УДК 519.718

В. В. Різник[1], М. Т. Соломко[2]
[1]Національний університет "Львівська політехніка",
кафедра автоматизованих систем управління,
[2]Національний університет водного господарства та природокористування, м. Рівне,
кафедра обчислювальної техніки

КОМБІНАТОРНИЙ МЕТОД МІНІМІЗАЦІЇ БУЛЕВИХ ФУНКЦІЙ

Розглянуто нову процедуру алгебри логіки – суперсклеювання змінних, яка здійснюється за наявності у структурі таблиці істинності повної бінарної комбінаторної системи з повторенням або неповної бінарної комбінаторної системи з повторенням. Ефективність алгебричної операції суперсклеювання змінних істотно спрощує алгоритм мінімізації булевих функцій, що уможливлює мінімізацію функцій з кількістю змінних до 10.

Ключові слова: булева функція, метод мінімізації, мінімізація логічної функції, блок-схема з повторенням, мінтерми, суперсклеювання змінних.

V. V. Riznyk[1], M. T. Solomko[2]
[1]Lviv Polytechnic National University, Department of Control Aided Systems
[2]National University of Water and Environmental Engineering,
Rivne, Department of Computer Engineering

COMBINATORIAL METHOD OF MINIMIZING BOOLEAN FUNCTIONS

Considered the new procedure of logic algebra – super-sticking of variables, which is carried out in the presence of a truth table in complete binary combinatorial system with a repetition or incomplete of the system considered. The efficiency of the algebraic operation of super-sticking variables greatly simplifies the algorithm for minimizing boolean functions, which allows manual minimization of the functions.

Key words: boolean function, method of minimizing, minimization of logic function, block-design with repetition, minterms, super-sticking of variables.

Вступ

Мінімізація булевих функцій все ще популярна у різних сферах цифрових технологій, таких як дизайн PLA, вбудований самотест (BIST), проектування систем управління тощо. Проблема мінімізації ДНФ є однією з багатоекстремальних логіко-комбінаторних задач і зводиться до оптимального зменшення кількості логічних елементів вентильної схеми без втрати її функціональності. Зазначимо, що у загальній постановці ця задача дотепер не розв'язана, однак добре досліджена у класі диз'юнктивно-кон'юнктивних нормальних форм.

Недоліки відомих методів мінімізації булевих функцій пов'язані зі стрімким збільшенням обсягу обчислень, наслідком чого є зростання розрядності обчислювальних операцій, і, отже, збільшення кількості змінних логічної функції. Наприклад, карта Карно зазвичай важко піддається розпізнаванню, якщо змінних більше від чотирьох–п'яти, тому цей метод недоцільно використовувати з більше ніж шістьма змінними. Незважаючи на вищу досконалість методу

Квайна – Мак-Класкі порівняно з картами Карно, його практичне застосування також обмежене з-за експоненціального зростання часу обчислення зі збільшенням кількості змінних. Можна показати, що для функції від *n* змінних верхня межа кількості основних імплікант становить $3^n \ln(n)$ [1]. Наприклад, для *n* = 32 кількість основних імплікант може перевершувати $6,5 \times 10^{15}$.

Від результату мінімізації булевої функції залежить швидкодія обчислювального пристрою, його надійність та енергозбереження. Особливості комбінаторного методу мінімізації [2, 3] полягають у більшій інформативності розв’язання задачі порівняно з алгебричним способом мінімізації функції за рахунок табличної організації та впровадження апарату образного перетворення. Об’єктом розв’язання задачі мінімізації булевої функції комбінаторним методом є блок-схема з повторенням, властивості якої, своєю чергою, дають змогу доповнити правила алгебри логіки новими правилами спрощення логічної функції. Алгоритм мінімізації булевої функції є однією з центральних та практично важливих проблем, яка постає під час проектування обчислювальних пристроїв. У зв’язку з цим вивчення нових правил алгебри логіки, встановлення їх властивостей актуальне для спрощення алгоритму мінімізації булевої функції без втрати її функціональності в разі збільшення кількості змінних.

Аналіз публікацій і окреслення проблеми

Умови логічного зведення до мінімуму булевої функції, поданої у ДНФ, розглядаються у [4]. Якщо функція задовольняє такі умови, то для її спрощення застосовують класичний алгоритм мінімізації Квайна–Мак-Класкі, що допускає автоматизацію. Зазначається, що кількість змінних функції для коду програми обмежується пам’яттю комп’ютера. Автори публікації [4] описують метод оптимізації, коли процес не тільки передбачає пошук еквівалентного логічного виразу, але й залучає визначення конкретних умов, за яких логічні вирази можна ще більше скоротити. Ці типи елементів у логічному дизайні розглядаються як “ступінь свободи”. У таких випадках користувач може оптимізувати заданий дизайн на підставі ступеня свободи. Тому пошук альтернативних рішень бажаний, оскільки він може забезпечити оптимальний булевий вираз у підсумку. У публікації [5] розглянуто узагальнені правила спрощення кон’юнктермів у поліноміальному теоретико-множинному форматі, які ґрунтуються на запропонованих теоремах для різних початкових умов перетворення пари кон’юнктермів, гемінгова відстань між якими може бути довільна. Зазначені правила можуть бути корисними для мінімізації у поліноміальному теоретико-множинному форматі довільних логічних функцій від *n* змінних. Ефективність запропонованих правил демонструється прикладами мінімізації функції, запозиченими з робіт відомих авторів з метою порівняння результатів. З огляду на порівняльні приклади запропоновані правила дають підставу для підтвердження доцільності застосування їх у процедурах мінімізації будь-якої логічної функції від *n* змінних у поліноміальній формі. У роботі [6] подана мінімізація булевої функції з використанням таблиці істинності, в якій послідовно зменшується одна змінна, поки всі змінні не вичерпаються. У стандартному методі таблицю істинності (ТІ) готують за заданою логічною функцією. Тоді функція виражається як сума мінімальних умов, що відповідають наборам змінних, на яких функція набуває значення одиниці. Нарешті, ця функція зменшується за допомогою булевих ідентичностей. Отже, всі спрощення концентруються в одному місці після ТІ. Ця процедура не завжди приводить до мінімальної реалізації. У роботі [6] розглянуто спрощення, що наприкінці кожного етапу здійснює скорочення ТІ. Показано, що метод є системним і, безумовно, веде до мінімальної функції. Він простіший в експлуатації, ніж на основі тільки булевих топонімів, карт Карно, Quine-McClusky, та може обробляти будь-яку кількість змінних. Це пояснюється декількома прикладами. Алгоритм і програма для мінімізації комбінаційних логічних функцій до 20 змінних подані у [7], де кількість змінних обмежується пам’яттю комп’ютерної системи. Алгоритм оснований на послідовній кластеризації термів, починаючи з групування термів з однією зміною. Алгоритм кластеризації закінчується тоді, коли змінні не можуть більше бути згруповані. Цей алгоритм аналогічний до алгоритму Квайна – Мак-Класкі, але він простіший, оскільки усуває низку дій алгоритму Квайна–Мак-Класкі. У роботі [8] демонструється метод логіко-мінімізаційного стиснення зображень, який залежить від логічної функції. Процес мінімізації розглядає сусідні пікселі зображення як роз'єднані мінтерми, що представляють логічну функцію, та стискає 24-розрядні кольорові зображення за допомогою процедури мінімізації функції. Коефіцієнт стиснення такого методу у середньому на 25 % більший порівняно з відомими методами стиснення зображень. У роботі [9] продемонстровано використання генетичного алгоритму для вибору побічних об'єктів процедури мінімізації логічної функції за допомогою карти Карно. У [10] запропоновано новий евристичний алгоритм для максимальної мінімізації булевих функцій. Для реалізації запропонованого алгоритму використовуються графічні дані й визначено умови для досягнення максимального рівня мінімізації булевої функції.

