УДК 004.2.004.8

В.Г. Зайцев, М.В. Плахотний, Є.І. Цибаєв

Національний технічний університет України «Київський політехнічний інститут», Київ

МАТЕМАТИЧНА МОДЕЛЬ ДЛЯ ОЦІНЮВАННЯ ЧАСУ ВИКОНАННЯ ПРОГРАМ

Запропоновано модель роботи програми у вигляді поглинаючого марковського ланцюга із дискретними станами і дискретним часом. Розроблено метод і алгоритм розрахунку середнього часу виконання програми та середньої кількості виконання окремих блоків. Наведено результати експериментальних досліджень

Ключові слова: модель програми, поглинаючий марковський ланцюг, час виконання програми

Предложена модель работы программы в виде поглощающей марковской цепи с дискретными состояниями и дискретным временем. Разработан метод и алгоритм расчета среднего времени выполнения программы и среднего количества выполнения отдельных блоков. Приведены результаты экспериментальных исследований

Ключевые слова: модель программы, поглощающая марковская цепь, время выполнения программы

The suggested model, the work programme in the form of an absorbing Markov chain with discrete States and discrete time. A method and algorithm of calculation of the average execution time and the average number of performance of separate blocks. The results of experimental investigations

Keywords: model programmes, absorbing Markov chain, the execution of the program

Вступ

Програмне забезпечення (ПЗ) спеціалізованих комп’ютерних систем має ряд особливостей, серед яких: виконання завдань у реальному часі; незмінність функцій ПЗ під час усього періоду експлуатації; необхідність обміну інформацією із зовнішнім абонентом.

Ці особливості на стадії проектування ПЗ [1] потребують виконати його оптимізацію та оцінити часові характеристики програм: середній час виконання; середню частоту виконання окремих блоків; перевірити можливість зациклювань і т.п. Існує декілька підходів до розв´язання цих задач: створення декількох варіантів програмної системи і на основі реальних даних вибір оптимального варіанта; імітаційне моделювання; створення і дослідження математичних моделей.

Перший підхід може виявитися дуже затратним та неприпустимим щодо часу розробки.

Другий підхід у більшості випадків не може бути рекомендований на ранніх стадіях проектування, оскільки не можна точно визначити вхідні параметри моделей і розрахувати відповідні навантаження окремих блоків обчислюваної системи. Отже, створення і дослідження математичних моделей функціонування ПЗ є дуже перспективним підходом.

Мета статті

Метою роботи є створення математичної моделі функціонування ПЗ, за допомогою якої можна оцінювати такі характеристики програми (комплексу програм): відсутність можливості зациклювання; середній час виконання; середня частота виконання окремих блоків.

Опис математичної моделі

Поставимо у відповідність блок-схемі алгоритму програми орієнтований граф $G(i)$, у якому вершини i будуть співставлені блокам і розгалуженням, а орієнтованим ребрам – шляхи переходу алгоритму між окремими блоками. Будемо вважати, що процесу виконання програми будуть відповідати переходи з однієї вершини до іншої, а знаходженню у певній вершині – виконанню деякої команди або блоку програми. Як приклад на рис. 1, *а* наведено блок-схему алгоритму програми пошуку максимального елемента в одновимірному масиві, а на рис. 1, *б* – відповідний граф $G(i)$.

Зауважимо, що граф $G(i)$ може бути для алгоритмів будь-якого рівня: рівень команд, модулів, процедур, операторів мови проектування програм PDL [1] і т.п.

Використовуючи таку модель, можна розглядати виконання програми як існування деякого випадкового процесу $S(t)$ з дискретними станами і з дискретним або неперервним часом, якому відповідає граф $G(i)$. Зважаючи на особливості виконання програм, можна вважати, що переходи процесу $S(t)$ із стану i в стан j не залежать від того, яким чином він потрапив у стан i, а тільки залежать від перехідних імовірностей P_{ij}, якими навантажують (розмічають) відповідні ребра графа $G(i)$.

