

# КОНСТРУКТИВНІ АЛГОРИТМИ РОЗБИТТЯ СКЛАДНИХ СИСТЕМ НА ОСНОВІ ОПТИМАЛЬНОГО ЗГОРТАННЯ

Роман Базилевич

ДУ “Львівська політехніка”; вул.С.Бандери, 12; Львів, 79646; Україна  
Тел: +380322 398877; Факс: +380322 744300; E-mail:rbaz@polynet.lviv.ua

**Абстракт.** Розглядається задача розбиття складних систем на частини, яка з математичної точки зору відноситься до неpolіноміальних (важкорозв’язних) комбінаторних проблем. Запропоновано нові конструктивні алгоритми на основі методу оптимального згортання схеми. Виділено основні макропроцедури для кількох різних стратегій. Дано оцінку їх ефективності.

## 1. Вступ

Постійне зростання складності систем опрацювання даних не тільки не знімає, а підвищує актуальність однієї з найскладніших комбінаторних задач аналізу даних – оптимального розбиття системи на частини за заданими критеріями та з забезпеченням необхідних обмежень. Звичайно знайти оптимум для задач даного класу у випадку їх великої розмірності, не говорячи вже про реальні задачі надвеликих розмірностей, надзвичайно важко. Тому з практичної точки зору задовольняються “добрими” розв’язками, наближеними до оптимальних. З математичної точки зору проблема відноситься до неpolіноміальних задач дискретної оптимізації, які отримали назву важкорозв’язних (клас  $NP$ ). Існує багато різноманітних методів та алгоритмів розбиття [1]. Ефективним виявився метод оптимального згортання схеми (ОЗС) [2]. Важлива ідея підходу полягає в тому, щоб в операторах алгоритмів замість базових елементів системи як основних “цеглинок” для формування розв’язку використовуються кластери довільної розмірності з заданою структурою їх взаємного ієрархічного входження. Задача має різноманітні прикладні застосування, зокрема для аналізу даних, ієрархічної класифікації об’єктів, автоматизованого проектування та інших прикладних проблем.

## 2. Формулювання задачі

Заданою є система  $S=(P, E)$ , де  $P=\{p_1, \dots, p_n\}$  – множина базових елементів та  $E=\{e_1, \dots, e_m\}$  – множина зв’язків. Необхідно отримати розбиття  $\tilde{P}^*=\{P_1^*, \dots, P_k^*\}$  множини  $P$  на  $k$  підмножин, яке за

вибраним критерієм якості наближається до оптимального:  $Q(\tilde{P}^*) \rightarrow \text{opt}_t Q(\tilde{P}^i)$ , де  $\tilde{P}^i$  –

довільне розбиття. Як критерій якості найчастіше розглядається число зовнішніх зв’язків  $Q=|E^{ex}|$ , що зв’язують між собою частини розбитої системи.

## 3. Класифікація алгоритмів

Суть конструктивних алгоритмів розбиття полягає у формуванні розв’язку шляхом покрокового визначення його окремих елементів. Кожна частина розбиття є одним елементом розв’язку. Перш за все перед розбиттям необхідно виділити окремо незв’язані частини схеми. Таку задачу може виконати метод ОЗС. Якщо схема має такі частини, то в процесі згортання утвориться не одне дерево, а ліс  $F^R = \{T_1^R, \dots, T_f^R\}$  з двох чи більшого числа дерев. Кожне дерево  $T_i^R \in F^R$ ,  $i=1, \dots, f$ , відповідає одній зв’язаній частині схеми. Першою при декомпозиції повинна бути спроба отримання окремих частин розбиття на основі аналізу лісу  $F^R$ . Виділяємо дві групи підсхем: що задовольняють бажані обмеження і що їх не задовольняють. Підсхеми, що задовольняють бажані обмеження, заносимо в розв’язок та усуваємо з подальшого розгляду. Серед решти підсхем, що не задовольняють обмеження, виділимо окремо більші за розмірами від потрібних та менші. Розгляд решти частини схеми розпочинаємо з підсхем, найбільших за розмірами. З кожної такої підсхеми формуємо ціле число частин розбиття, що задовольняють обмеження, а решту елементів усуваємо для подальшого включення в інші підсхеми. Виникає часткова задача розбиття великої зв’язаної підсхеми на задане число частин з залишком, який буде доданий до підсхем, менших за розміром ніж потрібно для частин розбиття. Саме ця задача є базовою і подальший розгляд буде їй присвячений.

