Робота виконувалась у межах держбюджетної теми "Розробка методів та засобів розподілення обчислень в задачах теплового проектування електронних пристроїв нового покоління" ДБ Діаграма.

1. Leslie Lamport Password Authentication with Insecure Communication // Communications of the ACM, Vol. 24 (1981), No 11, pp. 770–772. 2. Shiuh-Jeng Wang, Jin-Fu Chang Smart card based secure password authentication scheme // Computers & Security, Vol. 15 (1996), No. 3, pp. 231–237. 3. Wen-Her Yang, Shiuh-Pyng Shieh Password Authentication Schemes with Smart Cards // Computers & Security, Vol. 18 (1999), No.8, pp.727–733. 4. Wen-Shenq Juang Efficient password authenticated key agreement using smart cards // Computers & Security, Vol. 23 (2004), pp. 167–173. 5. Raphael C.-W. Phan Cryptanalysis of two password-based authentication schemes using smart cards // Computers & Security, Vol. 25 (2006), pp. 52–54.6. Wen-Gong Shieh, Jian-Min Wang Efficient remote mutual authentication and key agreement // Computers & Security, Vol. 25 (2006), pp. 72–77. 7. Chien H.Y., Jan J.K., Tseng Y.H. An efficient and practical solution to remote authentication: smart card. // Computers & Security, Vol. 21 (2002), No. 4, pp. 372–375. 8. A. Jr Evans, W. Kantrowitz, and E. Weiss A user authentication system not requiring secrecy in the computer // Communications of the ACM, Vol. 17 (1974), pp. 437–442. 9. R.E. Lennon, S.M. Matyas, and C.H. Mayer Cryptographic authentication of time-invariant quantities // IEEE Trans. on Communications, Vol. COM-29 (1981), No. 6, pp. 773–777. 10. K. Tan, and H. Zhu Remote password authentication scheme based on cross-product // Computer communications, Vol. 18 (1999), pp. 390–393. 11. Столлингс В. Криптография и защита сетей: принципы и практика. – М.: Вильямс, 2001. – 672 с. 12. Нильс Фергюсон, Брюс Шнайер Практическая криптография. – М.: Диалектика, 2004. – 432 с.

УДК 621.382

Р. Базилевич, А. Ждан
Національний університет "Львівська політехніка",
кафедра програмного забезпечення

АЛГОРИТМИ ПОСЛІДОВНОГО ПАКУВАННЯ СИЛЬНОЗВ'ЯЗНИХ ЧАСТИН СХЕМ ІЗ ЗАДАНИМИ ОБМЕЖЕННЯМИ

Розглянуто декілька стратегій та алгоритмів послідовного пакування схем із заданими обмеженнями. Розкрито особливості різних стратегій пакування схем із заданими обмеженнями.

A few strategies and algorithms line packing of charts with the set limitations. The features of different strategies of packing of charts are exposed with the set limitations.

Вступ

Пакування складних схем мінімальною кількістю ПЛМ (програмованих логічних матриць) є однією з важливих задач, які виникають при проектуванні сучасних засобів комп'ютерної техніки. Кількість елементів та зовнішніх зв'язків кожного утвореного модуля є найважливішими обмеженнями, які необхідно задовольняти при розбитті схем на декілька частин. Задача з математичного погляду належить до важкорозв'язуваних неполіноміальних комбінаторних

оптимізаційних задач. Проблема розглядалася багатьма авторами [2–5], проте отримувана за допомогою існуючих алгоритмів якість є недостатньою. Класичний підхід до розв'язання задачі пакування у більшості випадків поділяється на дві підзадачі: одержання початкового розв'язку та подальша його оптимізація. Пропонуються нові стратегії та алгоритми в задачі пакування, які використовують метод оптимального згортання схеми [1].

Формулювання задачі

Задано схему $N=\{P,E\}$, де

$P=\{p_1,\ldots, p_n\}$ – множина елементів,

$E=\{e_1,\ldots, e_m\}$ – множина зв'язків. (1.1)

На основі цієї вхідної інформації формуємо систему множин зв'язків, інцидентних до кожного елемента:

$$E(P)=\{E(p_1),\ldots,E(p_n)\},\ (\forall p\in P)\ [E(p)=\{e \mid e \text{ є інцидентним до } p\}]; \quad (1.2)$$

та систему множин елементів, інцидентних до кожного зв'язку:

$$P(E)=\{P(e_1),\ldots,P(e_m)\},\ (\forall e\in E)\ [P(e)=\{p \mid p \text{ є інцидентним до } e\}]. \quad (1.3)$$

Системи (2) та (3) є взаємно відповідними: $P\leftrightarrow E$.

