

ДЕКОМПОЗИЦІЯ СКЛАДНИХ СХЕМ НА ОСНОВІ ІЄРАРХІЧНОЇ КЛАСТЕРИЗАЦІЇ

Р.П.Базилевич, І.В.Подольський, П.Р.Базилевич

НУ "Львівська політехніка", вул. С.Бандери 12, Львів, 79013,

тел. (0322) 398578, ел. пошта: rbaz@polynet.lviv.ua

Анотація. Пропонується ряд алгоритмів декомпозиції складних схем, придатних для задач високих та надвисоких розмірностей. Методологічною основою є метод оптимального згортання схеми, який виявляє ієрархічно вкладені згустки схеми – її кластери. Деталізовано декілька алгоритмів початкового розбиття схеми на частини та його оптимізації. Замість оперування базовими елементами схеми, число яких може бути надзвичайно великим, використовуються кластери довільної розмірності. Це покращує якість розв'язку та зменшує обчислювальні затрати, підвищуючи швидкодію алгоритмів. Приводяться результати тестувань, які підтверджують ефективність запропонованої методики.

1. ВСТУП

Однією з важливих задач при розробці алгоритмів розпізнавання образів є розбиття складної схеми на частини з заданими обмеженнями. З математичної точки зору – це важковирішувана комбінаторна задача неполіноміальної складності (NP важка). У випадку її великих та надвеликих розмірностей (сотні тисяч та мільони складових елементів) для її ефективного розв'язання можливим є використання тільки наближених методів. Доцільним для цієї мети виявився метод оптимального згортання схеми [1,2], що дозволяє виявити ієрархічно вкладені кластери схеми. Кожна реальна електрична схема є неоднорідною, має певні згустки (кластери), в яких складові елементи зв'язані більш щільно, ніж у решті її частинах. При розв'язуванні різноманітних задач їх можна розглядати як макроелементи та описувати макромоделями. Це суттєво зменшує обчислювальну складність та сприяє покращенню якості розв'язку.

2. ФОРМУЛЮВАННЯ ЗАДАЧІ

Під розбиттям схеми, утвореної множиною $P = \{p_1, \dots, p_n\}$ складових елементів, на частини вважаємо систему $\tilde{P} = \{P_1, \dots, P_k\}$, для якої

$$(\forall P_i \in \tilde{P}) [(P_i \subset P) \ \& \ (P_i \neq \emptyset)];$$

$$(\forall (P_i, P_j) \in \tilde{P}) [P_i \cap P_j = \emptyset]; \bigcup_{i=1}^k \bigcup_{j=1}^{n_i} p_{ij} = P.$$

Тут $P_i = \{p_{i1}, \dots, p_{in_i}\}$ – множина всіх елементів частини P_i . Задача полягає у знаходженні найкращого з можливих розбиттів, що задовольняє

бажані обмеження. Вона може бути сформульована як знаходження такого розбиття \tilde{P}^* , при якому досягається екстремум значення критерію Q у допустимій області D можливих розв'язків:

$$Q(\tilde{P}^*) = \text{extr } Q(\tilde{P}_i); \tilde{P}_i \in D.$$

Як критерії та обмеження при розбитті можуть виступати: функційна завершеність утворених підсхем; еквівалентність виділених частин підсхемам з набору типових рішень, заданих бібліотекою (задача покриття); взаємна еквівалентність виділених підсхем з мінімізацією числа їх типів (задача типізації); число утворених підсхем; зовнішня зв'язність підсхем – сумарне число зв'язків між утвореними частинами; внутрішня зв'язність підсхем – сумарне число зв'язків всередині підсхем та їх середнє значення на одну підсхему; окремі характеристики кожної підсхеми: число її складових елементів, число зовнішніх зв'язків та деякі інші величини (сумарна площа всіх елементів, сумарний об'єм, сумарне тепло, що виділяється; множини елементів, які повинні бути призначені в різні підсхеми, або навпаки – в одну підсхему; прив'язка за характеристиками зв'язності до певних зовнішніх об'єктів, наприклад роз'ємів).