Менші ніж потрібно за розмірами підсхеми разом з “кусками”, відрізнаними від великих підсхем, які розглядалися спочатку, стараємося згрупувати разом в потрібні за розміром частини

розбиття, включити в розв'язок та усунути їх з подальшого розгляду. Тут виникає задача, подібна до типової комбінаторної задачі пакування "рюкзака". Відмінність типової задачі полягає в тому, що "рюкзак" заповнюється неподільними предметами заданого об'єму, причому в "рюкзаку" може залишатися певна частина незаповненого простору. У нашому випадку такий вільний простір не залишається, оскільки заданими є число частин розбиття і відповідно число складових базових елементів, які необхідно "упакувати" в рюкзак. Якщо число елементів виділених частин жорстко не задається, а можливим є певне відхилення від середнього значення, то це сприятиме отриманню кращого розв'язку, оскільки може не завжди вимагати розрізання певного кластера на дві частини, одна з яких входить у виділену частину, а друга переноситься для включення в решту ще несформованих частин. Проблема оптимального розв'язування цієї задачі потребує окремого додаткового дослідження.

Розглянемо базову задачу розбиття без залишку однієї зв'язаної частини схеми на підсхеми, що задовольняють бажані обмеження. Доцільно виділити три основні групи алгоритмів: послідовного, паралельно-послідовного та дихотомічного формування початкових розв'язків. Спільним для всіх них є побудова дерева згортання  $T^R$  методом ОЗС для формування кластерів. Послідовний алгоритм покроково виділяє окремі частини. Основними його кроками є такі: 1) будуємо дерево  $T^R$  до моменту виділення першого кластера  $CI^*$ , розмір якого є рівним бажаному значенню однієї частини розбиття або більшим за нього; 2) якщо розмір кластеру  $CI^*$  відповідає бажаному значенню, то переходимо до п.4, у протилежному випадку визначаємо число елементів  $\Delta n$ , які необхідно усунути з кластера для отримання частини розбиття бажаного розміру; 3) переносимо  $\Delta n$  елементів з кластера  $CI^*$  в решту частину схеми, утворюючи бажану частину  $P^* \in \tilde{P}$ ; 4) від всієї схеми відтинаємо частину  $P^*$  і, якщо ще залишаються непризначені у частини елементи, то оновлюємо дані та повертаємося на п.1 для формування наступної частини розбиття. У протилежному випадку робота алгоритма закінчена.

Паралельно-послідовний алгоритм на першому кроці на основі дерева  $T^R$  виділяє кластери, число яких дорівнює бажаному значенню та які за своїми розмірами найкраще наближаються до бажаних. Далі відбувається коригування розмірів частин.

Дихотомічний алгоритм є низхідним. Ідея його полягає в формуванні частин, кратних за числом елементів до бажаних, на основі дерева згортання  $T^R$  шляхом дихотомічного ділення зверху-вниз. Алгоритм має наступні кроки: 1) будуємо повне дерево згортання  $T^R$ ; 2) виділяємо на дереві  $T^R$  ребро  $v^*$ , що розділяє всю множину

елементів  $P$  на дві підмножини  $P'$  та  $P''$ , найбільш доцільні для їх подальшого рекурсивного дихотомічного дроблення на частини; 3) визначаємо число частин  $k_1$  та  $k_2$  ( $k_1 + k_2 = k$ ), які можуть бути утворені з обох підсхем.; 4) визначаємо число елементів  $\Delta n$ , які необхідно перенести з однієї підсхеми в іншу для формування підсхем з числом елементів, кратним до чисел  $k_1$  та  $k_2$  та переносимо їх процедурами, описаними нижче; 5) формуємо дві підмножини  $P_1$  та  $P_2$ , утворені відповідно з підмножин  $P'$  та  $P''$ . Якщо розмір однієї з підсхем (або обох) відповідає бажаному, то вона заноситься в розв'язок як виділена частина розбиття та з подальшого розгляду усувається. В протилежному випадку більшу за розміром підсхему заносимо в чергу для розгляду та ділення на наступних кроках, а меншу розглядаємо як початкову схему та дробимо поверненням на п.1; 6) беремо наступну з черги підсхему та повертаємося на п.1, розглядаючи її як початкову схему. Якщо черга порожня, то робота алгоритма закінчена.