Початкові дані містять певні характеристики вхідних елементів та зв'язків, які повинні бути враховані при прийнятті рішення (такі, як площа елементів або затримка у поширенні сигналів). Звичайно найважливішими обмеженнями є кількість елементів n_i та зовнішніх зв'язків m_i^{ex}, які не можуть перевищити заданого значення для кожної з утворених частин (ПЛМ):

$$n_i \le n_{i\,max}\ ,\quad m_i^{ex} \le m_i^{ex}{}_{max.} \quad (1.4)$$

Необхідно отримати таке розбиття P^* множини елементів P, щоб мінімізувати загальну кількість частин:

$$P \to P^*=\{P_1,\ldots, P_k\}\ ,\quad k \to min, \quad (1.5)$$

при забезпеченні виконання заданих обмежень:

$$(\forall P^i\in P^*)\ [(n_i \le n_{i\,max})\&(m_i^{ex} \le m_i^{ex}{}_{max})].$$

Стратегія послідовного пакування схем

Стратегія послідовного пакування схем полягає у визначенні бажаних підсхем у процесі побудови дерева оптимального згортання схем.

Стратегію послідовного виділення підсхем можна реалізувати двома підходами:

1. Формування підсхем (кластерів), менших за розмірами (кількість елементів менша за потрібну), що задовольняє певні обмеження, або потрібного розміру.

2. Побудова підсхем (кластерів) потрібного або більшого розміру, що задовольняє певні обмеження.

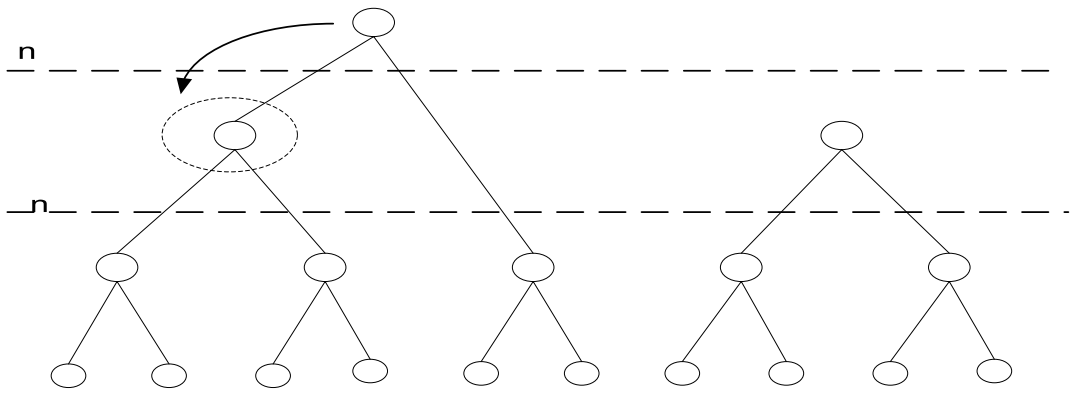

Рис. 2.1. Знаходження підсхем, що не перевищує задані обмеження

Використовуючи перший підхід, дерево згортання будується до появи першого кластера, який перевищує обмеження, які накладаються на розмір кластера. Для знаходження кластера, що не

перевищує обмеження, необхідно перейти на нижчий рівень (рис. 2.1) і у випадку, якщо на цьому рівні є декілька кластерів, вибрати кластер максимального розміру з мінімальною кількістю зовнішніх зв'язків. У знайдений таким чином кластер(підсхему), потрібно добирати елементи, доводячи кластер до потрібних розмірів. Добір може відбуватися одразу після побудови підсхеми із елементів, що не увійшли до цієї підсхеми, або після побудови всіх підсхем, доповнюючи кожну з підсхем із залишку елементів, що не увійшли до жодної сформованої підсхеми.

Елементи до підсхем добирають за критерієм максимальної зв'язності елементів із підсхемами, тобто першим долучається елемент, що має найбільшу кількість зв'язків із підсхемою. Також можливий критерій відношення кількості зв'язків елемента з певною підсхемою до всієї кількості зв'язків елемента; за цим критерієм також першими долучатимуться елементи з максимальним значенням цього відношення.

