині процедури:

GOTO мітка;

Мітка відокремлюється від оператора, на який потрібно перейти, символом двокрапки.

Оператор повернення значення:

RETURN вираз;

Обчислене значення виразу вертається як результат роботи процедури і відбувається вихід з неї.

Виклик PLUS-процедури виконується для обчислення виразу в блоці. Результат роботи процедури можна присвоїти якомусь параметру транзакта.

Приклад використання PLUS-процедури: пошук розв'язку задачі $y' = 0.6 - 0.2y, y(1) = 0.5$ у точці $t = 10$.

```
Y_       EQU 0
Y_       INTEGRATE (0.6-0.2#Y_)
PROCEDURE NEWINT(VALUEI)
BEGIN
Y_=VALUEI;
END;
GENERATE   1,,,1
PLUS(NEWINT(0.5))
ADVANCE 9
TERMINATE  1
```

У тексті моделі відсутній блок запуску моделювання. Після команди «Create Simulation» можна також отримати графік, який демонструє роботу моделі. Для цього слід обрати вікно «Plot Window». Зовнішній вигляд заповненого вікна налаштувань наведено на рис. 8.2.

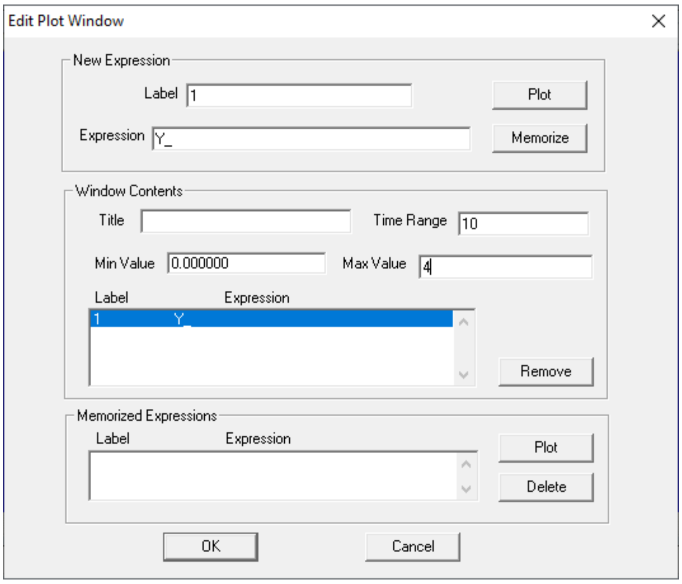

Рисунок 8.2 – Налаштування вікна графіка

Після натискання «ОК» з’явиться вікно «PLOTS». Після цього необхідно задати команду «START» для початку моделювання. Буде побудовано графік, який наведено на рис. 8.3.

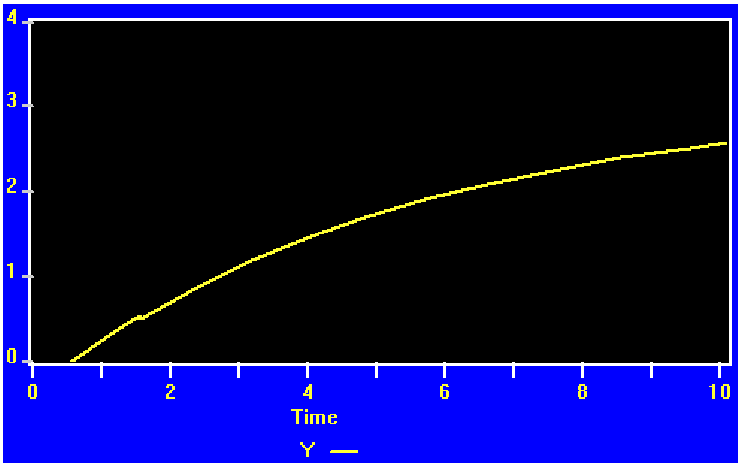

Рисунок 8.3 – Вікно PLOTS

Результат моделювання наведено у вікні «REPORT»:

```
Y_                                   2.587
```

8.2. Індивідуальні завдання

Розробіть модель з використанням зовнішніх потоків даних (вихідні дані та результати) та з PLUS-процедур. Оберіть та обґрунтуйте використовувані закони розподілів, генерування випадкових даних оформити в PLUS-процедурі. Опишіть налаштування, обрані для побудови графіка та сам графік.