Відзначимо основні переваги та недоліки описаних алгоритмів. Послідовний алгоритм є найпростішим, проте йому найбільш властивими є "жадібні" риси, а тим самим і незадовільні результати при великій розмірності задачі. В процесі відтинання окремих частин від кластерів залишаються "хвости", з яких формуються останні частини. Ці частини є неякісними, нещільними, з великим числом зовнішніх зв'язків. Недоліком паралельно-послідовного алгоритму є більша складність. Не завжди на дереві згортання вдається знайти розріз, який добре наближається до бажаного розв'язку за числом кластерів та їх розмірністю, тобто виділити добрі початкові ядра для подальшого їх зменшення чи збільшення. Початкові ядра можуть суттєво відрізнитися за розмірами між собою та від бажаного значення, що ускладнює роботу алгоритма. Алгоритм на основі дихотомічного ділення вважаємо найбільш ефективним, придатним для задач великих та надвеликих розмірностей. Він є складнішим від першого, проте на відміну від нього не утворює "хвостів", що погіршують якість останніх виділених частин розбиття. Він обирає найбільш раціональні розрізи схеми для подальшого дроблення на частини. Йому є найменш властивим принцип "жадібності", оскільки виділяє не по одній частині за найкращими умовами, а здійснює паралельний розділ схеми з урахуванням потреб всіх частин одночасно.

#### 4. Основні макропроцедури

Побудова дерева згортання є найбільш важливою макропроцедурою, яку використовують всі описані алгоритми. Основні особливості цієї процедури розглянуто в [2,3]. Якщо в результаті згортання схеми утворюється не одно дерево, а два чи більше їх число, то це вказує на наявність в

схемі більше однієї зв'язаної підсхеми. Кожне дерево відповідає одній такій підсхемі.

Для оптимального виділення та перенесення визначеного числа  $\Delta n$  елементів з однієї підсхеми в іншу пропонуються два підходи. Перший передбачає використання дерева згортання, другий – оціночних критеріїв. Вхідною інформацією для даної макропроцедури є кластер  $CI^*$ , з якого необхідно усунути елементи, та решта частина схеми, яку умовно позначаємо через  $R$ . Решта частина може складатися з однієї або кількох підсхем. Це суттєвого значення не має, оскільки  $R$  розглядається як єдине ціле. Вихідною інформацією повинна бути множина з  $\Delta n$  елементів, які усуваються з кластера  $CI^*$  та переносяться в  $R$ .

Розглянемо можливі підходи детальніше. При використанні першого підходу кластер  $CI^*$  розсіпають та розглядають його на рівні базових елементів. Решта частини схеми  $R$  розглядається як один елемент. На такій множині будується дерево  $T^R$ . Визначаємо число елементів, що приєднуються до дерева на шляху від вершини  $R$  до його кореня (найвищої вершини). Можливими є чотири випадки: 1) це число дорівнює бажаному значенню  $\Delta n$  елементів; 2) воно є більшим за нього; 3) є меншим за нього; 4) дорівнює нулю.

У першому випадку задача розв'язується безпосередньо шляхом включення всіх виділених елементів в розв'язок. У другому випадку йдучи від вершини  $R$  вгору по дереву до кореня знаходимо вершину, при відтинанні якої разом з попередніми вершинами утворюється бажана множина з  $\Delta n$  елементів. Якщо ж такої вершини не знаходимо, але є дві суміжні, при відтинанні яких виникають підсхеми з меншим та більшим від бажаного значення числа елементів, то можливими є дві процедури продовження. Для першої процедури беруть вершину, яка разом з попередніми утворює множину  $\Delta n'$  з меншим від бажаного числа елементів та відповідні елементи на шляху від неї до вершини  $R$  приєднують до  $R$ . Далі процедуру повторюємо виділяючи вже множину  $\Delta n' = \Delta n - \Delta n'$  на редукованій множині початкових елементів без приєднаних  $\Delta n'$  елементів до  $R$ . Процедура рекурсивно продовжуємо до отримання бажаного розв'язку або до моменту, коли першою утворюється вершина з більшим за бажане число елементів.