За другим підходом, формуючи підсхеми, розмір яких перевищує задані обмеження, дерево згортання будується до появи першого кластера, який перевищує обмеження. У знайденому так кластері (підсхемі) необхідно вилучати зайві елементи.

Критерієм вилучення елемента із підсхеми може бути максимальне значення відношення зовнішніх зв'язків елемента (зв'язків з іншими підсхемами) до усієї кількості зв'язків елемента. Вилучення зайвих елементів можна проводити відразу після формування підсхеми або після формування всіх підсхем. У разі вилучення елементів після формування всіх підсхем вилучені елементи формують залишок, з якого буде проводитись добір елементів у підсхеми з недостатньою кількістю елементів.

Для корегування розміру кластеру (підсхем), а також кількості його зовнішніх зв'язків можна застосовувати декілька підходів вилучення або добору елементів:

1. Перенесення елементів
2. Перенесення кластерів
3. Перенесення плаваючих елементів (рис. 2.2.).

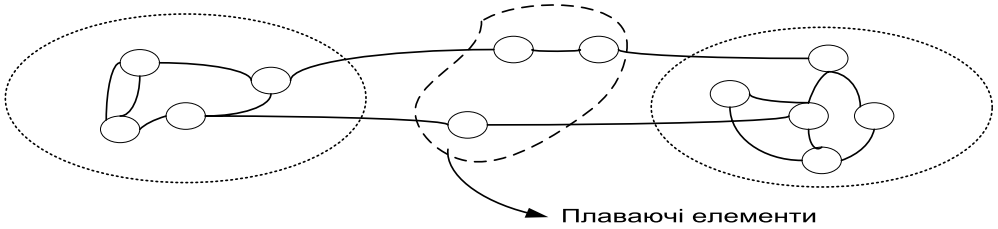

Рис. 2.2. Схематичне зображення плаваючих елементів

Під перенесенням розуміється процес вилучення або вставки елементів до підсхеми.

Перенесення елементів є найпростішим способом коригування параметрів підсхеми. Цей спосіб передбачає визначення ефективності перенесення для кожного елемента і формування впорядкованого списку ефективностей перенесень. У випадку вставки визначають ефективність перенесення для елементів, що не увійшли до знайденої підсхеми, а у випадку вилучення визначають ефективність перенесення для елементів знайденої підсхеми.

При кожному перенесені елемента необхідно перераховувати ефективності перенесення для елементів, що залишились, якщо при перенесенні елемента відбулась зміна у зовнішніх зв'язках підсхеми.

Недоліком підходу є те, що кожне перенесення елемента може зумовлювати перерахунок ефективностей для великої кількості елементів.

Підхід із перенесенням кластерів, на відміну від підходу із перенесенням елементів потребує побудови дерева оптимального згортання для елементів, що не увійшли до знайденої підсхеми. Після побудови дерева оптимального згортання знаходять ефективність перенесення кластерів, а також формують впорядкований список ефективностей перенесення.

При перенесенні кластерів необхідно враховувати, що кластери можуть входити один до одного, тому при перенесенні певного кластера необхідно вилучати із списку ефективностей перенесення кластерів кластери, що входять до перенесеного кластера. Так, як і за першим підходом із перенесенням елементів, при перенесенні кластерів необхідно перераховувати ефективності перенесення для кластерів, що залишились, якщо при перенесенні кластера відбулась зміна у зовнішніх зв’язках підсхеми.

Недоліком цього підходу є те, що при знаходженні нової підсхеми необхідно будувати дерево оптимального згортання для елементів, що не увійшли до цієї підсхеми. Перевагою є те, що перенесення кластера є перенесенням певної групи зв'язаних елементів за один підхід, що пришвидшує процес корегування параметрів знайденої підсхеми.

Перенесення плаваючих елементів дає змогу переносити ці елементи, корегуючи розмір підсхеми і не змінюючи кількості зовнішніх зв’язків цієї підсхеми. Цей підхід не може застосовуватися самостійно для корекції параметрів підсхем, оскільки він корегує тільки розмір підсхем і може застосовуватися за наявності плаваючих елементів, кількість яких є незначною.

