

$$\times n_c \times \left(A + \frac{4\pi F_{opt}}{n_{ef} \times K^2} \right) = 0.$$

На рис.2 подано залежності $(R/\omega_{\Sigma})_{\min} = f(P, n_{ef})$ для адаптивної (штрихова) та різницево-адаптивної (суцільна лінія) систем з покадровими структурами сигналів для $M_a = 4$ (а) та 9(б).

Різницево-адаптивна система з генератором часових позначок ефективніша від вищерозглянутої в 1,3-1,4 раза при $n_{ef} = 5$ та в 1,05 раза при $n_{ef} = 200$ ($P = 10^{-5}; 10^{-4}$), із зростанням P ефективність зменшується.

Найбільший внесок у сумарну середньоквадратичну відносну похибку при $P = 10^{-5}; 10^{-4}$ робить похибка квантування (в середньому – 80%), при $P = 10^{-3}; 10^{-2}$ близько 70% від сумарної складає сума похибок квантування та появи хибних відліків, причому при збільшенні P їх відношення зростає в бік останньої ($P = 10^{-3} - 2:3; P = 10^{-2} - 1:4$).

1. Калашников И.Д., Степанов В.С., Чуркин А.В. Адаптивные системы сбора и передачи информации. М., 1975.
2. Івахів О.В., Шигера І.Ю. Різницево-адаптивна система з фіксованою кількістю приростів// Вісник ДУ "Львівська політехніка", 1996, №305. С.3-8.
3. Івахів О.В., Пучинский Б.В., Шигера И.Ю. Разностно-адаптивная измерительная система// "Известия ВУЗов. Приборостроение", 1988, т.31. С. 25-29.

УДК 681.142

АВТОМАТИЗАЦІЯ РОЗРАХУНКУ ХАРАКТЕРИСТИК ОБЧИСЛЮВАЛЬНИХ СИСТЕМ НА ОСНОВІ ЗАМКНЕНИХ СТОХАСТИЧНИХ СІТКОВИХ МОДЕЛЕЙ

© Леонід Лукашук, Іван Цмоць, Ігор Шийка
НУ "Львівська політехніка", м. Львів, вул.С. Бандери, 12

Подається опис методики, алгоритмів та програм розрахунку характеристик обчислювальних систем на основі замкнених стохастичних сіткових моделей. Наведено приклади машинного розв'язання.

Description of the method, algorithms and programs of counting systems charac-

teristic calculating on the base of the locked stochastic net models is described in the article. Some examples of machine solve is given.

В основу розрахунку характеристик замкнених моделей обчислювальних систем покладений вираз для ймовірностей станів замкненої мережі [1]

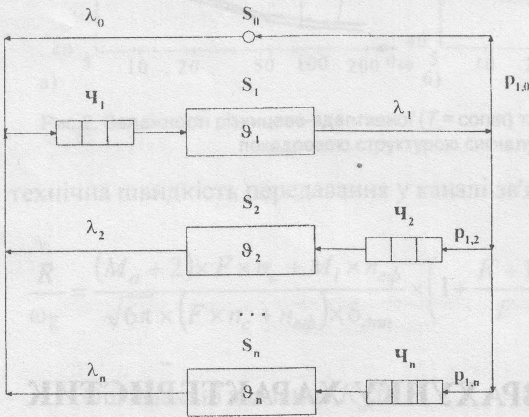
$$Pr(M_1, M_2, \dots, M_n) = \frac{\prod_{j=1}^n R_j(M_j) t_j^{M_j}}{\sum_{A(M,n)} \prod_{j=1}^n R_j(M_j) t_j^{M_j}} \quad (1)$$

ймовірність того, що в першій, другій, ..., n -й СМО знаходиться відповідно M_1, M_2, \dots, M_n заявок

$$R_j(M_j) = \begin{cases} 1/M_j, & \text{при } M_j \leq K_j, \\ (K_j/K_j^{M_j - K_j})^{-1}, & \text{при } M_j > K_j, \end{cases}$$

де $t_j = \alpha_j \cdot \theta_j$ – середнє значення часу обробки однієї заявки за одноразове її перебування в моделі, $\alpha_j = \lambda_j / \lambda_0$ – коефіцієнт передачі, де λ_j – інтенсивність потоку заявок через j -ту СМО, λ_0 – інтенсивність потоку заявок між входом і виходом моделі і є її продуктивність, інакше – потік заявок через фіктивну СМО з нульовим часом обробки; M_j –