На відміну від публікацій [3–10], у цій роботі об'єктом розв'язання задачі мінімізації булевої функції є комбінаторна блок-схема з повторенням, а об'єктом спрощення процесу мінімізації – нова алгебрична операція – суперсклеювання змінних, яка здійснюється за наявності у структурі таблиці істинності повної або неповної бінарних комбінаторних систем з повторенням. Процедура скорочення повної досконалої диз'юнктивної нормальної форми (ДДНФ) логічної функції дає одиницю. А оскільки повна ДДНФ однозначно визначає повну бінарну комбінаторну систему з повторенням і навпаки, це дає підставу видаляти всі блоки повної бінарної комбінаторної системи з повторенням з таблиці істинності, структура якої дає змогу виконувати правила суперсклеювання змінних. Математичний апарат блок-схеми з повторенням дає змогу отримати більше інформації стосовно ортогональності, суміжності, однозначності блоків таблиці істинності (комбінаторної системи). Рівносильні перетворення графічними образами, що за властивостями мають більшу інформаційну ємність, спроможні ефективно замінити вербальні процедури алгебричних перетворень.

Мета та завдання дослідження

Метою роботи є спрощення комбінаторного методу мінімізації булевої функції за допомогою нової алгебричної операції – суперсклеювання змінних та встановлення властивостей такої процедури.

Для досягнення поставленої мети необхідно розв'язати такі задачі:

1) встановити адекватність застосування алгебричної операції суперсклеювання змінних для мінімізації булевої функції;

2) визначити властивості операції суперсклеювання змінних з використанням структур повної та неповної бінарних комбінаторних систем з повторенням;

3) здійснити верифікацію комбінаторного методу із застосуванням правила суперсклеювання змінних та встановити оцінку складності алгоритму мінімізації булевої функції комбінаторним методом;

4) провести порівняльний аналіз продуктивності та зменшення складності алгоритму мінімізації булевих функцій, отриманих за допомогою правила суперсклеювання змінних, з прикладами мінімізації функції іншими методами.

Бінарна комбінаторна система з повторенням

Якщо задана деяка множина A, то множина всіх її підмножин, що мають k елементів $M_k(A)$ і N всіх k-елементних підмножин множини із n елементів, дорівнює $N(M_k(A)) = C_n^k = \dfrac{n!}{k!(n-k)!}$.

Крім того, $$\sum_{k=0}^{n} C_n^k = 2^n \qquad (1)$$

<u>*Приклад 1.*</u> Нехай A = {a, b, c}. Тоді

$$M(A) = \{\{a\},\{b\},\{c\},\{a,b\},\{a,c\},\{b,c\},\{a,b,c\},\varnothing\};$$

$$M_2(A) = \{\{a,b\},\{a,c\},\{b,c\}\}.$$

Оскільки C_n^k – кількість k-елементних підмножин множини з n елементів, то сума у лівій частині відомого виразу (1) дорівнює кількості всіх підмножин.

Приклад 2. За формулою (1) легко обчислити кількість всіх підмножин множини $A = \{a, b, c, d\}$:

$$N(M(A)) = C_4^0 + C_4^1 + C_4^2 + C_4^3 + C_4^4 = 1 + 4 + 6 + 4 + 1 = 16 = 2^4.$$

Зазначимо, що множина $A = \{a, b, c, d\}$, крім перерахунку своїх елементів, може визначати номери позицій, на яких розміщений елемент α. Так, наприклад, a може означати першу, b – другу позицію множини $A = \{a, b, c, d\}$ і т. д. Підмножинами множини $A = \{a, b, c, d\}$ у такому випадку будуть підмножини, що містять елемент α на k позиціях, k = 0, …, n, де n – кількість позицій множини A. У загальному випадку елемент α може займати декілька позицій на множині A, тому елемент α повторюється на множині A.

Нехай α = 1, тоді позиції, на яких відсутній елемент α, слід позначати нулями.

Приклад 3. Для множини $A = \{a, b, c, d\}$, що визначає номери позицій, приймемо α = 1. Тоді підмножини множини А будуть мати такий вигляд:

$$\begin{array}{ll} (0,0,0,0); & (1,0,0,0); \\ (0,0,0,1); & (1,0,0,1); \\ (0,0,1,0); & (1,0,1,0); \\ (0,0,1,1); & (1,0,1,1); \\ (0,1,0,0); & (1,1,0,0); \\ (0,1,0,1); & (1,1,0,1); \\ (0,1,1,0); & (1,1,1,0); \\ (0,1,1,1); & (1,1,1,1). \end{array} \qquad (2)$$

Кількість всіх k-елементних підмножин множини $A = \{a, b, c, d\}$, що визначає номери позицій, обчислюється за формулою (1):

$$N(M_0(A)) = C_4^0 = 1,\ \ N(M_1(A)) = C_4^1 = 4,\ \ N(M_2(A)) = C_4^2 = 6,$$
$$N(M_3(A)) = C_4^3 = 4,\ \ N(M_3(A)) = C_4^4 = 1.$$
$$N(M(A)) = N(M_0(A)) + N(M_1(A)) + N(M_2(A)) + N(M_3(A)) + N(M_4(A)) = 16.$$

Конфігурація (2) – повна комбінаторна система з повторенням елемента α, яку назвемо

2–(n,b)-design,

де n – розрядність блока системи; b – кількість блоків повної системи, що визначається за формулою $b = 2^n$, число 2 перед дужками означає бінарну структуру конфігурації (2). Наприклад, 2–(4,16)-design є повною бінарною комбінаторною системою з повторенням, що складається з чотирирозрядних блоків, загальна кількість блоків – 16.

Алгебрична операція суперсклеювання змінних

Комбінаторні властивості блок-схеми з повторенням дають змогу доповнити правило алгебри логіки склеювання змінних [2] правилом суперсклеювання змінних.

Для чотирирозрядної логічної функції правило суперсклеювання змінних має такий вигляд:

– перше правило:

$$\begin{vmatrix} 0 & 0 & 0 & x \\ 0 & 0 & 1 & x \\ 0 & 1 & 0 & x \\ 0 & 1 & 1 & x \\ 1 & 0 & 0 & x \\ 1 & 0 & 1 & x \\ 1 & 1 & 0 & x \\ 1 & 1 & 1 & x \end{vmatrix} = x; \tag{3}$$

– друге правило:

$$\begin{vmatrix} 0 & 0 & x & y \\ 0 & 1 & x & y \\ 1 & 0 & x & y \\ 1 & 1 & x & y \end{vmatrix} = xy; \tag{4}$$

– третє правило:

$$\begin{vmatrix} 0 & x & y & z \\ 1 & x & y & z \end{vmatrix} = xyz. \tag{5}$$

Перше правило використовує 2–(3,8)-design, друге 2–(2,4)-design, третє 2–(1,2)-design.

Процедура скорочення повної досконалої диз’юнктивної нормальної форми (ДДНФ) логічної функції дає одиницю. Наприклад, скорочення трирозрядної повної ДДНФ виглядає так:

$$\begin{aligned} &\overline{x_1}\,\overline{x_2}\,\overline{x_3} + \overline{x_1}\,\overline{x_2}x_3 + \overline{x_1}x_2\overline{x_3} + \overline{x_1}x_2x_3 + x_1\overline{x_2}\,\overline{x_3} + x_1\overline{x_2}x_3 + x_1x_2\overline{x_3} + x_1x_2x_3 = \\ &= \overline{x_1}\,\overline{x_2}(\overline{x_3} + x_3) + \overline{x_1}x_2(\overline{x_3} + x_3) + x_1\overline{x_2}(\overline{x_3} + x_3) + x_1x_2(\overline{x_3} + x_3) = \\ &= \overline{x_1}\,\overline{x_2} + \overline{x_1}x_2 + x_1\overline{x_2} + x_1x_2 = \overline{x_1}(\overline{x_2} + x_2) + x_1(\overline{x_2} + x_2) = \overline{x_1} + x_1 = 1 \end{aligned}$$

Оскільки повна ДДНФ однозначно визначає повну комбінаторну систему з повторенням 2 – (*n*, *b*)-design, і навпаки, це дає підставу видаляти всі блоки повної комбінаторної системи з матриць, які демонструють правила суперсклеювання (3)–(5). Після цього, застосувавши закон ідемпотентності до змінної, що залишилась – *x* (*xy*; *xyz*), отримуємо результат скорочення за правилом суперсклеювання змінних.

Правило (5) проявляє просте склеювання змінних та є частковим випадком правил суперсклеювання (3) та (4).

Змінні *x*, *y*, *z*, що утворюють повну комбінаторну систему з повторенням 2 – (*n*, *b*)-design, можуть займати будь-який розряд мінтерма логічної функції.

Аналогічно до правил суперсклеювання змінних для функції на чотири змінні, можна подати правила суперсклеювання для функцій п’яти змінних [3] і більше.