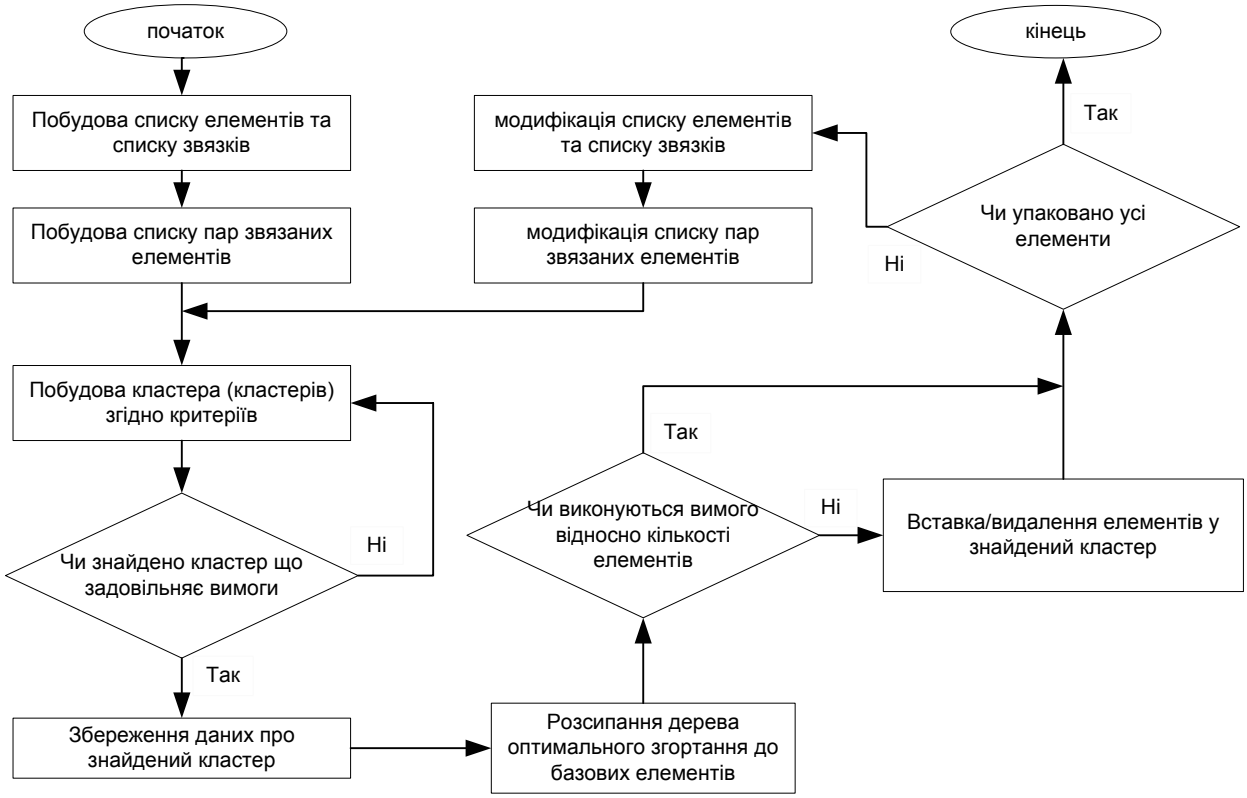

Рис. 2.1.1. Блок-схема роботи алгоритму розбиття схем в процесі побудови дерева оптимального згортання

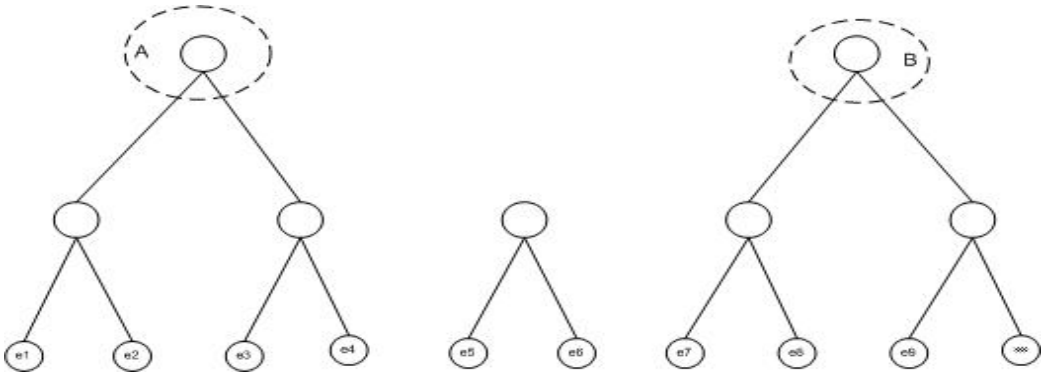

Рис. 2.1.2. Дерево оптимального згортання в момент знаходження підсхем

Алгоритми послідовного пакування схем в процесі побудови дерева оптимального згортання схем