кількість заявок в j -й СМО, $A(M, n)$ – символ розподілу M заявок по n пристроях – всі можливі варіанти



$$A_{M,n} = \frac{(M + n - 1)!}{M!(n - 1)!} \quad (2)$$

$Pr(M_j = r)$ – ймовірність того, що в j -й СМО знаходиться r заявок. Обчисливши цю ймовірність за відсутності заявок в СМО, можна визначити коефіцієнт її завантаження,

$$\rho_j = 1 - P_r(M_j = 0) = k_j / K_j, \quad (3)$$

Рис. 1. Замкнена стохастична сіткова модель обчислювальної системи

де ρ_j – коефіцієнт завантаження j -ї СМО, k_j – середня кількість зайнятих каналів в багатоканальній j -й системі, яка визначається таким співвідношенням:

$$k_j = K_j - (K_j - r) \cdot P_r(M_j = r), \quad (4)$$

де K_j – кількість каналів в j -й системі.

$$\lambda_j = \rho_j / \theta_j; \quad (5)$$

$$\lambda_j = k_j / \vartheta_j, \quad (6)$$

де λ_j – інтенсивність потоку заявок для одноканальної (5) і відповідно багатоканальної систем (6).

Середнє число заявок, що перебуває j -й системі,

$$m_j = \sum_{r=0}^M r \cdot \Pr(M_j = r). \quad (7)$$

Для контролю за правильністю розрахунків можна скористатись таким очевидним співвідношенням

$$m_1 + m_2 + \dots + m_n = M, \quad (8)$$

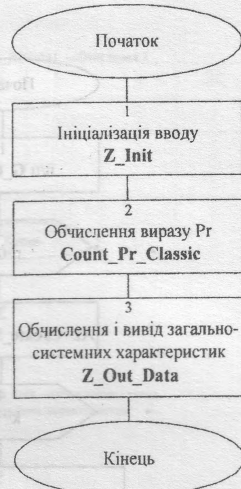
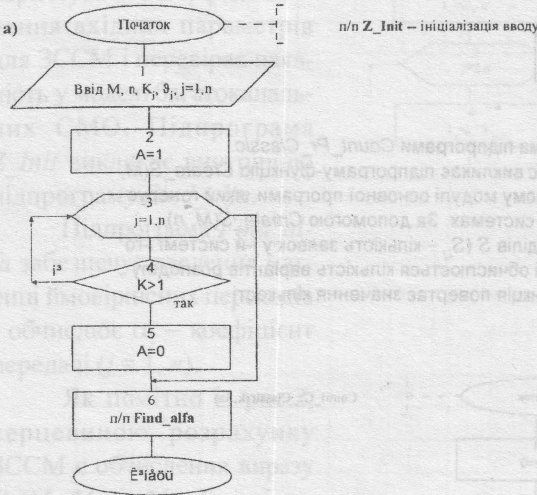
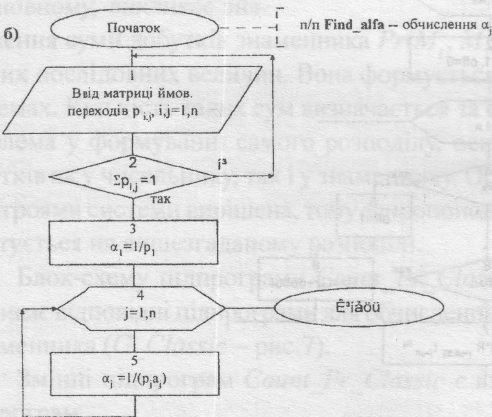


Рис.2. Блок-схема алгоритму розрахунку замкнених моделей



n/n Z_Init – ініціалізація вводу



n/n Find_alfa – обчислення α_i

де M – кількість заявок в обчислювальній системі.

Час циклу системи S_j визначається як проміжок часу між двома послідовними виходами однієї і тієї самої заявки із цієї системи,