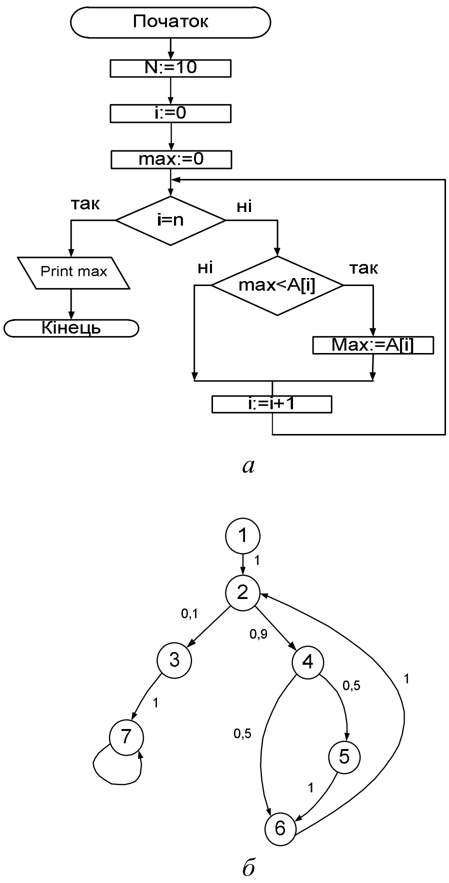


Рис. 1. Блок-схема алгоритму програми та можлива її інтерпретація графом $G(i)$

З урахуванням цього можна інтерпретувати процес виконання програми як випадковий марковський процес з дискретними станами, які пов'язані у марковський ланцюг матрицею перехідних імовірностей P.

Нагадаємо, що випадковий процес $S(t)$, який відбувається у системі, називається процесом з дискретним часом, якщо перехід з одного стану в інший відбувається тільки у попередньо визначені моменти часу $t(1)$, $t(2)$, що називаються кроками такого процесу. У проміжках між кроками система S зберігає свій стан. При цьому не виключається, що на деяких кроках система не буде змінювати свій стан. Як відомо [2], випадковий процес з дискретним часом можна представити випадковою послідовністю (по індексу k) таких подій

$$S_i(k), \quad \overline{i=1,n}; \quad \overline{k=1,2\ldots}, \qquad (1)$$

де n – кількість вершин графа $G(i)$.

Оскільки система $S_{(i)}$ у довільний момент часу t може перебувати тільки в одному із станів $S_1,\ldots,S_n$, то при кожному $k=1.2$ події $S_1(k),\ldots,S_n(k)$ несумісні і утворюють повну групу подій. Основними характеристиками марковських ланцюгів з дискретним часом є імовірність станів

$$P_i(k) = P(S_i(k)), \ \overline{i=1,n}; \ \overline{k=1,2\ldots},$$

та імовірності переходів P_{ij}, що утворюють квадратну матрицю переходів порядку $n-P$.

$$P = \begin{bmatrix} P_{11}, P_{12}, \ldots, P_{1n} \\ P_{21}, P_{22}, \ldots, P_{2n} \\ P_{n1}, P_{n2}, \ldots, P_{nn} \end{bmatrix}. \qquad (2)$$

$$\sum_{i=1}^{n} P_i(k) = 1; \ k = 1,2,\ldots, \qquad (3)$$

$$\sum_{j=1}^{n} P_{ij} = 1; \ \overline{i=1,n}. \qquad (4)$$

Наявність на розміченому графі стрілок та визначень відповідних імовірностей переходів вказує на відмінність їх від нуля. Відсутність стрілок говорить про те, що відповідні імовірності дорівнюють нулю. Якщо сума імовірностей, що виходять з певної вершини менше 1, це свідчить про імовірність затримки на певному кроці P_{ii}, яка дорівнює

$$P_{ii} = 1 - \sum_{j=1}^{n} P_{ij}; \quad i \neq j \cdot \qquad (5)$$

Якщо $P_{ii} = 1$, це свідчить про те, що процес потрапив у, так званий, поглинаючий стан, з якого він вже вийти не може, блукання по станах завершується. Вектор – рядок станів у початковий момент дискретного часу $t(0) - P^{(0)} = (P_1(0), P_2(0), \ldots, P_n(0))$ називають вектором початкового розподілення імовірностей.