У загальному випадку конфігурація таблиці істинності заданої функції, крім підматриці повної комбінаторної системи з повторенням 2–(*n*,*b*)-design, вміщує й підматриці неповної комбінаторної системи з повторенням

2–(*n*, *x/b*)-design,

де *x* – кількість блоків неповної комбінаторної системи з повторенням. Властивості неповної комбінаторної системи з повторенням 2–(*n*, *x/b*)-design дають змогу також встановлювати правила, що забезпечують ефективну мінімізацію булевих функцій [3]. Наприклад:

$$\left|\begin{array}{cccc} x & \bar{y} & 0 & 1 \\ x & \bar{y} & 1 & 0 \\ x & \bar{y} & 1 & 1 \\ x & y & 0 & 1 \\ x & y & 1 & 0 \\ x & y & 1 & 1 \end{array}\right| = \left|\begin{array}{cccc} x & \bar{y} & & 1 \\ x & \bar{y} & 1 & \\ x & y & & 1 \\ x & y & 1 & \end{array}\right| = \left|\begin{array}{ccc} x & & 1 \\ x & 1 & \end{array}\right|. \tag{6}$$

$$\left|\begin{array}{ccccc} 0 & 0 & 0 & \bar{x} & y \\ 0 & 0 & 1 & \bar{x} & y \\ 0 & 1 & 0 & \bar{x} & y \\ 0 & 1 & 1 & \bar{x} & y \\ 1 & 0 & 0 & x & y \\ 1 & 0 & 1 & x & y \\ 1 & 1 & 0 & x & y \\ 1 & 1 & 1 & x & y \end{array}\right| = \left|\begin{array}{ccc} 0 & \bar{x} & y \\ 1 & x & y \end{array}\right|. \tag{7}$$

Зазначені правила утворюють бібліотеку правил для процесу мінімізації булевих функцій як стандартні процедури, тому застосування окремого такого правила для змінних булевої функції зводиться до здійснення одного алгебричного перетворення.

Мінімізація чотирирозрядних булевих функцій

Приклад 4. Мінімізувати логічну функцію $F(x_1, x_2, x_3, x_4)$ комбінаторним методом, яка задана таблицею істинності (табл. 1) [11, с. 184].

Таблиця 1

Таблиця істинності логічної функції $F(x_1, x_2, x_3, x_4)$

№ з/п	x_1	x_2	x_3	x_4	F	№ з/п	x_1	x_2	x_3	x_4	F
0	0	0	0	0	**1**	8	1	0	0	0	**0**
1	0	0	0	1	**1**	9	1	0	0	1	**1**
2	0	0	1	0	**1**	10	1	0	1	0	**1**
3	0	0	1	1	**1**	11	1	0	1	1	**1**
4	0	1	0	0	**0**	12	1	1	0	0	**0**
5	0	1	0	1	**0**	13	1	1	0	1	**1**
6	0	1	1	0	**1**	14	1	1	1	0	**1**
7	0	1	1	1	**0**	15	1	1	1	1	**1**

Складемо досконалу диз’юктивну нормальну форму (ДДНФ) заданої функції з блоків, за яких функція набуває значення одиниці, тобто для наборів 0, 1, 2, 3, 6, 9, 10, 11, 13, 14, 15.

$$F(x_1, x_2, x_3, x_4) = \bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4 + \bar{x}_1\bar{x}_2\bar{x}_3x_4 + \bar{x}_1\bar{x}_2x_3\bar{x}_4 + \bar{x}_1\bar{x}_2x_3x_4 + \bar{x}_1x_2x_3\bar{x}_4 +$$
$$+ x_1\bar{x}_2\bar{x}_3x_4 + x_1\bar{x}_2x_3\bar{x}_4 + x_1\bar{x}_2x_3x_4 + x_1x_2\bar{x}_3x_4 + x_1x_2x_3\bar{x}_4 + x_1x_2x_3x_4.$$

Перший крок – це склеювання, заміщення та узагальнене склеювання змінних. З множини варіантів мінімізації, отриманих на першому кроці, розглянемо два варіанти мінімізації чотирирозрядної функції.

Перший варіант: мінімізація функції з використанням правила суперсклеювання змінних за наявності повної бінарної комбінаторної системи з повторенням 2–(n, b)-design.

$$F=\left|\begin{array}{c|cccc} 0 & 0 & 0 & \mathbf{0} & \mathbf{0} \\ 1 & 0 & 0 & \mathbf{0} & \mathbf{1} \\ 2 & 0 & 0 & \mathbf{1} & \mathbf{0} \\ 3 & 0 & 0 & \mathbf{1} & \mathbf{1} \\ 6 & 0 & 1 & 1 & 0 \\ 9 & 1 & \mathbf{0} & \mathbf{0} & \mathbf{1} \\ 10 & 1 & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ 11 & 1 & \mathbf{0} & \mathbf{1} & \mathbf{1} \\ 13 & 1 & \mathbf{1} & \mathbf{0} & \mathbf{1} \\ 14 & 1 & \mathbf{1} & \mathbf{1} & \mathbf{0} \\ 15 & 1 & \mathbf{1} & \mathbf{1} & \mathbf{1} \end{array}\right| = \left|\begin{array}{cccc} 0 & 0 & & \\ 0 & 1 & 1 & 0 \\ 1 & & & 1 \\ 1 & & 1 & \end{array}\right| = \left|\begin{array}{cccc} 0 & 0 & & \\ 0 & & 1 & 0 \\ 1 & & & 1 \\ 1 & & 1 & \end{array}\right| = \left|\begin{array}{cccc} 0 & 0 & & \\ & & 1 & 0 \\ 1 & & & 1 \\ 1 & & 1 & \end{array}\right| = \left|\begin{array}{cccc} 0 & 0 & & \\ & & 1 & 0 \\ 1 & & & 1 \end{array}\right|. \qquad (8)$$

До блоків 0–3 першої матриці блок-схеми (8) застосовано правило суперклеювання (4), наявна повна комбінаторна система з повторенням 2–(2, 4)-design (виділена жирним шрифтом); блок 6 не змінюється. До блоків 9–15 застосовано правило (6). Тут є неповна бінарна комбінаторна система з повторенням 2–(3, 6/8)-design (виділена жирним шрифтом). Результат образного перетворення записано у другу матрицю блок-схеми (8).

Алгебричні перетворення другої матриці (результат перетворення записаний у третю матрицю):

▪ заміщення змінних у першому та другому блоках другої матриці блок-схеми (8):

$$\overline{x}_1\overline{x}_2+\overline{x}_1x_2x_3\overline{x}_4=\overline{x}_1(\overline{x}_2+x_2x_3\overline{x}_4)=$$
$$=\overline{x}_1(\overline{x}_2+x_3\overline{x}_4)=\overline{x}_1\overline{x}_2+\overline{x}_1x_3\overline{x}_4),$$

$$\begin{array}{cccc} 0 & 0 & & \\ 0 & 1 & 1 & 0 \end{array} \rightarrow \begin{array}{cccc} 0 & 0 & & \\ 0 & & 1 & 0. \end{array}$$

Алгебричні перетворення третьої матриці, результат якої записано у четверту матрицю:

▪ заміщення змінних у другому та четвертому блоках матриці блок-схеми (8):

$$\overline{x}_1x_3\overline{x}_4+x_1x_3=x_3(\overline{x}_1\overline{x}_4+x_1)=$$
$$=x_3(\overline{x}_4+x_1)=x_1x_3+x_3\overline{x}_4,$$

$$\begin{array}{cccc} 0 & & 1 & 0 \\ 1 & & 1 & \end{array} \rightarrow \begin{array}{cccc} & & 1 & 0 \\ 1 & & 1 & \end{array}$$

Алгебричні перетворення четвертої матриці, результат якої записаний у п’яту матрицю:

▪ узагальнене склеювання змінних другого, третього та четвертого блоків четвертої матриці блок-схеми (8):

$$x_3\overline{x}_4+x_1x_4+x_1x_3=x_3\overline{x}_4+x_1x_4.$$

$$\begin{array}{cccc} & & 1 & 0 \\ 1 & & & 1 \\ 1 & & 1 & \end{array} \rightarrow \begin{array}{cccc} & & 1 & 0 \\ 1 & & & 1 \end{array}$$

У підсумку отримуємо мінімальну функцію (9):

$$F = \overline{x_1}\,\overline{x_2} + x_1 x_4 + x_3 \overline{x_4}. \tag{9}$$

Другий варіант: мінімізацію функції з використанням правила суперсклеювання змінних за наявності неповної бінарної комбінаторної системи з повторенням 2–(*n*, *x*/*b*)-design розглянуто у [3].