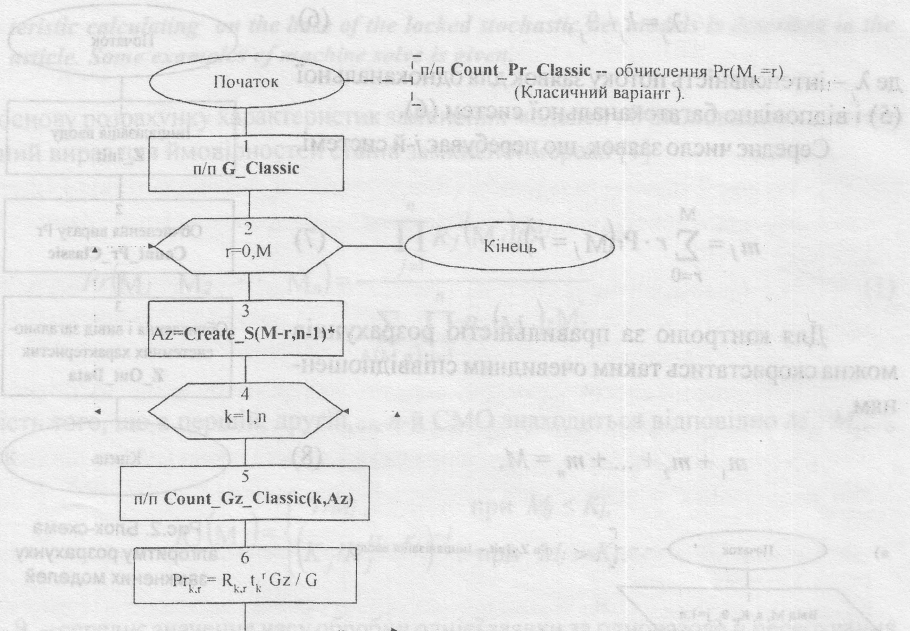
$$U_j = M / \lambda_j, \quad (9)$$

тобто середній час циклу j -ї СМО дорівнює часу проходження через цю систему точно M заявок.

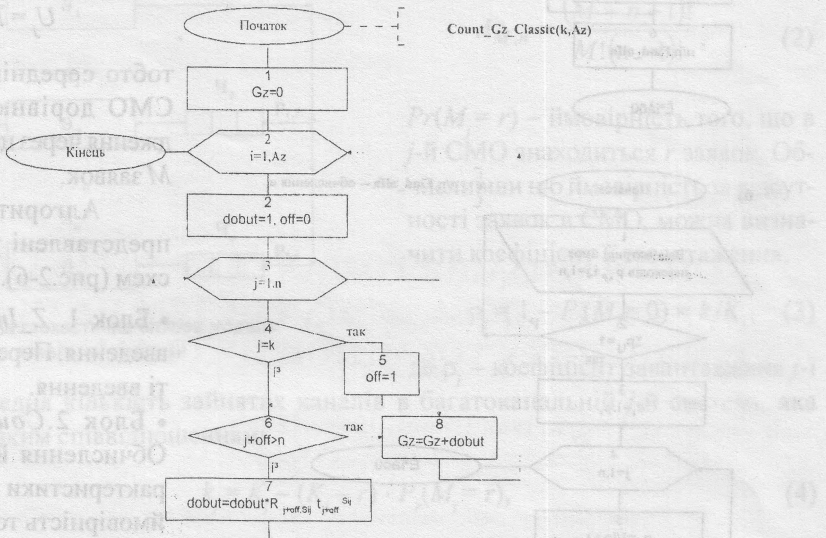
Алгоритми розрахунку представлені у вигляді блок-схем (рис.2-6). Тут:

- Блок 1. *Z_Init*. Ініціалізація введення. Перевірка коректності введення.
- Блок 2. *Count_Pr_Classic*. Обчислення ймовірнісної характеристики $\Pr(M_1, M_2, \dots, M_n)$ ймовірність того, що у першій, другій, ..., n -й СМО знаходиться відповідно M_1, M_2, \dots, M_n заявок.

Рис.3. Блок-схема підпрограми Z_Init

Рис. 4. Блок-схема підпрограми *Count_Pr_Classic*.

Процедура *Count_Pr_Classic* викликає підпрограму-функцію *Create_S(M, n)*, яка знаходиться в окремому модулі основної програми, який генерує розподіл M заявок по n системам. За допомогою *Create_S(M, n)* створюється таблиця розподілів S (S_{ij} – кількість заявок у j -й системі i -го розподілу, $j = 1, n$; $i = 1, Az$) і обчислюється кількість варіантів розподілу $A(M, n) - Az$. Функція повертає значення кількості

Рис. 5. Блок-схема підпрограми *Count_Gz_Classic*

- Блок 3. *Z_Out_Data* - обчислення і виведення кінцевих результатів.

Отже, програма *Z_Model* складається з наступних основних підпрограм:

- *Z_Init*
- *Count_Pr_Classic*
- *Z_Out_Data*

Блок-схему підпрограми *Z_Init* подано на рис.3. *Z_Init* створює зручний для користувача інтерфейс введення вхідних параметрів для ЗССМ і перевіряє наявність у моделі багатоканальних СМО. Підпрограма *Z_Init* викликає внутрішню підпрограму *Find_alfa*.