Для наведеного прикладу (рис.1,б) він дорівнює$(1,0,0,0,0,0,0)$. Як відомо з [3], для визначення вектора - рядка імовірностей переходу від кроку k до кроку $k+1$ слід взяти добуток вектора - рядка переходу від стану $k-1$ до k на матрицю перехідних імовірностей **P**, тобто

$$P^{(k)} = P^{(k-1)}\mathbf{P}, \qquad (6)$$

у загальному вигляді. Як правило, тривалість часу між моментами переходу розглядається, як стала величина, тобто, $t(i)-t(i-1) = const$ ..

$$p^{(k)} = p^{(0)}\mathbf{P}^k$$

або [2]

$$P_i(k) = \sum_{j=1}^{n} P_j(k-1)P_{ji}; \quad \overline{i=1,n} \qquad (7)$$

Пов'яжемо тривалість між моментами переходу з попереднім станом процесу Будемо вважати, що якщо процес переходить із стану S_i в стан S_j, то такий перехід можливий після завершення проміжку часу τ_i - знаходження процесу у стані S_i. Власне кажучи, у такому разі розглядаємо випадковий процес з дискретним часом, у якого переходи відбуваються через попередньо визначені, але нерівномірні інтервали часу.

Визначимо величину проміжку часу $T(k)$ між сусідніми кроками. Ця величина буде дорівнювати

$$T(k) = t(k) - t(k-1), \qquad (8)$$

або $T(k)$ буде відповідати формулі

$$T(k) = \sum_{i=1}^{n} P_i(k-1)\tau_i \qquad (9)$$

при нерівномірній дискретизації та $T(k) = \tau_i = const$ при рівномірній дискретизації. Якщо $\tau_i = 1$, то

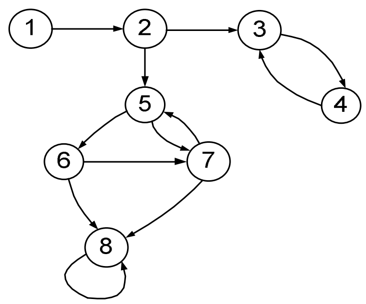
$$T(k) = \sum_{i=1}^{n} P_i(k-1) = 1, \qquad (10)$$

що збігається з результатом (3).

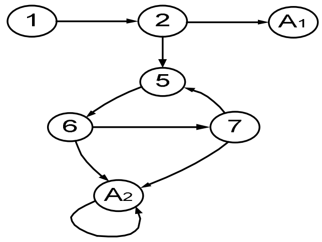
Якщо кроки нерівномірні за часом, то величина $T(k)$ визначається за формулою (9) і буде залежатиме від τ_i.

Кожний алгоритм або програма мають початок і кінець. Таким чином, ця необхідна умова відповідає структурі графа $G(i)$. Як відомо з [3], перетворюючи алгоритм або програму у граф $G(i)$, можна попередньо побудувати діаграму порядку G_1, оскільки вона являє собою граф без контурів. При цьому у графі G_1 можуть існувати тупикові вершини – вершини, з яких дуги не виходять за межі контуру. Підграфи, що відповідають тупиковим вершинам, називають ергодичними класами марковського ланцюга. Якщо ергодичний клас складається з однієї вершини, то такий стан називають поглинаючим. Наприклад, на рис. 2 наведено граф $G(i)$ та його діаграма порядку. З діаграми порядку видно, що у графі $G(i)$ існує два ергодичні класи:

$$A_1 = \{3,4\} \; та \; A_2 = \{8\}.$$



а



б

Рис. 2. Граф переходів $G(i)$ – *а*, та його діаграма порядку – *б*

Клас A_2 є поглинаючим.

Під час блукання по графу з імовірністю 1, траєкторія марковського ланцюга опиниться в одному з ергодичних класів і вийти з нього не зможе. Якщо наведений граф $G(i)$ відповідає деякій конкретній програмі, то можна казати, що початку програми відповідає вершина 1, кінцю - вершина 8, ергодичний клас A_1 відповідає помилці «зациклюванню» програми, яка має бути виправлена, а A_2 – поглинаючий.

Метод використання моделі програми

За допомогою моделі програми у вигляді поглинаючого марковського ланцюга можна обчислити кількість звертань до окремих блоків програми, середній час виконання програми, середню кількість кроків до переходу у поглинаючий стан, що свідчить про завершення програми. Ці характеристики програм можна у подальшому використовувати для застосування методів профілювання коду. Як відомо, для визначення вказаних характеристик програм застосовувались методи імітаційного моделювання [4] і метод, заснований на побудові фундаментальної матриці, кожний з елементів якої дорівнює середній кількості раз потрапляння системи в той чи інший стан до поглинання [4].

