УДК 519.718

К.т.н, доцент Романкевич В.О., магістрант Морозов К.В.

Національний технічний університету України
«Київський політехнічний інститут»

БАЗОВА *K(3,N)* GL-МОДЕЛЬ, ЯКА БУДУЄТЬСЯ МЕТОДОМ ЗСУВУ ЗМІННИХ

Abstract

Vitaliy O. Romankevich, assoc. prof., PhD; Kostyantyn Morozov, student
Basic K (3, n) GL-model that is built by the shift of variables

In this paper a new basic GL-model that models a system resistant to failure of three components is proposed. The basic properties of the proposed model are discussed. The method of optimization of calculations of edge functions for this model is proposed.

Вступ

В наш час все більш важливою стає необхідність побудови складних систем керування, що мають дуже високі показники надійності. При цьому необхідно мати спосіб розрахунку надійності цих систем. Але зважаючи на складність цих систем, ця задача стає досить непростою.

Складність розрахунку надійності таких систем аналітичними способами стає неприпустимо високою. Тому найчастіше в таких випадках використовується інший підхід, що базується на проведенні статистичних експериментів з моделями, які відтворюють поведінку системи в потоці відмов. В літературі запропоновано декілька типів моделей, кожні з яких мають свої переваги та недоліки.

На кафедрі СКС НТУУ «КПІ» були запропоновані нові, так звані графо-логічні, або GL-моделі [1], що можуть бути успішно використані для моделювання поведінки досить широкого кола складних систем в потоці відмов. GL-модель являє собою неорієнтовані граф G, кожному ребру якого відповідає своя булева функція. Аргументами реберних функцій є індикаторні змінні x_i ($i = 1,\dots,n$, де n – кількість компонентів системи), рівні 1 якщо і-й компонент системи працездатний або 0 в іншому випадку. Ребро видаляється з графа G, якщо відповідна йому реберна функція приймає нульове значення. Зв'язність графа відображає роботоздатність ВБС в цілому.

Базовою ВБС К(n, m) називається така багатопроцесорна система, яка складається з m процесорів і залишається роботоздатною при відмові n та менше процесорів.

В [2] була запропонована так звана 2р-модель, що є базовою GL-моделлю на базі циклічного графу, причому вона втрачає лише 2 ребра якщо відмовив $m + 1$ елемент системи, яку представляє модель.

Алгоритм побудови даної моделі базується на багаторазовому поділі множини змінних на дві частини і операції склеювання та описаний в [2].

Постановка задачі

В зв'язку з тим, що реальні системи досить часто є відмінними від базових, постає задача модифікації базових моделей. Вона може бути вирішена як шляхом зміни структури графа, так і шляхом модифікації реберних функцій. Наявність у кожній з функцій усіх змінних може дещо спростити методи їх модифікації, тому необхідно розробити базову GL-модель, кожна з реберних функцій якої містить усі вхідні змінні. Іншим критерієм є складність, а отже і час розрахунку значень реберних функцій. Другою задачею є створення методу оптимізації часу розрахунку реберних функцій для розроблених моделей.

Опис моделі

Модель будується на базі циклічного графа із $n - 2$ ребрами, де n – кількість змінних у моделі (або кількість процесорів у системі). Кожна із реберних функцій умовно може бути представлена у вигляді диз'юнкції трьох компонентів: $f_i = A_i \vee B_i \vee C_i$. При цьому компонент B_i завжди включає в себе тільки одну змінну. Компоненти A_i , B_i та C_i представляються таким чином:

$$A_i = \bigwedge_{j=1}^{i} x_j$$

$$B_i = x_{i+1}$$

$$C_i = \bigwedge_{j=i+2}^{n} x_j,$$

При такому алгоритмі побудові очевидно, що кількість реберних функцій, а отже і кількість ребер циклічного графа моделі дорівнює $n - 2$, як і для аналогічної *2р*-моделі [2].

Порівняємо, наприклад, реберні функції базової *K(3,8)* моделі, побудованої даним методом та *2р*-моделі.