Ідея запропонованого методу полягає в зупинці процесу побудови дерева оптимального згортання схем при виявленні першого кластера, що задовольняє задані обмеження (1.4). Одержаний таким шляхом розв'язок корегуємо добором або вилученням елементів із підсхеми. Елементи знайденого кластера (підсхеми) відділяються від загальної множини елементів схеми. Процес побудови дерева згортання починається з початку для елементів, що не увійшли до знайденої підсхем.

Основні кроки алгоритму є такими:

Алгоритми послідовного пакування схем в процесі побудови дерева оптимального згортання
Вхідні дані: *Файла даних звязків та елементів схеми*
1. Побудова списку звязків та елементів схеми на основі вхідних даних 2. Побудова списку звязаних елементів
3. Побудова кластера згідно критерію згортання (формування дерева згортання схем)
4. Перевірка кластера на відповідність заданим обмеженям, якщо обмеження виконується здійснюється перехід на крок 5 і накше на крок 3
5. Відділяємо знайдений кластер від решти кластерів.
4. Розсипаємо дерево оптимального згортання до базових елементів за виключенням знайденого кластера.
5. При необхідності здійснюється добір, або вилучення елементів у кластері
6. Здійснення модифікації списку пар звязаних елементів і перехід на крок 3 поки не буде знайдено всіх підсхем.
6. Збереження результатів роботи. **Вихідні дані: Масив списків елементів знайдених підсхем**

Алгоритми послідовного пакування схем на основі побудованого дерева оптимального згортання

Ідея запропонованого методу полягає у послідовному виділенні підсхем, що задовольняють потрібний розв'язок або є близькими до нього. Для цього будують дерево оптимального згортання схем. На основі інформації, що дає дерево оптимального згортання, шукають підсхему, що задовольняє або є близькою до заданих обмежень. Цей пошук полягає в спуску із верхнього рівня дерева на нижчі рівні дерева оптимального згортання схем. Знайдена підсхема (кластер) відділяється від решти елементів дерева; для елементів, що не увійшли до знайденої підсхеми, будуємо дерево оптимального згортання схем. Цей процес триває циклічно доки не буде знайдено всіх підсхем. Для підсхем, кількість елементів в яких є меншою або більшою за потрібну, добираємо або вилучаємо елементи.

Основні кроки алгоритму є такими:

Алгоритми послідовного пакування схем на основі побудованого дерева оптимального згортання
Вхідні дані: *Спискова структура даних пар зв'язаних елементів схеми*
1. Побудова дерева оптимального згортання схем.
2. Визначення кластера, що є максимально близьким до потрібного розміру
3. Відділяємо знайдений кластер від решти кластерів.
4. Розсипаємо дерево оптимального згортання до базових елементів за винятком знайденого кластера.
5. Виконуємо крок 1 для елементів, отриманих на кроці 4, доки не буде знайдено всіх підсхем.
6. За умови виникнення залишку елементів розподіляємо їх між частинами підсхеми, які не набрали потрібного розміру, враховуючи зв'язність елементів з цими частинами.
Вихідні дані: Масив списків елементів знайдених підсхем

Нижче наведено приклад роботи запропонованого алгоритму. На рис. 2.2.2 відображається дерево згортання всієї схеми із виділеною першою підсхемою. На рис. 2.2.3 відображено дерево згортання схеми без врахування елементів першої підсхеми із виділеною другою підсхемою. В ситуації, коли виникає залишок із елементів, що не увійшли до знайдених підсхем, він розподіляється між ними, доповнюючи недостачу елементів у цих підсхемах (рис. 2.2.4).