Основна ідея розвинутого підходу полягає в тому, що спочатку будується дерево оптимального згортання T^R за висхідною стратегією, яке виділяє кластери схеми, після чого за низхідною та комбінованою стратегіями на основі попередньо отриманої інформації здійснюється розбиття схеми на частини. Замість аналізу схеми на рівні початкових елементів здійснюється комбінаторний аналіз ієрархічно вкладених кластерів. Базуючись на розвинутому підході, задача розв'язується за три етапи: побудова дерева оптимального згортання; початкове розбиття; оптимізація розбиття.

3. АЛГОРИТМИ

ПОЧАТКОВОГО РОЗБИТТЯ

Використовуючи дерево оптимального згортання можна реалізувати декілька алгоритмів отримання початкового розв'язку для задачі розбиття схеми на частини: послідовного та паралельно-послідовного виділення частин; дихотомічного низхідного ділення; ділення схеми по її мінімальних розрізах та комбіновані.

Алгоритми послідовного виділення частин розбиття є найпростішими для реалізації, проте вони мають суттєвий недолік – в процесі вилучення

елементів та формування частин розбиття утворюються "хвости" – частини схеми, які слабо пов'язані між собою, та з яких формуються останні частини розв'язку. Перші виділені частини будуть доброї якості, а останні – погані.

В алгоритмах паралельно-последовного виділення частин розбиття дерево згортання формується повністю. Виділяються всі підхеми, число елементів яких дорівнює бажаному значенню або перевищило це значення вперше в процесі нарощення дерева згортання. З них процедурою вилучення зайвих елементів формуємо частини розбиття. Для решти елементів будуюмо нове дерево та виділення частин рекурсивно продовжуємо. У порівнянні з попереднім типом вказаний підхід є більш швидкодіючим, проте він може давати гірші результати, оскільки "хвости" формуються паралельно без врахування решти частини схеми. Для одного дерева згортання утворюється одночасно декілька "хвостів". У попередньому випадку після формування кожної окремої частини дерево згортання будувалось наново.

Ефективним підходом для усунення "хвостів" може бути "вимушене" призначення елементів у частини, яке одночасно враховує потреби рівномірного розподілу елементів серед усіх частин розбиття з врахуванням характеристик їх зв'язності. Такий підхід реалізується в алгоритмах дихотомічного низхідного ділення. Виділяються дві найвищі вершини дерева згортання та визначається число частин розбиття, що може бути утворене з

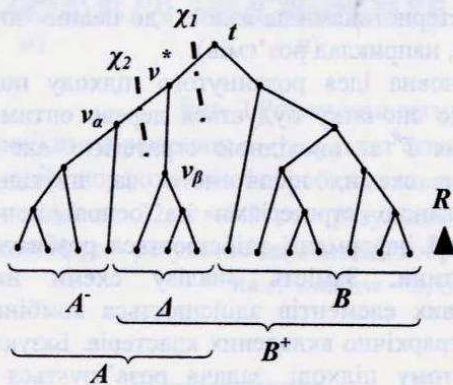


Рис. 1. Початкове виділення частин

кожної з них. Якщо числа їх елементів є кратними до бажаного значення елементів однієї частини (або близькими з врахуванням заданих обмежень на їх значення), то задача розбивається на дві ідентичні задачі менших розмірів, які продовжуємо дробити аналогічним чином. У більшості випадків ця умова не виконується. Тоді визначаємо, яке число елементів необхідно перенести з однієї частини в іншу, щоб задовольнити це значення. Виділяємо ту частину, з якої необхідно вилучити менше число елементів. Нехай це буде частина A , що відповідає вершині v^* (Рис. 1). Розглянемо процедуру

перенесення елементів з однієї частини в іншу. Тут можливими є два варіанти – спрощеного або розширеного аналізу. При спрощеному аналізі порівнюємо за числом елементів обидві складові підхеми вершини v^* : v_α та v_β , виділяємо меншу з них (нехай це буде v_β) та визначаємо серед її елементів бажані для перенесення. Можливими є три випадки. У першому з них число її елементів відповідає бажаному для перенесення значенню. Тоді приєднуємо цю підхему до частини B , утворюючи дві нові частини: $A' (A - \Delta)$ та $B' (B + \Delta)$ з бажаним для наступного ділення числом елементів. У другому випадку це число може виявитися меншим за бажане значення. Тоді можливими є два варіанти. Для першого з них, який забезпечує більшу точність, приєднуємо елементи Δ до частини B , моделюємо частину $(B + \Delta)$ однією вершиною, а з елементів частини $(A - \Delta)$ будуюмо нове дерево та продовжуємо пошук необхідних для перенесення елементів аналогічним чином. Для другого, більш простого, йдемо вниз по дереву, розглядаючи наступні дві складові більшої підхеми v_α і т.д. аж поки не отримаємо бажане для перенесення значення число елементів або перше більше за це значення. У третьому випадку це число є більше бажаного значення. Тоді задача зводиться до розподілу елементів множини Δ між частинами A' та B' таким чином, щоб задовольнити бажані умови на число елементів, що призначаються в обидві частини.