Обидва методи досить складні у реалізації та мають проблеми із пошуком зворотних розріджених матриць $n \times n$. Тут запропоновано метод обчислень, побудований на використанні безпосередньо математичної моделі марковського ланцюга із дискретними станами та дискретним часом. Метод полягає у послідовному обчисленні імовірностей станів

$$P_i(k) \; при \;\; k \to \infty . \qquad (10)$$

Критерієм завершення процедури обчислень служить:

$$P_n(k) \to 1, \; при \;\; k \to \infty , \qquad (11)$$

якщо P_n відповідає поглинаючому стану.

Поняття $k \to \infty$ є математичною абстракцією. Як доведено на практиці, для відповідних розрахунків достатньо обрати $P_n\,(k) \approx 0,99999$.

Позначимо середню кількість кроків до поглинання через K. Тоді середній час виконання K кроків до завершення програми (попадання у поглинаючий стан) T можна обчислити так:

$$T = \sum_{j=1}^{K} T(j) \cdot \qquad (12)$$

Якщо уявити, що поглинаючий стан винесено за «кінець» алгоритму, то час його виконання τ_n в розрахунках має дорівнювати 0. Тоді з урахуванням (8) час T визначається так:

$$T = \sum_{j=1}^{K} \sum_{i=1}^{n} P_i(j-1)\tau_i \cdot \qquad (13)$$

Якщо уявити, що $\tau_i = const = 1; \quad \overline{i = 1, n-1}; \quad \tau_n = 0;$ то з урахуванням (9)

$$T = \sum_{j=1}^{K} T\;(j) = K \;\cdot \qquad (14)$$

Визначимо загальний час перебування системи у стані $S_i - T_i$.

На кожному кроці система знаходиться у стані S_i, середній час $P_i(j-1)\;\tau_i$.

Тоді за K кроків загальний час T_i визначається так:

$$T_i = \sum_{j=1}^{K} P_i(j-1)\tau_i \cdot \qquad (15)$$

У середньому, кількість разів потрапляння системи у стан $S_i - K_i$ буде становити:

За результатами досліджень створена спеціальна програма обчислення вказаних характеристик.

Наведемо приклад розрахунку параметрів T, K та K_i програми, наведеної на рис. 1

$$K_i = T_i \;/\; \tau_i \;, \qquad (16)$$

або

$$K_i = \sum_{j=1}^{K} P_i(j-1) \cdot \qquad (17)$$

Не важко бачити, що $K = \sum_{i=1}^{n} K_i$ (18)

	1	2	3	4	5	6	7
1	0	1	0	0	0	0	0
2	0	0	0.1	0.9	0	0	0
3	0	0	0	0	0	0	1
4	0	0	0	0	0.5	0.5	0
5	0	0	0	0	0	1	0
6	0	1	0	0	0	0	0
7	0	0	0	0	0	0	1

Рис. 3. Матриця перехідних імовірностей P_{ij}

Результати розрахунків для випадку: матриця перехідних імовірностей, як на рис. 3; $\tau_1 = 10$, $\tau_2 = 20$, $\tau_3 = 50$, $\tau_4 = 20$, $\tau_5 = 10$, $\tau_6 = 20$.

Результати: T = 665, K = 34.5, K_1 = 1, K_2 = 10, K_3 = 1, K_4 = 9, K_5 = 4,5, K_6 = 9.

Висновки

Створено математичну модель і запропоновано метод обчислення часових характеристик програм.

Згідно запропонованого методу, основною операцією обчислень є операція – добуток вектора на матрицю n x n, що достатньо просто реалізується для великих n, особливо зважаючи на те, що відповідні матриці сильно розріджені і для обчислень можна застосовувати спеціальні методи [5].

Список літератури

1. *Зелковиц М., Шоу А., Геннон Дж. Принципы разработки программного обеспечения. – М.: Мир, 1982. – 368 с.*
2. *Вентцель Е.С. Исследование операций. – М.: Советское радио, 1972. – 552 с.*
3. *Романовский И.В. Дискретный анализ: Учебное пособие для студентов, 3-е изд, – СПб.: Невский диалект; БХВ Петербург, 2003. – 320 с.*
4. *Бережная Е.В. Математические методы моделирования экономических систем. – М.: Финансы и статистика, 2001. – 368 с.*
5. *Тьюарсон Р. Разряженные матрицы. – М.: Мир, 1977. – 192 с.*

Стаття надійшла до редколегії 19.09.2013

Рецензент: д-р техн. наук, проф. С.В. Цюцюра, Київський національний університет будівництва і архітектури, Київ.