Підпрограма *Find_alfa* забезпечує введення матриці ймовірнісних переходів і обчислює α_j - коефіцієнт передачі ($j = 1, n$).

Як помітно із рис.2, серцевиною розрахунку ЗССМ є обчислення виразу $Pr(M_1, M_2, \dots, M_n)$. Трудність, в основному, викликає знаходження суми добутків знаменника $Pr(M_1, M_2, \dots, M_n)$. Причому це не є проста сума певних послідовних величин. Вона формується залежно від розподілу M заявок по n системах. Кількість таких сум визначається за відомою формулою (2)[1], але виникає проблема у формуванні самого розподілу, оскільки він необхідний для обчислення добутків як у чисельнику, так і у знаменнику. Оскільки проблема розподілу заявок між пристроями системи вирішена, тому запропоновано класичний метод обчислення, який ґрунтується на вищезгаданому розподілі.

Блок-схему підпрограми *Count_Pr_Classic* подано на рис.4. *Count_Pr_Classic* викликає відповідні підпрограми для обчислення чисельника (*Count_Gz_Classic* - рис.5) і знаменника (*G_Classic* - рис.7).

Змінні підпрограм *Count_Pr_Classic* є вхідними параметрами для внутрішніх підпрограм.

- Підпрограма *Count_Gz_Classic* - обчислення чисельника виразу (1)

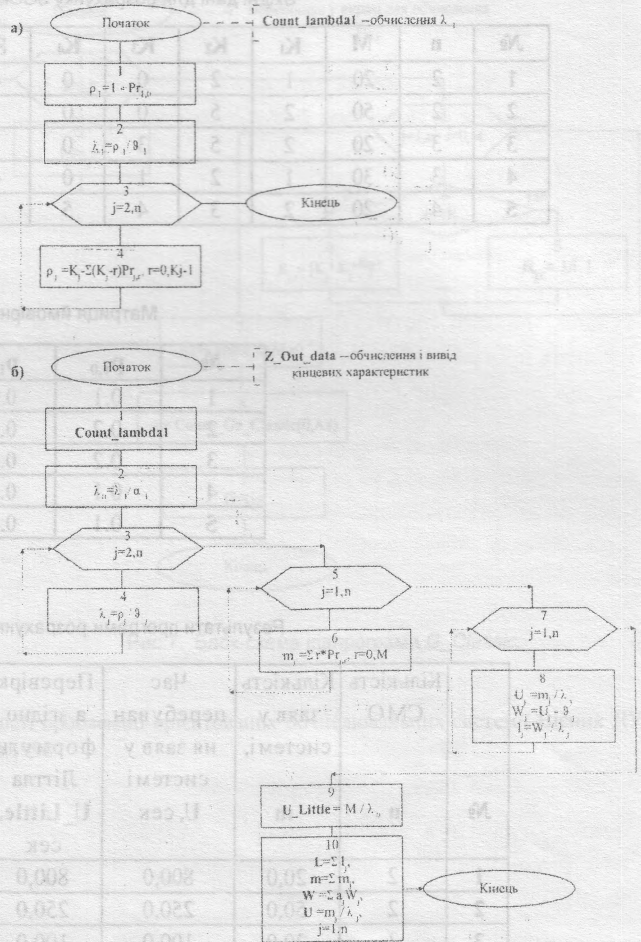


Рис. 6. Блок-схема підпрограми *Z_Out_Data*.

Таблиця 1

Вхідні дані для розрахунку ЗССМ

| № | n | M | K ₁ | K ₂ | K ₃ | K ₄ | g ₁ | g ₂ | g ₃ | g ₄ |
|---|---|----|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| 1 | 2 | 20 | 1 | 2 | 0 | 0 | 4 | 5 | 0 | 0 |
| 2 | 2 | 50 | 2 | 5 | 0 | 0 | 1 | 2 | 0 | 0 |
| 3 | 3 | 20 | 2 | 5 | 3 | 0 | 1 | 2 | 1 | 0 |
| 4 | 3 | 30 | 1 | 2 | 1 | 0 | 4 | 5 | 4 | 0 |
| 5 | 4 | 20 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 |