При розширеному аналізі пошук елементів для перенесення здійснюємо серед всіх елементів частини A . Тут можуть бути використані два

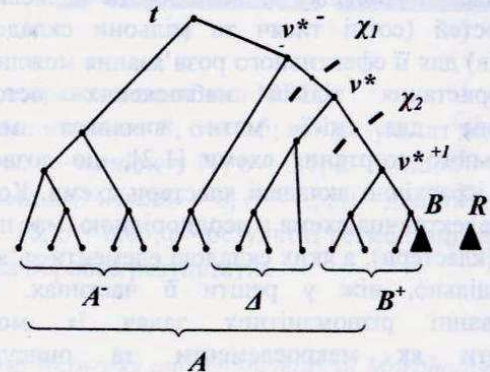


Рис. 2. Перерозподіл елементів частини A

підходи: на основі дерева згортання та на основі визначення ефективності перенесення окремих елементів та кластерів. Розглянемо спочатку перший підхід. Суть його полягає в побудові дерева згортання на основі всіх елементів частини A з розглядом частини B як єдиного цілого (Рис. 2). Решта частини схеми R також розглядається як єдине ціле, проте в процесі побудови дерева згортання до неї не дозволяється приєднувати елементи. Аналізуємо шлях від вершини B до вершини дерева t . Визначаємо на цьому шляху першу таку вершину v^* , що включення всіх елементів (включно з нею) на шляху від неї до B утворює вперше множину з

числом елементів, рівним або більшим від бажаного для перенесення. Нехай перетин χ_1 (між вершинами v^{*1} та v^*) виділяє таку вершину. Якщо це число відповідає бажаному для перенесення значенню, то розрізаємо схему за цим перерізом, утворюючи нові частини. Якщо воно є більшим від бажаного, то утворюємо нові частини A^- та B^+ , а елементи множини Δ розсіпаємо. До частини A^- приєднуємо всі елементи, що знаходяться на рисунку лівіше перерізу χ_1 , а до частини B^+ - всі елементи від перерізу χ_2 до вершини B включно з останньою.

Може трапитися частковий випадок, коли на виділеному вище шляху від вершини B до t немає належного числа елементів, або такі елементи взагалі є відсутніми. У першому разі до частини B приєднуємо всі ці елементи та задачу зводимо до нової аналогічної задачі зі збільшеним числом елементів частини B та зменшеним числом елементів частини A і продовжуємо задачу ітеративно розв'язувати аналогічним чином. Виродженням може трапитися випадок, коли на цьому шляху взагалі вершин немає. Рациональним тут слід вважати застосування підходу на основі визначення ефективності перенесення окремих елементів та кластерів.

Перейдемо до розгляду процедури розподілу елементів множини Δ між частинами A^- та B^+ (або B^-). Частини A^- та B^+ та решта частина схеми R розглядаються як окремі елементи, а елементи множини Δ розсіпаються. Будується дерево