Другий крок – верифікація отриманої мінімізованої функції (9) за допомогою вихідної таблиці істинності (табл. 1).

Мінімізована логічна функція (9) задовольняє вихідну таблицю істинності (табл. 1).

У табл. 2 подано результати мінімізації функції $F(x_1, x_2, x_3, x_4)$ за допомогою ациклічного графу [11, с. 184] та комбінаторним методом.

Таблиця 2

Результат мінімізації функції $F(x_1, x_2, x_3, x_4)$

За допомогою ациклічного графу	Комбінаторним методом
$F = \overline{x_1}\,\overline{x_2} + x_1 x_4 + x_1 x_3 \overline{x_4} + \overline{x_1} x_2 x_3 \overline{x_4}$	$F = \overline{x_1}\,\overline{x_2} + x_1 x_4 + x_3 \overline{x_4}$

З табл. 2 бачимо, що комбінаторний метод дає функцію з меншою кількістю вхідних змінних.

Приклад 5. Мінімізувати логічну функцію

$F(x_1, x_2, x_3, x_4)$ =(0,0,1,1,1,0,1,1,1,1,1,1,0,0,0,1) [12] комбінаторним методом.

Складемо таблицю істинності заданої чотирирозрядної функції з блоків, за яких функція набуває значення одиниці, тобто для наборів: 2, 3, 4, 6, 7, 8, 9, 10, 11, 15, та здійснимо мінімізацію.

$$F = \left|\begin{array}{c|cccc} 2 & 0 & 0 & 1 & 0 \\ 3 & 0 & 0 & 1 & 1 \\ 4 & 0 & 1 & 0 & 0 \\ 6 & 0 & 1 & 1 & 0 \\ 7 & 0 & 1 & 1 & 1 \\ 8 & 1 & 0 & \mathbf{0} & \mathbf{0} \\ 9 & 1 & 0 & \mathbf{0} & \mathbf{1} \\ 10 & 1 & 0 & \mathbf{1} & \mathbf{0} \\ 11 & 1 & 0 & \mathbf{1} & \mathbf{1} \\ 15 & 1 & 1 & 1 & 1 \end{array}\right| = \left|\begin{array}{cccc} 0 & 0 & 1 & \\ 0 & 1 & & 0 \\ 1 & 0 & & \\ & 1 & 1 & 1 \end{array}\right| = \left|\begin{array}{cccc} & 0 & 1 & \\ 0 & 1 & & 0 \\ 1 & 0 & & \\ & 1 & 1 & 1 \end{array}\right| = \left|\begin{array}{cccc} & 0 & 1 & \\ 0 & 1 & & 0 \\ 1 & 0 & & \\ & & 1 & 1 \end{array}\right|.$$

До блоків 8–11 першої матриці застосовано правило суперсклеювання змінних, оскільки є комбінаторна система 2–(2,4)-design (виділена жирним шрифтом). В інших блоках першої матриці застосовано просте склеювання змінних. В останніх двох матрицях виконано заміщення (неповне склеювання) змінних.

У підсумку отримуємо мінімальну функцію:

$$F = x_1 \overline{x_2} + \overline{x_1} x_2 \overline{x_4} + \overline{x_2} x_3 + x_3 x_4.$$

У табл. 3 наведено результати мінімізації функції $F(x_1, x_2, x_3, x_4)$ за допомогою паралельного розщеплення кон’юнктермів [12] та комбінаторним методом.

Таблиця 3

Результат мінімізації функції $F(x_1, x_2, x_3, x_4)$

Методом паралельного розщеплення кон'юнктермів	Комбінаторним методом
$F = x_1\overline{x_2} + \overline{x_1}x_2\overline{x_4} + \overline{x_1}x_3 + x_3x_4$	$F = x_1\overline{x_2} + \overline{x_1}x_2\overline{x_4} + \overline{x_2}x_3 + x_3x_4$

З табл. 3 бачимо, що обидві функції мають однакові параметри і проходять верифікацію, хоч відрізняються складом змінних у третій імпліканті. Приклад 5 демонструє меншу обчислювальну складність мінімізації булевої функції комбінаторним методом.

Приклад 6. Мінімізувати систему чотирирозрядних булевих функцій f_1, f_2, f_3 [12] комбінаторним методом.

$$\begin{cases} f_1 = 2, 5, 6, 13, 14 \\ f_2 = 5, 7, 13, 14 \\ f_3 = 2, 6, 7, 13, 15 \end{cases}$$

Складемо таблицю істинності заданої системи чотирирозрядних функції з блоків, за яких функція набуває значення одиниці (табл. 4).

Таблиця 4

Таблиця істинності системи булевих функції f_1, f_2, f_3

№	x_1	x_2	x_3	x_4	f_1	f_2	f_3
2	0	0	1	0	**1**	**0**	**1**
5	0	1	0	1	**1**	**1**	**0**
6	0	1	1	0	**1**	**0**	**1**
7	0	1	1	1	**0**	**1**	**1**
13	1	1	0	1	**1**	**1**	**1**
14	1	1	1	0	**1**	**1**	**0**
15	1	1	1	1	**0**	**0**	**1**

Є два підходи до мінімізації системи булевих функцій від *n* змінних: 1) мінімізацію здійснюють окремо для кожної функції; 2) сумісна мінімізація системи, коли метод мінімальної системи використовує загальні кон'юнктерми окремих функцій.

Усунення надлишкових кон'юнктермів у окремій функції не гарантує усунення надлишковості у самій системі. З іншого боку, сумісна мінімізація системи не завжди може бути кращою. Тому для деяких систем функцій доводиться застосовувати обидва методи мінімізації. Сумісна мінімізація системи є громіздкішою порівняно з першим методом.

Для сумісної мінімізації об'єднуємо всі різні кон'юнктерми окремих функцій у функцію Y системних кон'юнктермів

$$Y = 0010_{(1,3)} + 0101_{(1,2)} + 0110_{(1,3)} + 0111_{(2,3)} + 1101_{(1,2,3)} + 1110_{(1,2)} + 1111_{(3)}.$$

Системним кон'юнктермом називається мінтерм булевої функції з індексами, які показують, до яких функцій він належить [12]. Серед системних кон'юнктермів розрізняють тотожні елементи – з однаковими індексами, та нетотожні – з неоднаковими індексами, але перетин їх не порожній. Наприклад, $(101)_{2,4}$ і $(111)_{2,4}$ утворюють тотожний елемент $(1\text{-}1)_{2,4}$, а $(101)_{2,4}$ і $(001)_4$ утворюють нетотожний елемент $(–01)_4$ [12].

Функцію Y подамо таблицею істинності.

Для сумісної мінімізації системи застосуємо такі правила:

– склеювання змінних у системних кон'юнктермах функції *Y* здійснюється тільки для тих кон'юнктермів, які мають хоча б один загальний індекс;

– результату склеювання кон'юнктермів присвоюється множина індексів, яка є перетином вихідних множин індексів склеюваних кон'юнктермів;

– якщо кон’юнктерми не мають спільних індексів, склеювання не відбувається;

– тотожні кон’юнктерми склеюються з іншими тотожними кон’юнктермами;

– нетотожні кон’юнктерми склеюються з іншими нетотожними кон’юнктермами.