Для другої процедури, яка виникає також тоді, коли першою на шляху до кореня дерева згортання є вершина, що відповідає кластеру з більшим ніж потрібно числом елементів, доцільною є наступна реалізація. Елементи цього кластера розсіпаються, решта елементів з однієї сторони вершини утворюють нову підсхему  $R'$ , куди входить також частина початкових елементів, а з другої – підсхему  $R''$ , куди входять також елементи множини  $R$ . На утвореній множині будується нове дерево згортання. В процесі

згортання забороняється об'єднуватися підсхемам, що включають одночасно  $R'$  та  $R''$ . Далі на шляху між вершинами  $R'$  та  $R''$  визначають найкращу для розрізання схеми вершину і т.д. аж поки не буде знайдено бажаний розв'язок. На кожному такому кроці редукується множина невизначених елементів, частина елементів призначається або в одну або в другу підсхему. Проте в процесі реалізації першої та другої процедур може виникнути ситуація, коли на шляху є одна або кілька вершин, що відповідають кластерам, елементи яких не приєднуються ні до однієї з двох вершин  $R'$  чи  $R''$ , а згортаються при побудові дерева  $T^R$  повністю окремо. У цьому випадку використовуємо описаний нижче підхід, що реалізує призначення елементів та кластерів на основі оціночних критеріїв для всіх непризначених базових елементів та утворених з них кластерів.

У третьому випадку, коли число елементів, що приєднуються до дерева на шляху від вершини  $R$  до його кореня є меншим за бажане значення  $\Delta n$  елементів, проте частина необхідних елементів виділилася, приєднуємо ці елементи до решти схеми  $R$  та рекурсивно продовжуємо пошук подібним чином. Якщо ж число всіх елементів, що приєднуються визначеними процедурами є недостатнім і на шляху від  $R$  до кореня не появляються при черговій рекурсивній реалізації названої процедури нові вершини, або вони не виникали з самого початку, що відповідає четвертому з наведених вище випадків, необхідно йти від кореня вниз по дереву суміжною гілкою. Проте дати однозначну рекомендацію яким чином виділяти тут нові елементи чи кластери важко. В даному разі краще продовжити виділення решти потрібних для перенесення елементів на основі другого підходу, що використовує оціночні критерії.

Розглянемо процедуру призначення елементів на основі оціночних критеріїв, яка може мати як самостійне значення, так і допоміжне в описаному вище другому випадку. На елементах підсхеми, з якої необхідно виділити для перенесення бажане число елементів, будується дерево згортання  $T^R$ . Решта частина схеми розглядається як єдине ціле у вигляді одного базового елемента  $R$  у четвертому випадку або як два базових елементи  $R'$  та  $R''$  у другому випадку. При цьому беруться до уваги зв'язки з рештою частиною схеми. На дереві виділяються всі кластери, число елементів яких є рівним або меншим від числа елементів, що необхідно перенести. Для кожного такого кластера та базового елемента визначається значення критерію його перенесення в бажану частину (приріст числа зовнішніх зв'язків або інші) та вони відповідно впорядковуються. Наступними можливі дві стратегії – поодинокого призначення (кластера або елемента) з наступною переоцінкою значення критерію або групового призначення, при якому

один раз визначається множина елементів бажаної потужності.

Задача призначення виділених з однієї підсхеми елементів в решту (декілька) зв'язаних з нею підсхем полягає у призначенні заданого числа  $\Delta n$  елементів, виділених з кластеру  $CI^*$ , в решту підсхем  $R_1, \dots, R_r$ . Можливими як і попередньому випадку є два підходи – на основі дерева згортання схеми та оціночних критеріїв. При використанні дерева згортання елементи множини  $\Delta n$  розглядаються як базові нарівні з окремими підсхемами  $R_1, \dots, R_r$ . Будується ліс дерев  $F^R$ , в якому кожне окреме дерево відповідає одній частині з множини  $R_1, \dots, R_r$ . Згортання схеми відбувається з обмеженням, при якому підсхемам, що включають різні частини, об'єднуватися не дозволяється, що й викликає утворення лісу замість одного дерева. В результаті отримуємо  $r$  окремих дерев з приєднаними до них  $\Delta n$  елементами. Другий підхід полягає у використанні оціночних критеріїв та призначенні кластерів та елементів в ту чи іншу підсхему за найкращими їх значеннями.