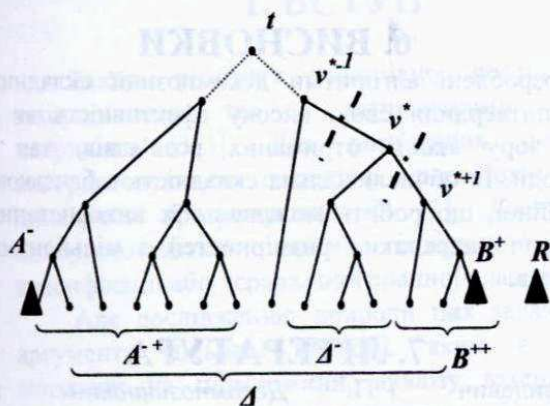


Рис. 3. Розподіл елементів множини Δ

згорання (Рис. 3). У процесі його побудови не дозволяється об'єднуватися піддеревом, що включають елементи A^- та B^+ . До елемента R також не дозволяється приєднувати елементи, проте зв'язки з цим елементом беруться до уваги. Після завершення згорання вершин утворюються два піддеревом, які умовно об'єднуємо вершиною t . На шляху від B^+ до A^- знаходимо вершину v^* таку, що включення її елементів та решти елементів цієї частини шляху буде відповідати бажаному значенню або буде першим значення, більшим за задане число. Якщо це число відповідає бажаному значенню, то задачу вважаємо розв'язаною. В протилежному

випадку всі елементи до цієї вершини зліва додаємо до частини A^- , а всі елементи справа - до частини B^+ , а елементи самої вершини v^* розсіпаємо на множини Δ та продовжуємо задачу ітеративно розв'язувати аналогічним чином. Як вироджений може трапитися випадок, коли всі елементи між вершинами A^- та B^+ об'єднуються в процесі згорання в одну вершину v^* , тобто не виникає умов для їх перерозподілення. Тоді необхідно застосувати підхід на основі визначення ефективності призначення окремих елементів та кластерів в кожному з підсхем зокрема.

4. АЛГОРИТМИ ОПТИМІЗАЦІЇ РОЗБИТТЯ

Доцільно виділити дві основні стратегії оптимізації розбиття: попарну та групову. Перша реалізується між окремими сформованими на попередньому етапі частинами розбиття. Групова оптимізація дозволяє переформовувати одночасно декілька груп. Початковими умовами для оптимізації є деяке розбиття $\tilde{P} = \{P_1, \dots, P_k\}$, яке оцінюється за певним значенням критерію якості $Q(\tilde{P})$. Метою оптимізації є отримання розбиття $\tilde{P}^* = \{P_1^*, \dots, P_k^*\}$ з кращим значенням критерію якості: $Q(\tilde{P}^*) > Q(\tilde{P})$. Запропоновані алгоритми реалізують покроковий процес покращання розв'язку.

В алгоритмах попарної оптимізації заданими є частини P_i та P_j , які описуються критерієм якості $Q(P_i, P_j)$. Необхідно отримати оптимізовані частини P_i^* та P_j^* , для яких $Q(P_i^*, P_j^*) > Q(P_i, P_j)$. Мова йде про перерозподіл елементів між частинами, який може покращити значення критерію. Можливими є два шляхи: для випадку розбиття на декілька частин (більше двох) поновне розділення множин, утворених елементами всіх пар P_i та P_j , або ітераційна оптимізація кожної пари (P_i, P_j) . Перший шлях передбачає об'єднання елементів обох частин в одну множину з повторним розв'язуванням задачі розбиття розглянутими вже методами початкового розбиття. Другий шлях передбачає ітераційну оптимізацію кожної пари (P_i, P_j) . Можливими є декілька стратегій: поодинокого переносу кластерів та елементів з коригуванням показників ефективності після кожної такої процедури, що забезпечуватиме найкращу якість, проте вимагає суттєвих обчислювальних затрат, або групового одночасного перенесення всіх елементів та кластерів з найкращим значенням критерію, а також парного чи групового обміну множинами кластерів та елементів. Для виходу з локальних екстремумів досліджено декілька типів різних збурень отриманого розв'язку з наступним продовженням оптимізаційного процесу. Збурення реалізуються перенесенням одного або декількох кластерів з однієї частини в іншу з погіршенням значення критерію, а також вилученням кластерів з найкращим значенням критерію для перенесення в обох частинах та

наступним перерозподілом їх складових елементів. Для цього можуть бути використані вже описані алгоритми початкового розподілу, які реалізуються тільки на виділеній частині елементів з початковими умовами, що визначаються залишеними частинами, або їх випадковий перерозподіл з наступною оптимізацією.

5. ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ

Ефективність запропонованих алгоритмів підтверджена рядом експериментів, виконаних на відомих тестах фірми IBM [3]. Графік часової залежності для базової процедури – побудови дерева

згортання, при якому мінімізується кількість зовнішніх зв'язків кожного кластера. Для виходу з локальних екстремумів застосовувались збурення кількох типів: перенесення одного, двох або трьох кластерів одночасно, які мають оптимальне відношення оцінки ефективності перенесення до кількості елементів або найкраще значення оцінки ефективності перенесення, комбіновані збурення та зрізання заданого відсотку елементів “шапки” (кластери, що мають найкраще значення оцінки перенесення) в обох частинах з подальшим випадковим призначенням цих елементів частинам розбиття. Часові характеристики наведено для процесора Intel Pentium III 800 MHz.