Після операції склеювання нетотожні кон’юнктерми переносяться до наступної таблиці для подальшої мінімізації, крім випадку, коли індекси результату склеювання збігаються з індексами одного з нетотожних кон’юнктермів. Поглинання одного кон’юнктерма іншим здійснюється тільки за умови збігу множин індексів двох кон’юнктермів.

$$Y = \left|\begin{array}{ccccc} \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{(1,3)} \\ \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{(1,3)} \\ 0 & 1 & 0 & 1 & (1,2) \\ 0 & 1 & 1 & 1 & (2,3) \\ 1 & 1 & 0 & 1 & (1,2,3) \\ 1 & 1 & 1 & 0 & (1,2) \\ 1 & 1 & 1 & 1 & (3) \end{array}\right| = \left|\begin{array}{ccccc} \mathbf{0} & & \mathbf{1} & \mathbf{0} & \mathbf{(1,3)} \\ 0 & 1 & & 1 & (2) \\ & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{(1,2)} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{(1,2)} \\ 1 & 1 & 1 & 0 & (1,2) \\ & 1 & 1 & 1 & (3) \\ 0 & 1 & 1 & 1 & (2,3) \\ 1 & 1 & 0 & 1 & (1,2,3) \\ 1 & 1 & & 1 & (3) \end{array}\right| = \left|\begin{array}{ccccc} 0 & & 1 & 0 & (1,3) \\ 0 & 1 & & 1 & (2) \\ & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{(1,2)} \\ 1 & 1 & 1 & 0 & (1,2) \\ & 1 & 1 & 1 & (3) \\ 0 & 1 & 1 & 1 & (2,3) \\ 1 & 1 & 0 & 1 & (1,2,3) \\ 1 & 1 & & 1 & (3) \end{array}\right|.$$

Третя матриця представляє тупикову ДНФ функції Y. (Перетворення у матрицях функції Y див. [3]). Далі завдання пошуку мінімальної ДНФ функції Y вирішується на підставі таблиці покриття Б. Рицара [12] (табл. 5), у якій необхідно вилучити усі зайві прості імпліканти.

Таблиця 5

Таблиця покриття функції $Y(x_1, x_2, x_3, x_4)$

	$01 : 1_{(2)}$		$01 : 1_{(2)}$	$\mathbf{11 : 1_{(3)}}$		$\mathbf{11 : 1_{(3)}}$
$\mathbf{0 : 10_{(1,3)}}$		$\mathbf{0 : 10_{(1,3)}}$				
	$\mathbf{: 101_{(1,2)}}$		$: 111_{(3)}$	$\mathbf{: 101_{(1,2)}}$		$: 111_{(3)}$
$0010_{(1,3)}$	$0101_{(1,2)}$	$0110_{(1,3)}$	$\mathbf{0111_{(2,3)}}$	$1101_{(1,2,3)}$	$\mathbf{1110_{(1,2)}}$	$1111_{(3)}$

У табл. 5 жирним шрифтом виділено елементи мінімального покриття, що мінімізують функцію Y методом сумісної мінімізації системи

$$Y = \overline{x}_1 x_3 \overline{x}_4 (1,3) + x_2 \overline{x}_3 x_4 (1,2) + \overline{x}_1 x_2 x_3 x_4 (2,3) + x_1 x_2 x_4 (3) + x_1 x_2 x_3 \overline{x}_4 (1,2). \qquad (10)$$

Результат мінімізації функції системних кон’юнктермів (10) комбінаторним методом збігається із результатом мінімізації методом паралельного розщеплення кон’юнктермів [12].

Оскільки для мінімізації функції Y комбінаторним методом у прикладі 6 операцію склеювання застосовано для тотожних кон’юнктермів, її не застосовували між тотожними та нетотожними кон’юнктермами, це зумовило зменшення кількості зайвих простих імплікант та розміру таблиці покриття (табл. 5).

Після розподілу системних кон’юнктермів функції (10) отримуємо мінімізовану систему булевих функцій

$$\begin{cases} f_1 = \overline{x}_1 x_3 \overline{x}_4 + x_2 \overline{x}_3 x_4 + x_1 x_2 x_3 \overline{x}_4 \\ f_2 = x_2 \overline{x}_3 x_4 + \overline{x}_1 x_2 x_3 x_4 + x_1 x_2 x_3 \overline{x}_4 \\ f_3 = \overline{x}_1 x_3 \overline{x}_4 + \overline{x}_1 x_2 x_3 x_4 + x_1 x_2 x_3 \end{cases}.$$

Приклад 7. Мінімізувати логічну функцію

$F(x_1, x_2, x_3, x_4)$ =(0,1,6,8,11,14,15), що задана у канонічній формі [13, с. 49], комбінаторним методом.

$$F = \left| \begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 6 & 0 & 1 & 1 & 0 \\ 8 & 1 & 0 & 0 & 0 \\ 11 & 1 & 0 & 1 & 1 \\ 14 & 1 & 1 & 1 & 0 \\ 15 & 1 & 1 & 1 & 1 \end{array} \right| = \left| \begin{array}{cccc} 0 & 0 & 0 & 1 \\ & 0 & 0 & 0 \\ & 1 & 1 & 0 \\ 1 & & 1 & 1 \end{array} \right| = \left| \begin{array}{cccc} 0 & 0 & 0 & \\ & 0 & 0 & 0 \\ & 1 & 1 & 0 \\ 1 & & 1 & 1 \end{array} \right|.$$

Результати мінімізації функції $F(x_1, x_2, x_3, x_4)$ за допомогою паралельного розщеплення кон’юнктермів [13, с. 49] та комбінаторним методом подано у табл. 6.

Таблиця 6

Результат мінімізації функції $F(x_1, x_2, x_3, x_4)$

Методом паралельного розщеплення кон’юнктермів	Комбінаторним методом
$\{(000\ :\ ), (\ :\ 000), (\ :\ 110), (1\ :\ 11)\}$	$\left| \begin{array}{cccc} 0 & 0 & 0 & \\ & 0 & 0 & 0 \\ & 1 & 1 & 0 \\ 1 & & 1 & 1 \end{array} \right|$

З табл. 6 можна бачити, що результати мінімізації двох порівнюваних методів однакові. Збігається й показник мінімізації $k_q / k_l = 4/12$, де k_q – кількість простих імплікант, k_l – кількість вхідних змінних. Однак обчислювальна складність мінімізації булевої функції комбінаторним методом менша.

Приклад 8. Мінімізувати логічну функцію

$F(x_1, x_2, x_3, x_4)$ =(1,1,1,1,1,1,0,0,1,1,1,1,0,0,0,0) [14] комбінаторним методом.

Складемо таблицю істинності заданої чотирирозрядної функції з блоків, за яких функція набуває значення одиниці, тобто для наборів: 0, 1, 2, 3, 4, 5, 8, 9, 10, 11, та здійснимо мінімізацію:

$$\left| \begin{array}{c|cccc} 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \\ 2 & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ 3 & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} \\ 4 & 0 & 1 & 0 & 0 \\ 5 & 0 & 1 & 0 & 1 \\ 8 & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 9 & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} \\ 10 & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ 11 & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} \end{array} \right| = \left| \begin{array}{cccc} & 0 & & \\ 0 & 1 & 0 & \end{array} \right| = \left| \begin{array}{cccc} & 0 & & \\ 0 & & 0 & \end{array} \right|.$$

До блоків 0–3 та 8–11 першої матриці застосовано правило суперсклеювання змінних, оскільки наявна комбінаторна система 2–(3, 8)-design (виділена жирним шрифтом). В останній матриці здійснено заміщення (неповне склеювання) змінних.

У підсумку отримуємо мінімальну функцію:

$$F = \bar{x}_1\bar{x}_3 + \bar{x}_2.$$

Результат мінімізації комбінаторним методом збігається з результатом мінімізації, отриманим за допомогою методу самопонижувальних циклів [14]. Метод самопонижувальних циклів для мінімізації заданої функції використовує чотири понижувальних цикли і таблицю покриття для видалення зайвих імплікант. Комбінаторний метод задану функцію мінімізує за три образні перетворення. Оскільки метод самопонижувальних циклів для мінімізації булевої функції використовує повну комбінаторну систему з повторенням 2–(n, b)-design [14], але не використовує неповну комбінаторну систему з повторенням 2–(n, x/b)-design, цей метод можна зарахувати до часткових випадків мінімізації комбінаторним методом.