Для визначення найкращого ребра для розрізання схеми на дві частини можливими є декілька підходів. У першому випадку вибираємо два найвищих кластери. Тут можливою є поява ситуації, коли число елементів одного з кластерів є набагато меншим за потрібне середнє значення  $n_{сеп}$  елементів в одній частині ( $n_{сеп} = n/k$ ). Тоді необхідно опуститися по дереву до ребра так, щоб число елементів однієї з підсхем, що виділяється, не було меншим за  $\alpha n_{сеп}$ , де  $\alpha$  – наперед заданий коефіцієнт, наприклад  $\alpha = 0,7 - 0,9$ .

Другий випадок передбачає вибір такого ребра розрізу, коли число  $\Delta n$  елементів, що необхідно переносити з однієї підсхеми в іншу для забезпечення кратності чисел їх елементів до числа  $n_{сеп}$ , є найменшим. Третій випадок передбачає вибір такого розрізу, коли числа елементів обох підсхем, що утворюються внаслідок розділу, відрізняються найменше. У четвертому випадку вибирається такий розріз, коли число зв'язків розрізу, поділене на число елементів меншої підсхеми, є найменшим.

## 5. Експериментальні дослідження

Обчислювальні затрати для основної процедури – побудови дерева згортання є близькими до лінійних від числа базових елементів [3]. Для розробленої програми розглянуто вісімнадцять тестів фірми IBM, що стали типовими для задач розбиття [4]. Час згортання схеми з 12752 елементів (тест *ibm01*) склав 14 сек., а для схеми з 210613 елементів (тест *ibm18*) – 320 сек. при 100% об'єднанню на кожному кроці (жорстка кластеризація) та 50 сек. і 1253 сек. відповідно при об'єднанні тільки найкращих пар (вільна кластеризація). Розрахунки виконано на ПК з процесором Intel-Celeron 450.

Експерименти [5], підтвердили добру якість отриманих результатів на множині відомих тестів при сумісному використанні конструктивних та оптимізаційних алгоритмів [6]. Для дванадцяти розглянутих тестів результати не є гіршими ніж отримані іншими методами, у п'яти випадках одержані кращі від відомих (тести C5315, C7552, s13207, s38417, s38584), у восьми випадках результати є оптимальними (тести C499, C1355, C1908, C3540, C7552, C6288, s15850, s38584).

## 6. Висновки

Розглянуті алгоритми конструктивного розбиття дозволяють задачу великої розмірності рекурсивно зводити до розв'язування аналогічних задач меншої розмірності. Як простішим для реалізації вважаємо метод послідовного типу; кращі результати слід очікувати від дихотомічного методу. Алгоритми використовують однакові базові макропроцедури, що дозволяє побудувати єдиний пакет програм, який реалізує всі зазначені підходи. Це створить умови для розв'язування кожної задачі кількома алгоритмами з подальшим вибором найкращого результату. Бажаним є їх сумісне використання з оптимізаційними алгоритмами. Для цього можуть бути використані будь-які ітераційні алгоритми оптимізації розбиття, проте доцільнішим вважаємо використання підходів, оснований на методі оптимального згортання схеми [6], що дозволить застосувати основні виділені тут макропроцедури, які є спільними в обох випадках.

## 7. Література

1. Sadiq M.Sait, Habib Youssef. *VLSI physical design automatic, Theory and practice*. New York, IEEE Press, 1995.
2. Базилевич Р.П. *Декомпозиционные и топологические методы автоматизированного конструирования электронных устройств*. Львов: Вища школа. Изд-во при Львов. гос. ун-те, 1981. -168с.
3. Базилевич Р.П., Подольський І. *Ієрархічна кластеризація складних схем*. Вісник ДУ "Львівська політехніка", № 392, "Комп'ютерна інженерія та інформаційні технології", 2000.
4. Alpert C.J. *The ISPD-98 Circuit Benchmark Suite*. Proc. ACM/IEEE Intern. Symposium on Physical Design, April 1998, pp.80-85.
5. Bazylevych R.P., Melnyk R.A., Rybak O.G. *Circuit partitioning for FPGAs by the optimal circuit reduction method*. VLSI Design. Vol 11, No 3, 2000.
6. Базилевич Р. *Ітераційні алгоритми оптимізації розбиття складних систем на основі оптимального згортання*. В даному збірнику.