1. Модель, яка відображає обробку вхідних запитів користувачів на сервері, з врахуванням обробки даних та затримок. Графік навантаження серверу в часі.

2. Модель для системи черги запитів до серверу, що обробляє великі обсяги даних. Графік часу обробки та кількості очікуючих запитів.

3. Модель тестової системи, яка здійснює автоматичні тести на програмному продукті. Графік розподілу часу на виконання тестів.

4. Модель процесу обробки запитів до веб-сервісів через API, враховуючи черги та затримки. Графік часового навантаження веб-сервісу.

5. Модель для симуляції процесу резервного копіювання даних на сервері з урахуванням часу обробки та навантаження на диск. Графік прогресу копіювання.

6. Модель для симуляції обробки транзакцій в онлайн-банкінгу, з урахуванням часу затримки для кожної транзакції. Графік часу обробки транзакцій.

7. Модель потоку листів, які проходять через поштовий сервер, з урахуванням часу перевірки, відправки та зберігання. Графік навантаження поштового серверу.

8. Модель для симуляції процесу стиснення та розпаковування даних з різними алгоритмами. Графік часу стиснення.

8.3. Контрольні запитання

1. Що таке зовнішні потоки даних?
2. Які операції можна виконувати з потоками даних?
3. Які вікна моделювання є в системі GPSS World?
4. Для чого в GPSS World реалізовані PLUS-процедури?
5. Яким чином формуються PLUS-процедури?

СПИСОК ЛІТЕРАТУРИ

1. 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.

2. Великодний С. С. Моделювання систем : конспект лекцій. Одеса : Одеський державний екологічний університет, 2018. 186 с.

3. Гниленко О. Б. Практикум з імітаційного моделювання комп'ютерних систем в середовищі GPSS World. Дніпро : «Ліра», 2023. 120 с.

4. Жерновий Ю. В. Імітаційне моделювання систем масового обслуговування : Практикум. Львів : Видавничий центр ЛНУ імені Івана Франка, 2007. 307 с.

5. Кравченко І. В., Микитенко В. І., Тимчик Г. С. Комп'ютерне моделювання: системи і процеси : підручник. Київ : КПІ ім. Ігоря Сікорського, 2022. 215 с.

6. Моделювання систем в середовищі GPSS World : навч. посіб. / Я. І. Соколовський [та ін.]. Львів : «Новий Світ – 2000», 2020. 288 с.

7. Павленко П. М., Філоненко С. Ф., Чередніков О. М., Трейтяк В. В. Математичне моделювання систем і процесів : навч. посіб. Київ : НАУ, 2017. 392 с.

8. Поворознюк А. І., Поворознюк О. А., Панченко В. І., Філатова Г. Є. Основи наукових досліджень : навч. посібник. Харків : НТУ "ХПІ", 2024. 199 с.

9. 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

Методичні вказівки
до виконання практичних робіт
з навчальної дисципліни «Проблеми і методи математичного і комп'ютерного моделювання в наукових дослідженнях»
для студентів денної та заочної форми навчання
за спеціальністю «Комп'ютерна інженерія»

Автори:
КОЛОМІЙЦЕВ Олексій Володимирович
ПАНЧЕНКО Володимир Іванович

Відповідальний за випуск проф. Олександр ЗАКОВОРОТНИЙ
Роботу до видання рекомендував проф. Микола ЗАПОЛОВСЬКИЙ

В авторській редакції

План 2024 р., поз. 916

Підп. до друку . Формат 60x84 1/16.
Папір офсет. Друк ризографічний. Ум. друк. арк. 2.3.

Видавничий центр НТУ «ХПІ»,
Свідоцтво про державну реєстрацію ДК № 5478 від 21.08.2017 р.
вул. Кирпичова, 2, м. Харків, 61002

Електронна версія