Мінімізація п’ятирозрядних булевих функцій

Приклад 9. Мінімізувати логічну функцію

$F(x_1, x_2, x_3, x_4, x_5)$ =(0,1,3,4,7,13,15,19,20,22,23,29,31) [15] комбінаторним методом.

$$\left|\begin{array}{c|ccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 3 & 0 & 0 & 0 & 1 & 1 \\ 4 & 0 & 0 & 1 & 0 & 0 \\ 7 & 0 & 0 & 1 & 1 & 1 \\ 13 & 0 & 1 & 1 & 0 & 1 \\ 15 & 0 & 1 & 1 & 1 & 1 \\ 19 & 1 & 0 & 0 & 1 & 1 \\ 20 & 1 & 0 & 1 & 0 & 0 \\ 22 & 1 & 0 & 1 & 1 & 0 \\ 23 & 1 & 0 & 1 & 1 & 1 \\ 29 & 1 & 1 & 1 & 0 & 1 \\ 31 & 1 & 1 & 1 & 1 & 1 \end{array}\right| = \left|\begin{array}{ccccc} 0 & 0 & 0 & 0 & \\ 0 & 0 & & 1 & 1 \\ & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & & 1 \\ 1 & 0 & & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & & 1 \end{array}\right| = \left|\begin{array}{ccccc} 0 & 0 & 0 & 0 & \\ & 0 & & 1 & 1 \\ & 0 & 1 & 0 & 0 \\ & 1 & 1 & & 1 \\ 1 & 0 & 1 & 1 & 0 \end{array}\right| = \left|\begin{array}{ccccc} x_5 & x_4 & x_3 & x_2 & x_1 \\ 0 & 0 & 0 & 0 & \\ & 0 & & 1 & 1 \\ & 0 & 1 & 0 & 0 \\ & 1 & 1 & & 1 \\ 1 & 0 & 1 & 1 & \end{array}\right|. \qquad (11)$$

У першій та другій матрицях блок-схеми (11) виконано просте склеювання змінних, в останній матриці – неповне склеювання змінних.

У підсумку отримуємо мінімальну функцію:

$$F = \bar{x}_5\bar{x}_4\bar{x}_3\bar{x}_2 + \bar{x}_4x_3\bar{x}_2\bar{x}_1 + \bar{x}_4x_2x_1 + x_5\bar{x}_4x_3x_2 + x_4x_3x_1 \qquad (12)$$

У табл. 7 наведено результати мінімізації функції $F(x_1, x_2, x_3, x_4, x_5)$ за допомогою методу таблиці істинності з розподіленим спрощенням [15] та комбінаторним методом.

Таблиця 7

Результат мінімізації функції $F(x_1, x_2, x_3, x_4, x_5)$

Методом таблиці істинності з розподіленим спрощенням
$F = \bar{x}_5\bar{x}_4\bar{x}_3\bar{x}_2 + \bar{x}_5\bar{x}_4\bar{x}_2\bar{x}_1 + \bar{x}_4x_2x_1 + x_5\bar{x}_4x_3\bar{x}_1 + x_4x_3x_1$
Комбінаторним методом
$F = \bar{x}_5\bar{x}_4\bar{x}_3\bar{x}_2 + \bar{x}_4x_3\bar{x}_2\bar{x}_1 + \bar{x}_4x_2x_1 + x_5\bar{x}_4x_3x_2 + x_4x_3x_1$

Основну відмінність мінімальних функцій табл. 7 демонструють друга та четверта імпліканти. Функція, мінімізована методом “таблиці істинності з розподіленим спрощенням”, для підтримання своєї функціональності потребує на два інвертори більше. Отже, використовуючи, наприклад, технологію К-МОН (комплементарна структура метал–оксид–напівпровідник), апаратна реалізація функції (12) потребуватиме на два інвертори менше.

Мінімізована логічна функція (12) задовольняє задану таблицю істинності (перша матриця блок-схеми (11)).

Мінімізація шестирозрядних булевих функцій

Приклад 10. Мінімізувати логічну функцію

$F(x_1, x_2, x_3, x_4, x_5, x_6)$ =(0,1,0,1,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,1,1,0,0,1,1,0,0,1,1,1,1,1,1,1,1,0,0,1,1,0,0,1,1,1,1,1,1,1,1,1) [16] комбінаторним методом.

Складемо таблицю істинності, що задається шестирозрядною функцією з блоків, за яких функція отримує значення одиниці, тобто для наборів: 1, 3, 10, 11, 12, 13, 26, 27, 28, 29, 32, 33, 36, 37, 40, 41, 42, 43, 44, 45, 46, 47, 50, 51, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, та здійснимо мінімізацію заданої булевої функції.

0	0	0	0	0	1
0	0	0	0	1	1
0	0	1	0	1	0
0	0	1	0	1	1
0	0	1	1	0	0
0	0	1	1	0	1
0	1	1	0	1	0
0	1	1	0	1	1
0	1	1	1	0	0
0	1	1	1	0	1
1	0	0	0	0	0
1	0	0	0	0	1
1	0	0	1	0	0
1	0	0	1	0	1
1	0	1	**0**	**0**	**0**
1	0	1	**0**	**0**	**1**
1	0	1	**0**	**1**	**0**
1	0	1	**0**	**1**	**1**
1	0	1	**1**	**0**	**0**
1	0	1	**1**	**0**	**1**
1	0	1	**1**	**1**	**0**
1	0	1	**1**	**1**	**1**
1	1	0	0	1	0
1	1	0	0	1	1
1	1	0	1	1	0
1	1	0	1	1	1
1	1	1	**0**	**0**	**0**
1	1	1	**0**	**0**	**1**
1	1	1	**0**	**1**	**0**
1	1	1	**0**	**1**	**1**
1	1	1	**1**	**0**	**0**
1	1	1	**1**	**0**	**1**
1	1	1	**1**	**1**	**0**
1	1	1	**1**	**1**	**1**

$$
=\left|\begin{array}{cccccc}
0 & 0 & 0 & 0 & & 1 \\
0 & 0 & 1 & 0 & 1 & \\
0 & 0 & 1 & 1 & 0 & \\
0 & 1 & 1 & 0 & 1 & \\
0 & 1 & 1 & 1 & 0 & \\
1 & 0 & 0 & 0 & 0 & \\
1 & 0 & 0 & 1 & 0 & \\
\mathbf{1} & \mathbf{0} & \mathbf{1} & & & \\
1 & 1 & 0 & 0 & 1 & \\
1 & 1 & 0 & 1 & 1 & \\
\mathbf{1} & \mathbf{1} & \mathbf{1} & & &
\end{array}\right|
=\left|\begin{array}{cccccc}
0 & 0 & 0 & 0 & & 1 \\
0 & & 1 & 0 & 1 & \\
0 & & 1 & 1 & 0 & \\
1 & 0 & 0 & & 0 & \\
1 & & 1 & & & \\
1 & 1 & 0 & & 1 &
\end{array}\right|
=\left|\begin{array}{cccccc}
0 & 0 & 0 & 0 & & 1 \\
 & & 1 & 0 & 1 & \\
 & & 1 & 1 & 0 & \\
1 & 0 & & & 0 & \\
1 & & 1 & & & \\
1 & 1 & & & 1 &
\end{array}\right| \quad (13)
$$

Для блоків 15–22 та 27–34 першої матриці комбінаторної системи (13) використано правило суперсклеювання змінних, оскільки у цих блоках наявна повна комбінаторна система з повторенням 2–(3,8)-design (виділена жирним шрифтом). Для решти блоків першої матриці здійснено просте склеювання змінних (результат алгебричних перетворень записано у другу матрицю). У другій матриці системи (13) виконано просте склеювання змінних, а у третій – заміщення змінних.

У підсумку отримуємо мінімальну функцію:

$$F = \overline{x}_1\overline{x}_2\overline{x}_3\overline{x}_4 x_6 + x_3\overline{x}_4 x_5 + x_3 x_4\overline{x}_5 + x_1\overline{x}_2\overline{x}_5 + x_1 x_3 + x_1 x_2 x_5. \quad (14)$$

Результат мінімізації комбінаторним методом (14) збігається з результатом мінімізації, одержаним за допомогою тривимірної карти Карно [16]. Приклад 10 демонструє меншу обчислювальну складність мінімізації булевої функції комбінаторним методом.