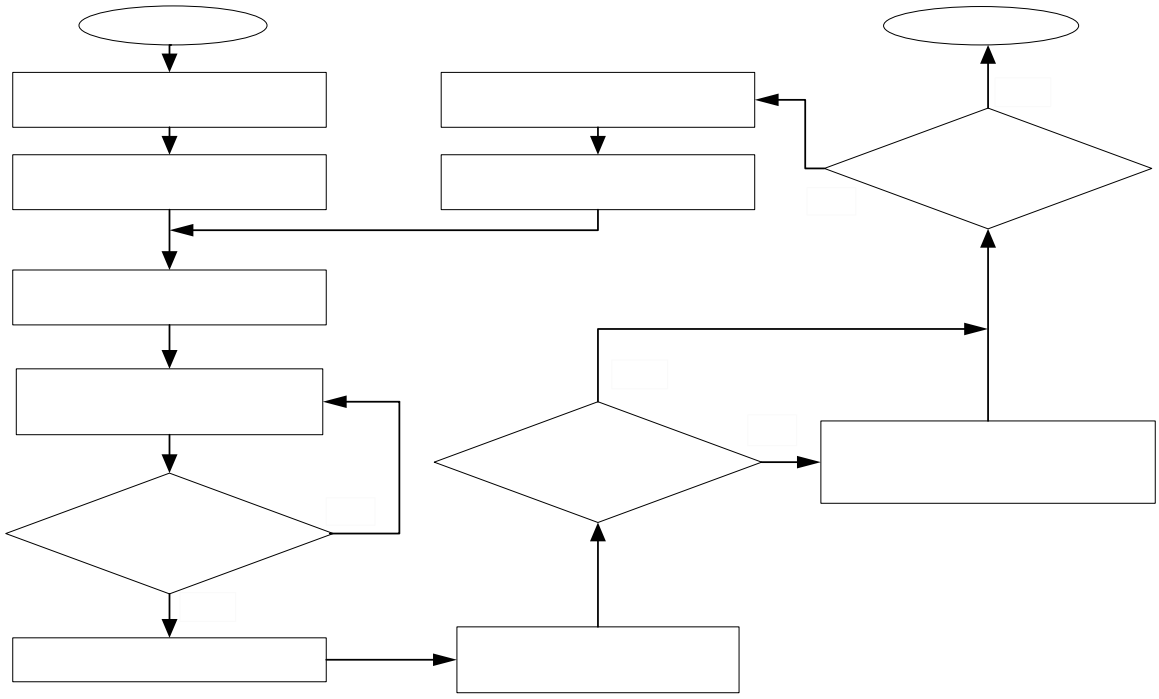

Рис. 2.2.1. Блок-схема роботи алгоритму розбиття схем

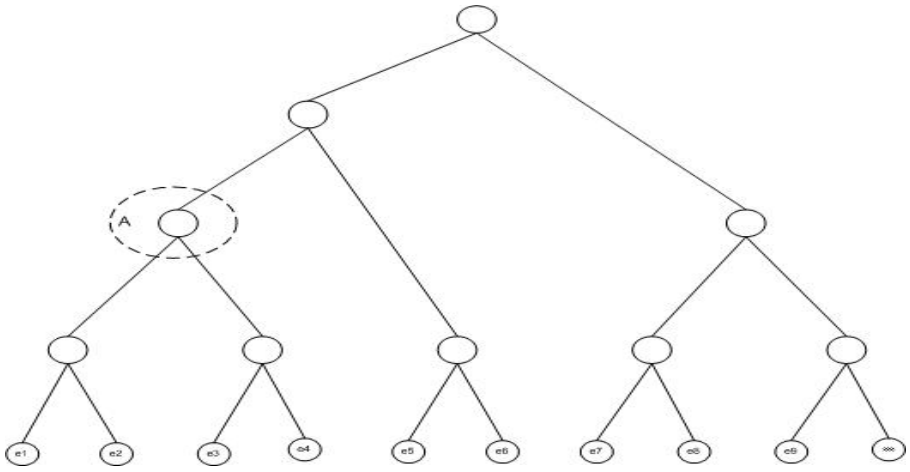

Рис. 2.2.2. Дерево оптимального згортання із виділеною першою підсхемою

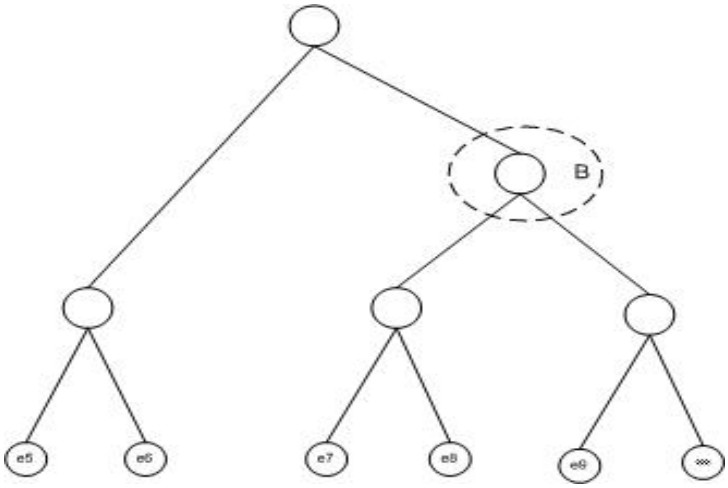

Рис. 2.2.3. Дерево оптимального згортання із елементів між двома знайденими підсхемами

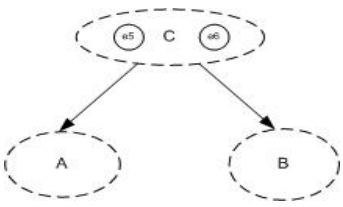

Рис. 2.2.4. Розподіл залишку

Недоліком цього алгоритму є те, що необхідно постійно перебудовувати дерево оптимального згортання схем для знаходження кожної з підсхем.

Висновок

Запропоновано та розкрито особливості стратегій та алгоритмів послідовного пакування схем із заданими характеристиками та обмеженнями. Розглянуті алгоритми піддаються реалізації на ПК і можуть бути базою для реалізації ефективних програмних засобів вирішення задачі пакування схем із заданими характеристиками та обмеженнями.

1. Базилевич Р.П. Декомпозиционные и топологические методы автоматизированного конструирования электронных устройств. – Львов: Вища школа: Изд-во при Львов. гос. ун-те, 1981. – 168 с. 2. Базилевич Р.П., Подольський І.В. Особливості організації пакету програм для ієрархічної кластеризації схем // Вісник Нац. ун-ту "Львівська політехніка" "Радіоелектроніка та телекомунікації". – 2002. – № 440. – С. 139–144. 3. Базилевич Р.П., Подольський І.В. Ієрархічна кластеризація – ефективний засіб розв'язування неполіноміальних комбінаторних задач схемного типу високої розмірності // Штучний інтелект. – 2002. – № 3. – С. 474–483. 4. Charles J. Alpert, The ISPD98 Circuit Benchmark Suite. ISPD98 Monterey CA USA, 1998. 5. R.. P. Bazylevych, R . A . Melnyk and O. G. Rybak. "Circuit Partitioning for FPGAs by the Optimal Circuit Reduction Method", VLSID DESIGN 2000, Vol. 11, No. 3, pp. 237–248.