Таблиця 1. Результати оптимізації

Стратегія	Зріз “шапки”, %	Кількість кластерів збурення	Тип збурення	Кількість збурень	Макс. значення критерію	Мін. значення критерію	Оптимальні	≤1%	≤2%	≤5%	≤10%	Середнє арифметичне	Середня кількість ітерацій	Середній час, с.
1	0	1	оптимальний кластер	5	699	180	2	16	60	60	60	227	62	274
2	0	1	найкращий кластер	5	707	180	7	28	40	44	44	299	39	194
3	0	1	чергування 2 опт./ 2 найкр.	5	699	180	3	24	58	59	59	230	53	244
4	0	1	чергування 2 найкр./ 2 опт.	5	699	180	6	35	52	52	52	241	49	292
5	0	1+1	одночасно опт. + найкр.	5	699	180	2	16	61	61	61	227	62	252
6	0	2+2	одночасно опт. + найкр.	5	699	180	4	17	62	62	62	226	62	214
7	10	0	зрізання “шапки”	2	707	180	7	32	47	49	49	290	38	204
8	10	0	зрізання “шапки”	5	707	180	20	43	49	49	49	288	60	281
9	10	0	зрізання “шапки”	10	707	180	29	48	50	50	50	284	96	407
10	2	0	зрізання “шапки”	5	707	180	12	37	46	46	46	293	54	272

згортання є близьким до лінійного [4].

На пачці зі ста випадково згенерованих початкових розбиттів досліджено вплив ряду параметрів алгоритмів оптимізації, таких як тип критерію згортання, кількість та тип збурень, відсоток згортання дерева, відсоток зрізання “шапки” для збурення, на якість отримуваних розв'язків та кількість попадань в певний окіл оптимуму. Стратегії з високим відсотком попадань розв'язків у заданий окіл оптимуму доцільно застосовувати для подальшої оптимізації розв'язків у задачах розбиття.

У Таблиці 1 наведено результати досліджень для алгоритму оптимізації, що виявився при попередніх тестуваннях швидкодіючим та ефективнішим, у порівнянні з іншими розглянутими алгоритмами. Обраний алгоритм оптимізації передбачає по чергове одностороннє перенесення з однієї частини в іншу кластерів/елементів з від'ємним та нульовим значенням оцінки ефективності перенесення q_i в кількості, що не перевищує заданого відхилення розміру схеми від точного розбиття на два. Завершується робота алгоритму при умові повторення значення критерію якості $Q(\tilde{P}^*)$.

Результати подано для відсотку згортання дерева на кожній ітерації у 20% від діапазону значень критерію, допустимому відхиленні розбиття схеми від точного розбиття на два - 10% та критерію

6. ВИСНОВКИ

Розроблені алгоритми декомпозиції складних схем підтвердили свою високу ефективність як з точки зору якості отриманих розв'язків, так і швидкодії. Їх обчислювальна складність є близькою до лінійної, що робить можливим їх використання для задач надвеликих розмірностей з мільйонами елементів.

7. ЛІТЕРАТУРА

1. Базилевич Р.П. *Декомпозиционные и топологические методы автоматизированного конструирования электронных устройств*. Львів: Вища школа, 1981. –168 с.
2. Bazylevych R. *The Optimal Circuit Reduction Method as an Effective Tool to Solve Large and Very Large Size Intractable Combinatorial VLSI Physical Design Problems* // Proc. of 10th NASA Symposium on VLSI Design. - Albuquerque. - 2002. – P. 6.1.1-6.1.14.
3. Alpert C.J. *The ISPD-98 Circuit Benchmark Suit* // Proc. ACM/IEEE Intern. Symposium on Physical Design.- April 1998. – P. 80-85.
4. Базилевич Р.П., Подольський І.В. *Ієрархічна кластеризація складних схем*. Вісник Державного університету “Львівська політехніка”, №. 392, 2000, с. 155-158