0	**0**	**0**	**0**	0	1	0	1
0	**0**	**0**	**1**	0	1	0	1
0	**0**	**1**	**0**	0	1	0	1
0	**0**	**1**	**1**	0	1	0	1
0	**1**	**0**	**0**	0	1	0	1
0	**1**	**0**	**1**	0	1	0	1
0	**1**	**1**	**0**	0	1	0	1
0	**1**	**1**	**1**	0	1	0	1
1	**0**	**0**	**0**	0	0	0	0
1	**0**	**0**	**1**	0	0	0	0
1	**0**	**1**	**0**	0	0	0	0
1	**0**	**1**	**1**	0	0	0	0
1	**1**	**0**	**0**	0	0	0	0
1	**1**	**0**	**1**	0	0	0	0
1	**1**	**1**	**0**	0	0	0	0
1	**1**	**1**	**1**	0	0	0	0
1	**0**	**0**	**0**	0	0	0	1
1	**0**	**0**	**1**	0	0	0	1
1	**0**	**1**	**0**	0	0	0	1
1	**0**	**1**	**1**	0	0	0	1
1	**1**	**0**	**0**	0	0	0	1
1	**1**	**0**	**1**	0	0	0	1
1	**1**	**1**	**0**	0	0	0	1
1	**1**	**1**	**1**	0	0	0	1
1	**0**	**0**	**0**	0	0	1	0
1	**0**	**0**	**1**	0	0	1	0
1	**0**	**1**	**0**	0	0	1	0
1	**0**	**1**	**1**	0	0	1	0
1	**1**	**0**	**0**	0	0	1	0
1	**1**	**0**	**1**	0	0	1	0
1	**1**	**1**	**0**	0	0	1	0
1	**1**	**1**	**1**	0	0	1	0
1	**0**	**0**	**0**	0	1	1	1
1	**0**	**0**	**1**	0	1	1	1
1	**0**	**1**	**0**	0	1	1	1
1	**0**	**1**	**1**	0	1	1	1
1	**1**	**0**	**0**	0	1	1	1
1	**1**	**0**	**1**	0	1	1	1
1	**1**	**1**	**0**	0	1	1	1
1	**1**	**1**	**1**	0	1	1	1

1	**0**	**0**	**0**	1	1	1	1
1	**0**	**0**	**1**	1	1	1	1
1	**0**	**1**	**0**	1	1	1	1
1	**0**	**1**	**1**	1	1	1	1
1	**1**	**0**	**0**	1	1	1	1
1	**1**	**0**	**1**	1	1	1	1
1	**1**	**1**	**0**	1	1	1	1
1	**1**	**1**	**1**	1	1	1	1
1	**0**	**0**	**0**	1	0	1	0
1	**0**	**0**	**1**	1	0	1	0
1	**0**	**1**	**0**	1	0	1	0
1	**0**	**1**	**1**	1	0	1	0
1	**1**	**0**	**0**	1	0	1	0
1	**1**	**0**	**1**	1	0	1	0
1	**1**	**1**	**0**	1	0	1	0
1	**1**	**1**	**1**	1	0	1	0
1	**0**	**0**	**0**	1	0	0	1
1	**0**	**0**	**1**	1	0	0	1
1	**0**	**1**	**0**	1	0	0	1
1	**0**	**1**	**1**	1	0	0	1
1	**1**	**0**	**0**	1	0	0	1
1	**1**	**0**	**1**	1	0	0	1
1	**1**	**1**	**0**	1	0	0	1
1	**1**	**1**	**1**	1	0	0	1
1	**0**	**0**	**0**	1	0	0	0
1	**0**	**0**	**1**	1	0	0	0
1	**0**	**1**	**0**	1	0	0	0
1	**0**	**1**	**1**	1	0	0	0
1	**1**	**0**	**0**	1	0	0	0
1	**1**	**0**	**1**	1	0	0	0
1	**1**	**1**	**0**	1	0	0	0
1	**1**	**1**	**1**	1	0	0	0

Приклад 11. Мінімізувати комбінаторним методом логічну функцію $F(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8)$ [17], яка задана поданою нижче таблицею істинності.

$$
= \left| \begin{array}{ccccc}
0 & 0 & 1 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 & 0 \\
1 & 0 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 \\
1 & 1 & 0 & 1 & 0 \\
1 & 1 & 0 & 0 & 1 \\
1 & 1 & 0 & 0 & 0
\end{array} \right| =
$$

$$
= \left| \begin{array}{ccccc}
0 & 0 & 1 & 0 & 1 \\
1 & & 0 & 0 & \\
1 & & 0 & 1 & 0 \\
1 & & 1 & 1 & 1
\end{array} \right| =
$$

$$
= \left| \begin{array}{ccccc}
0 & 0 & 1 & 0 & 1 \\
1 & & 0 & 0 & \\
1 & & 0 & & 0 \\
1 & & 1 & 1 & 1
\end{array} \right| . \qquad (15)
$$

Для кожних восьми блоків першої матриці системи (15) використано правило суперсклеювання змінних, оскільки у кожній “вісьмірці” блоків наявна повна комбінаторна система з повторенням 2–(3,8)-design (ці системи виділено).

У другій матриці блок-схеми (14) застосовано правило суперсклеювання (присутня 2-(2,4)-design, блоки матриці виділено) та правило (7) (блоки матриці виділено). Результат склеювання записано у третю матрицю. У третій матриці здійснено неповне склеювання змінних із записом результату у четверту матрицю.

У підсумку отримуємо мінімальну функцію (16):

$$
F = \overline{x_1 x_5} x_6 \overline{x_7} x_8 + x_1 \overline{x_6 x_7} + x_1 \overline{x_6 x_8} + x_1 x_6 x_7 x_8 . \qquad (16)
$$

У табл. 8 подано результати мінімізації функції $F(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8)$ картою Карно [17] та комбінаторним методом.

Таблиця 8

Результат мінімізації функції $F(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8)$

Мінімізація картою Карно
$F(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8) = x_1\overline{x_6}\,\overline{x_7} + x_1\overline{x_6}x_7\overline{x_8} + x_1x_6x_7x_8 + \overline{x_1}\,\overline{x_5}x_6\overline{x_7}x_8$
Мінімізація комбінаторним методом
$F(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8) = x_1\overline{x_6}\,\overline{x_7} + x_1\overline{x_6}\,\overline{x_8} + x_1x_6x_7x_8 + \overline{x_1}\,\overline{x_5}x_6\overline{x_7}x_8$

З табл. 8 бачимо, що комбінаторний метод дає другу імпліканту мінімальної булевої функції з меншою кількістю вхідних змінних.

Результати дослідження

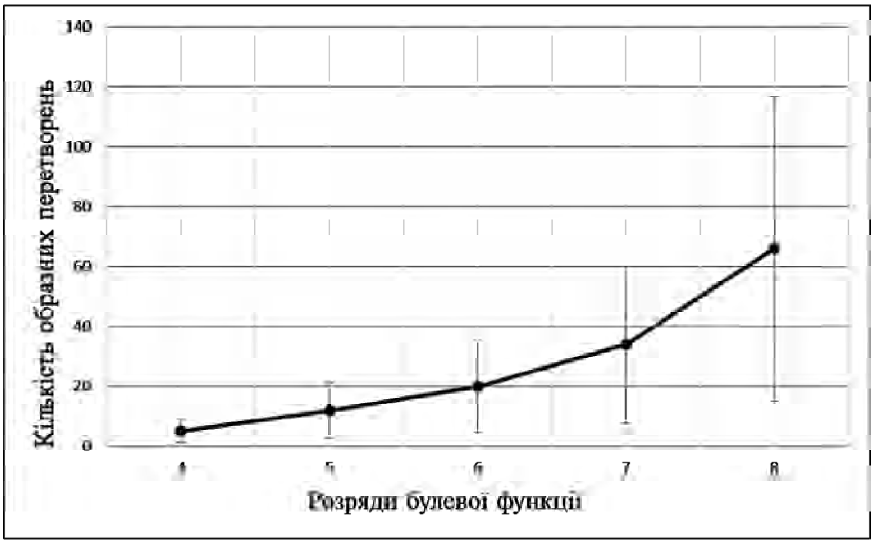
Складність алгоритму мінімізації булевої функції – це кількісна характеристика, що відображає споживані алгоритмом ресурси під час його виконання. Основними ресурсами, якими оцінюється складність алгоритму, є час обчислення і простір пам'яті, необхідний для здійснення обчислення за цим алгоритмом. Для оцінювання алгоритмічної складності мінімізації булевої функції комбінаторним методом як споживані алгоритмічні ресурси беруть операції образних перетворень, які здійснюються під час пошуку мінімальної функції. Одну операцію суперсклеювання, простого склеювання, узагальненого склеювання, поглинання та заміщення змінних приймають як одне перетворення. Кількість зазначених перетворень залежить від розрядності булевої функції, кількості вихідних кон'юнктермів функції та структури таблиці істинності. Можливу кількість таких перетворень залежно від розрядності булевої функції подано у табл. 9.