Для моделі, побудованої методом послідовного зсуву змінних у виразах реберних функцій маємо:

$$f_1 = x_1 \vee x_2 \vee x_3x_4x_5x_6x_7x_8$$
$$f_2 = x_1x_2 \vee x_3 \vee x_4x_5x_6x_7x_8$$
$$f_3 = x_1x_2x_3 \vee x_4 \vee x_5x_6x_7x_8$$
$$f_4 = x_1x_2x_3x_4 \vee x_5 \vee x_6x_7x_8$$
$$f_5 = x_1x_2x_3x_4x_5 \vee x_6 \vee x_7x_8$$
$$f_6 = x_1x_2x_3x_4x_5x_6 \vee x_7 \vee x_8$$

Для *2p*-моделі маємо:

$$f_1 = x_1 \vee x_2 \vee x_3x_4$$
$$f_2 = x_1x_2 \vee x_3 \vee x_4$$
$$f_3 = (x_1 \vee x_2)(x_1x_2 \vee x_3x_4)(x_3 \vee x_4) \vee x_5x_6x_7x_8$$
$$f_4 = x_1x_2x_3x_4 \vee (x_5 \vee x_6)(x_5x_6 \vee x_7x_8)(x_7 \vee x_8)$$
$$f_5 = x_5 \vee x_6 \vee x_7x_8$$
$$f_6 = x_5x_6 \vee x_7 \vee x_8$$

Як бачимо, крайні функції *2p*-моделі мають меншу складність у порівнянні із функціями запропонованої моделі, на відміну від середніх функцій, які є більш складними.

Алгоритм побудови запропонованої моделі, очевидно, буде значно простішим, ніж для аналогічної *2p*-моделі.

Властивості моделі

У кожній з функцій запропонованої моделі є $n-1$ логічна операція, оскільки кожна функція містить у собі n змінних. Ребер та реберних функцій в свою чергу $n-2$, таким чином, у реберних функціях моделі $(n-1)\cdot(n-2) = n^2-3n+2$, а складність моделі: $o(n^2)$. Цей показник є гіршим, ніж для *2p*-моделі.

Для запропонованої моделі, як і для аналогічної *2p*-моделі виконується така властивість: якщо в модель подати вектор із k нулями, то кількість реберних функцій, що приймають значення 0 буде: $m = k-2$ [3]. Цю властивість досить легко довести.

Доведення:

Помітимо, що для того, щоб реберна функція $f_i = A_i \vee B_i \vee C_i$ прийняла значення 0 необхідно і достатньо, щоб A_i, B_i та C_i прийняли нульове значення. Очевидно, що для того, щоб A_i дорівнював 0, необхідно, щоб деякий $x_j, j \leq i$ мав нульове значення. Для C_i необхідно, щоб будь-який з $x_j, j \geq i+2$ прийняв нульове значення. А для B_i необхідно, щоб нулю дорівнював x_{i+1}.

З вищенаписаного можемо зробити висновок: для того, щоб i-та реберна функція прийняла нульове значення необхідно і достатньо, щоб нулю дорівнював x_{i+1}, а також хоча б по одному з x_j, що мають індекс більший та менший, ніж *i+1*. Ця умова виконуватиметься для всіх j, для яких $x_j = 0$, окрім тих, які мають найбільше та найменше значення. Очевидно, що таких j, та відповідних їм реберних функцій буде всього k-*2,* якщо k – загальна кількість нулів у векторі.

Іще однією властивістю є кількість різних векторів з трьома нулями, при яких кожна з реберних функцій приймає нульове значення. Зауважимо, що змінні x_j, що входять до A_i, B_i та C_i не повторюються. Тому, для того, щоб реберна функція прийняла нульове значення на векторі із трьома нулями необхідно, щоб одна (і саме одна!) з нульових змінних входила у вираз A_i, ще одна – у B_i, та ще одна – у C_i. Кількість варіантів вибору змінної, що входить до A_i дорівнює кількості елементів, з яких складається цей вираз, тобто i. Для B_i, очевидно, можлива лише одна змінна, що приведе його у нульове значення. Для C_i, аналогічно до A_i, варіантів вибору *n-i-1*.

Таким чином, для f_i кількість векторів з трьома нулями, при яких вона приймає нульове значення дорівнює $l_i = i \cdot (n - i - 1)$.

Мінімальне значення цей вираз приймає для $i = 1$ та $i = n - 2$:

$$l_1 = l_{n-2} = n - 2.$$

Максимальне значення вираз приймає при $i = \frac{n}{2}$:

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

Оптимізація часу розрахунку значень реберних функцій

Розглянемо у загальному випадку вигляд реберної функції запропонованої моделі:

$$f_i = \bigwedge_{j=1}^{i} x_j \vee x_{i+1} \vee \bigwedge_{k=i+2}^{n} x_k$$

Введемо дві послідовності, що визначаються рекурентно: $\alpha_1 = x_1$, $\alpha_{i+1} = \alpha_i \wedge x_{i+1}, i > 1$, а також $\beta_1 = x_n$, $\beta_{i+1} = \beta_i \wedge x_{n-i+1}, i > 1$.