УДК 519.878.5

І. Дияк[*], М. Копитко, А. Коркуна
Львівський національний університет імені Івана Франка,
[*]кафедра прикладної математики,
кафедра математичного моделювання соціально-економічних процесів

КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ НАПРУЖЕНО-ДЕФОРМОВАНОГО СТАНУ ПРОСТОРОВИХ КОНСТРУКЦІЙ

Досліджено напружено-деформований стан просторової конструкції складної форми за допомогою програмного комплексу Comsol Multiphysic 3.4. Показано, що використання заданих за замовчуванням у Comsol Multiphysic 3.4 параметрів побудови сітки методу скінченних елементів приводить до отримання неправильних розв'язків задачі. Проведено додаткові числові експерименти, які дали змогу виробити правильний алгоритм розв'язування задач деформування трубоподібних конструкцій. Запропоновано підхід, за яким можна за незначного збільшення обчислювальних затрат розв'язати задачу.

Research of elastic problem of a spatial complex form construction is carried out with Comsol Multiphysic 3.4 program. It is shown, that use default parameters of finite elements mesh of construction at Comsol Multiphysic 3.4. gives incorrect numerical results of a problem. Additional numerical experiments which have allowed to develop correct algorithm of the decision of problems of deformation of pipelike designs was carried out. The approach which allows to receive the correct results at insignificant increase of computing resources is offered.

Постановка задачі

Інженерна конструкція як просторовий об'єкт, який зображений на рис.1, перебуває під дією внутрішнього навантаження постійної інтенсивності. Необхідно визначити її напружено-деформований стан.