Таблиця 9

Витрачені образні перетворення комбінаторного методу

Розрядність функції	Кількість образних перетворень комбінаторного методу
4	4–5
5	7–18
6	8–32
7	10–58
8	15–117

На рисунку відображено динаміку зростання кількості образних перетворень комбінаторним методом мінімізації зі збільшенням розрядності булевої функції.



Динаміка зростання кількості образних перетворень комбінаторного методу мінімізації зі збільшенням розрядності булевої функції

За даними, що є у нашому розпорядженні, можна у першому наближенні вважати складність алгоритму за комбінаторним методом лінійно залежною від кількості образних перетворень з оцінкою складності – О (n) для $n < 7$. Зі збільшенням кількості змінних від n=6 до 8 динаміка зростання кількості перетворень характеризується законом О (n^2) з подальшим зростанням О (f (n)) зі збільшенням розрядності булевої функції за поліноміальним законом.

Висновки

1. Упровадження алгебричної операції суперсклеювання змінних дає змогу спростити процедуру мінімізації булевої функції без втрати її функціональності

2. Алгебрична операція суперсклеювання змінних здійснюється за наявності у структурі таблиці істинності повної бінарної комбінаторної системи з повторенням або неповної бінарної комбінаторної системи з повторенням. Операція суперсклеювання змінних найефективніша за наявності повної бінарної комбінаторної системи з повторенням. Ефективність операції суперсклеювання змінних за наявності неповної бінарної комбінаторної системи з повторенням зменшується несуттєво.

3. Встановлено, що результати верифікації мінімізованої функції, отриманої з використанням правила суперсклеювання змінних, задовольняють вихідний протокол обчислення заданої функції і, отже, засвідчують оптимальне зменшення кількості змінних функції без втрати її функціональності. Оцінка складності алгоритму пошуку мінімальної функції комбінаторним методом становить О (n) і є лінійною для $n < 7$. Зі збільшенням кількості змінних від n=6 до 8 динаміка зростання кількості перетворень характеризується законом О (n^2) з подальшим зростанням О (f (n)) зі збільшенням розрядності булевої функції за поліноміальним законом.

4. Ефективність комбінаторного методу демонструється прикладами мінімізації функцій, запозиченими з робіт інших авторів з метою порівняння:

– *приклад* 4 [11], *приклад* 5 [12], *приклад* 7 [13], *приклад* 8 [14] – мінімізація 4-розрядних булевих функцій;

– *приклад* 6 [12] – мінімізація системи 4-розрядних булевих функцій;

– *приклад* 9 [15] – мінімізація 5-розрядних булевих функцій;

– *приклад* 10 [16] – мінімізація 6-розрядних булевих функцій;

– *приклад* 11 [17] – мінімізація 8-розрядних булевих функцій.

З огляду на зазначені приклади комбінаторний метод мінімізації функції дає підставу для висновку про доцільність застосування його у процесах мінімізації логічної функції.

*1. Quine–McCluskey algorithm [Electronic resource]. – Access mode: https://en.wikipedia.org/wiki/Quine%E2 %80 %93McCluskey_algorithm – 15.10.2017. – Title from the screen. 2. Riznyk V. Minimization of boolean functions by combinatorial method [Text] / V. Riznyk, M. Solomko // Technology audit and production reserves. – Vol 4/2 (36), 2017. – P. 49–64. http://journals.uran.ua/tarp/article/view/108532 . 3. Riznyk V. Application of super-sticking algebraic operation of variables for boolean functions minimization by combinatorial method [Text] / V. Riznyk, M. Solomko // Technology audit and production reserves. – Vol. 6 (38), 2017. P. 50 – 66. 4. Manojlović, Vladislav (2013) Minimization of Switching Functions using Quine-McCluskey Method. International Journal of Computer Applications (0975 – 8887) Volume 82 – No 4, November 2013, 12–16. http://research.ijcaonline.org/volume82/number4/pxc3892127.pdf 5. Rytsar, Bohdan (2015) The Minimization Method of Boolean Functionns in Polynomial Set-theoretical Format. Conference: Proc. 24th Inter. Workshop, CS@P'2015, Sept. 28-30, 2015, 130–146 pp. (17), At Rzeszow, Poland, Volume: vol.2 http://dspace.nbuv.gov.ua/handle/123456789/87194 6. Rathore, T. S. (2014) Minimal Realizations of Logic Functions Using Truth Table Method with Distributed Simplification. IETE JOURNAL OF EDUCATION, Vol. 55, No. 1, JAN_JUN 2014, 26–32 http://www.tandfonline.com/doi/pdf/ 10.1080/09747338.2014.921412 7. Dan, Rotar (2010) Software for The Minimization of The Combinational Logic Functions. The Romanian Review Precision Mechanics, Optics & Mechatronics, 2010 (20), No. 37, 95–99. https://www.researchgate.net/publication/ 268270733_Software_for_*

The_Minimization_of_The_Combinational_Logic_Functions_SOFTWARE_FOR_THE_MINIMIZATION_OF_THE_COMBINATIONAL_LOGIC_FUNCTIONS 8. Zolfaghari, Behrouz, Sheidaeian, Hamed (2011) A NEW CASE FOR IMAGE COMPRESSION USING LOGIC FUNCTION MINIMIZATION. The International Journal of Multimedia & Its Applications (IJMA) Vol.3, No. 2, May 2011, 45–62. http://aircconline.com/ijma/V3N2/3211ijma04.pdf 9. Nosrati M., Karimi R., Nariri M. (2012) MINIMIZATION OF BOOLEAN FUNCTIONS USING GENETIC ALGORITHM. Anale. Seria Informatica. Vol. X fasc. 1 – 2012, 73–77. https://pdfs.semanticscholar.org/c53d/2240a2aa5531832a7707ad186dee23129ed8.pdf 10. Nosrati M., Karimi R. (2011) An Algorithm for Minimizing of Boolean Functions Based on Graph DS. World Applied Programming, Vol (1), No. (3), August 2011. 209–214. http://waprogramming.com/papers/50ae59d04ee143.95681909.pdf 11. Bunyak, A. Electronics and chip technology [Electronic resource] / A. Bunyak – Kyiv: View. "Aston", 2001. – 385 p. – Access mode: http://radio-best.cf/informatsiya/67-bunyak-a-elektronika-ta-mikroskhemotekhnika-2001-djvu-otsifrovano-gurtom – Title from the screen. 12. Rytsar, B. Ye. Minimization of the system of logical functions by the method of parallel decoupling of conjunctures [Text] / B. Ye. Rytsar // Bulletin of the "Lviv Polytechnic" National University. Radio Electronics and Telecommunications. – 2013. – No. 766. – P. 18–27. – Access mode: http://nbuv.gov.ua/UJRN/VNULPPT_2013_766_6 . 13. Rytsar, B. Ye. New minimization method of logical functions in polynomial set-theoretical format. 1. Generalized rules of conjuncterms simplification / B. Ye. Rytsar // Control systems and machines. 2015. № 2. P. 39–57. – Available at: \www/URL: http://dspace.nbuv.gov.ua/handle/123456789/87194. 14. Samofalov, K. G. Applied theory of digital automata [Electronic resource] / K. G. Samofalov, A. M. Romlinkevich, V. N. Valuisky, Yu. S. Kanevsky, M. M. Pinevich – K.: Vishcha shk. Head Publishing House, 1987. – 375 p. – Access mode: http://stu.scask.ru/book_pta.php?id=62 – 15. 12. 2017. – Title from the screen. 15. Rathore T. S. (2014) Minimal Realizations of Logic Functions Using Truth Table Method with Distributed Simplification / T. S. Rathore // IETE Journal of Education, 55:1, 26–32, DOI: 10.1080/09747338.2014.921412.http://www.tandfonline.com/doi/pdf/10.1080/09747338.2014.921412?needAccess=true 16. The three-dimensional map of Carnot [Electronic resource] – Access mode: http://cyclowiki.org/wiki/%D0 %A2 %D1 %80 %D1 %91 %D1 %85 %D0 %BC%D0 %B5 %D1 %80 %D0 %BD%D0 %B0 %D1 %8F_%D0 %BA%D0 %B0 %D1 %80 %D1 %82 %D0 %B0_%D0 %9A%D0 %B0 %D1 %80 %D0 %BD%D0 %BE – 15. 10. 2017. – Title from the screen. 17. Map of Carnot [Electronic resource] – Access mode: https://ru.wikipedia.org/w/index.php?oldid=36798414 – 15. 10. 2017. – Title from the screen.