Тоді вирази реберних функцій можуть бути записані таким чином:

$$f_i = \alpha_i \vee x_{i+1} \vee \beta_{n-i-1}.$$

Розрахунок всіх α_i для $1 \le i \le n - 2$ потребує виконання $n - 3$ елементарних логічних операцій (кон’юнкцій). Аналогічно і розрахунок всіх β_i для $1 \le i \le n - 2$ також потребує виконання $n - 3$ елементарних логічних операцій (кон’юнкцій). Таким чином, на розрахунок всіх α_i та β_i

для $1 \leq i \leq n-2$ необхідно $2n-6$ елементарних логічних операцій. Якщо ж додатково оптимізувати алгоритм розрахунку, то після отримання першого значення 0 подальші кон'юнкції можуть не проводитися. Таким чином, кількість операцій може додатково зменшитися.

Розрахунок значення кожної реберної функції при відомих α_i та β_i потребує дві елементарні логічні операції (диз'юнкції). Оскільки реберних функцій загалом $n-2$, то кількість елементарних операцій: $2n-4$.

Таким чином, для розрахунку всіх реберних функцій моделі запропонованим методом необхідно $(2n-6)+(2n-4)=4n-10$ елементарних логічних операцій.

Розрахуємо значення затримки при апаратній реалізації розрахунку реберних функцій за допомогою комбінаційної схеми з використанням запропонованої оптимізації. Нехай час затримки одного вентиля – τ. Очевидно, що значення α_i буде отримане із затримкою $(i-1)\cdot\tau$. Аналогічно і значення β_i буде отримане із затримкою $(i-1)\cdot\tau$. Для функції $f_i = \alpha_i \vee x_{i+1} \vee \beta_{n-i-1}$ найбільша затримка буде у α_i, якщо $i \geq \frac{n-1}{2}$ або у β_i, якщо $i \leq \frac{n-1}{2}$. Таким чином, найбільша затримка від компонентів α_i і β_i буде $(n-3)\cdot\tau$ для f_1 та f_{n-2}. Найменша затримка буде для $f_{\left\lceil\frac{n-1}{2}\right\rceil}$, а саме $\left(\left\lceil\frac{n-1}{2}\right\rceil-1\right)\cdot\tau$. Додатково при отриманні значення функції елементами АБО буде внесена затримка 2τ (при використанні двовходових елементів) або τ (при використанні трьохвходових елементів). Таким чином, мінімальна затримка буде для $f_{\left\lceil\frac{n-1}{2}\right\rceil}$, а саме $\left(\left\lceil\frac{n-1}{2}\right\rceil+1\right)\cdot\tau$, а максимальна – для f_1 та f_{n-2} і матиме значення $(n-1)\cdot\tau$ (при використанні двовходових елементів АБО).

Висновки

Таким чином: запропонована модель має ту саму кількість реберних функцій, що і аналогічна *2p*-модель. Також для неї, як і для *2p*-моделі, виконується властивість втрачати на векторах із *k* нулями рівно *k – 2* ребра. Обчислювальна складність реберних функцій моделі є дещо вищою, ніж для *2p*-моделі. Проте, алгоритм побудови такої моделі є набагато простішим. Розподіл векторів із трьома нулями, що перетворюють реберні функції в 0, є в деяких випадках більш зручним, ніж для *2p*-моделі. Кількість змінних та складність кожної з реберних функцій даної моделі є постійною.

При оптимізації розрахунку значень реберних функцій складність розрахунку реберних функцій становить $4n-10$. При апаратній реалізації розрахунку реберних функцій за допомогою комбінаційної схеми з використанням запропонованої оптимізації мінімальна затримка розповсюдження сигналу становить $\left(\left\lceil\frac{n-1}{2}\right\rceil+1\right)\cdot\tau$, а максимальна – $(n-1)\cdot\tau$, де τ – час затримки одного схемного елемента.

Література

1. *Романкевич А.М.* Графо-логические модели для анализа сложных отказоустойчивых вычислительных систем / А.М. Романкевич, Л.Ф. Карачун, В.А. Романкевич // Электронное моделирование. - 2001.- т.23, №1.- с.102-111.

2. *Романкевич В. А.* GL-модель поведения отказоустойчивых многопроцессорных систем с минимальным числом теряемых рёбер В. А. Романкевич, Е. Р. Потапова, Бахтари Хедаятоллах, В. В. Назаренко // Вісник НТУУ “КПІ” Інформатика, управління та обчислювальна техніка. - 2006. - № 45. - С. 93-100.

3. *Майданюк И.В.* Об одном свойстве GL модели с минимальным числом теряемых ребер / И.В. Майданюк, К.В. Морозов, Е.Р. Потапова, А.В. Шурига // Науковий вісник Чернівецького університету. Серія : Комп’ютерні системі та компоненти. – 2010. – Т.1, № 2. – С.72-81.