Таблиця 2

Матриця ймовірнісних переходів

| № | P _{1,0} | P _{1,2} | P _{1,3} | P _{1,4} |
|---|------------------|------------------|------------------|------------------|
| 1 | 0.1 | 0.9 | 0.0 | 0.0 |
| 2 | 0.2 | 0.8 | 0.0 | 0.0 |
| 3 | 0.2 | 0.5 | 0.3 | 0.0 |
| 4 | 0.1 | 0.5 | 0.4 | 0.0 |
| 5 | 0.1 | 0.2 | 0.3 | 0.4 |

Таблиця 3

Результати програми розрахунку ЗССМ

| № | Кількість СМО n | Кількість заяв у системі, m | Час перебування заяв у системі U, сек | Перевірка згідно формули Літла U _{Little} , сек | Продуктивність λ ₀ | Час роботи програми, сек |
|---|--------------------|--------------------------------|--|---|----------------------------------|--------------------------|
| 1 | 2 | 20,0 | 800,0 | 800,0 | 0,025 | 0,05 |
| 2 | 2 | 50,0 | 250,0 | 250,0 | 0,200 | 0,11 |
| 3 | 3 | 20,0 | 100,0 | 100,0 | 0,200 | 0,38 |
| 4 | 3 | 30,0 | 1200,0 | 1200,0 | 0,025 | 0,88 |
| 5 | 4 | 20,0 | 200,1 | 200,1 | 0,100 | 5,11 |

- Підпрограма *G_Classic* – обчислення знаменника виразу (1).

Блок-схему підпрограми *Z_Out_Data* подано на рис.6. *Z_Out_Data* викликає підпрограму *Count_lambda1*, а також обчислює і виводить кінцеві характеристики.

Підпрограма *Count_lambda1* обчислює величину λ_1 – інтенсивність заявок на вхід першого пристрою.

Нижче наведено приклади розрахунку програми (табл.1-3).

Висновки

1. Трудомісткість розроблених програм задовільна і, наприклад, для розглянутих задач при розв'язанні на *IBM PC 80386DX, 4Mb RAM, 16MHz* становить лічені секунди (див. табл.3).

2. Для нормальної роботи даної програми кількість можливих СМО децю обмежена і визначається складністю задачі, але ж це обмеження таке, що для реальних задач не є істотним, бо у будь-якому випадку допускає моделювання при значній кількості пристроїв та заявок.

1. Основи теорії вычислительных систем. Под ред. С.А.Майорова. М., 1978.

2. Лукашук Л.О. Оптимальний розподіл швидкості між пристроями обчислювальної системи. МОУ ІСДО ДУ "ЛП". Київ 1995.

4. Лукашук Л.О. Оцінка продуктивності проблемно-орієнтованих обчислювальних систем// Вісник ДУ "Львівська політехніка". №351, Львів 1998.

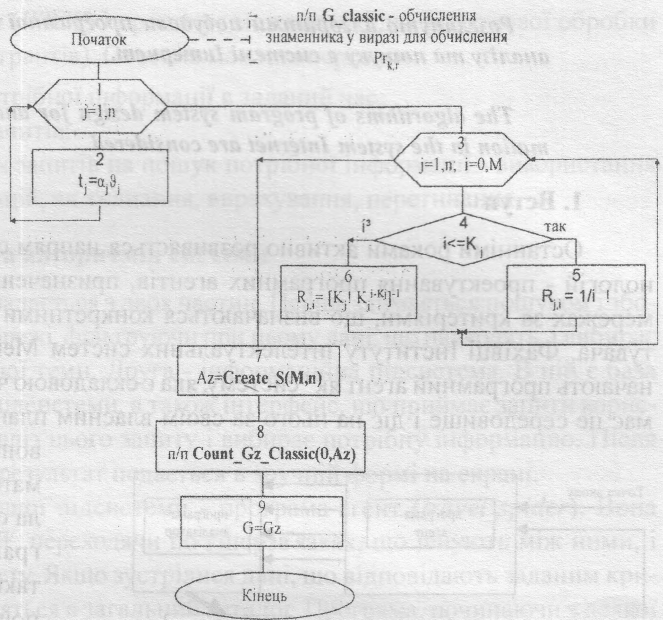


Рис.7. Блок-схема підпрограми G_Classic.

УДК 681.142.2

ПРОЕКТУВАННЯ ПРОГРАМНОГО АГЕНТА ДЛЯ ІНФОРМАЦІЙНОГО ПОШУКУ ГРАНТІВ У МЕРЕЖІ INTERNET

© Євдоким Федорчук, Олексій Шпек, Юрій Шпек
НУ "Львівська політехніка", м. Львів, вул.С. Бандери, 12