

Пошук оптимальних дерев на основі методу імітації відпалу

Доц., к.т.н. Ю. Цимбал, студ. А. Асіїв, студ. Т. Грицько

Lviv Polytechnic National University, Automated Control Systems Department

28a/804 Bandera St., Lviv-13, 79013, Ukraine

E-mail: yurij.tsymbal@gmail.com

Abstract. The problem of the search for optimal combinatorial designs on the example of Leech and Golomb trees has been considered. Criteria of optimality for these trees have been generalized. The application of a simulated annealing to the finding near-optimal trees has been suggested. Outcomes of the application of the algorithm to the finding some optimal trees with 10-12 edges have been presented.

Keywords: optimal combinatorial design, Leech tree, Golomb ruler, simulated annealing.

Вступ

Оптимальні комбінаторні структури [1] є достатньо цікавими та різноманітними математичними об'єктами, які реалізують загальні поняття балансу та симетрії. Такі структури знайшли застосування у технологіях заводського кодування, системах захисту інформації, при плануванні експериментів, у математичних іграх та головоломках, як джерело вхідних даних при розробці та відлагодженні методів, алгоритмів та програмних засобів для розв'язання оптимізаційних задач [2-5].

Постановка задачі

Об'єктом нашого дослідження є так звані *оптимальні дерева*. Це дерева, ребра яких позначаються натуральними числами, що задають їхню довжину. Відстань між двома вершинами є сумою довжин ребер на шляху між ними.

Оптимальним за Лічем [6] вважається таке дерево, в якому усі відстані між вершинами утворюють *найдовшу* можливу (для даної топології дерева) послідовність чисел натурального ряду від 1 до S_1 . Значення відстаней *можуть* повторюватися (рис. 1а, 1б). При цьому дерево буде *обмеженим*, якщо серед відстаней відсутні значення більші за S_1 , і *необмеженим*, якщо присутні (максимальне значення відстані між вершинами позначимо S_2) [2, 3, 6].

Оптимальним за Голомбом вважається таке дерево (як правило, необмежене), значення відстаней в якому *не можуть* повторюватися і максимальна відстань між вершинами S_2 є *найкоротшою* серед можливих для даної топології дерева (рис. 1в).

Ідеальним будемо називати обмежене дерево, в якому відстані між вершинами утворюють *найдовшу* можливу (для даної топології) послідовність чисел натурального ряду *без повторень*. При цьому $S_1 = S_2 = n \cdot (n + 1) / 2$, де n – розмірність дерева (кількість ребер). Тобто ідеальне дерево є оптимальним одночасно і за Лічем і за Голомбом (рис. 1г, 1д).

Отже оптимальні дерева слід вважати узагальненням таких відомих різновидів оптимальних комбінаторних структур як оптимальні лінійки Голомба [7] та

розріджені лінійки [8]. Зазначимо, що для пошуку лінійок Голомба був створений глобальний проект розподілених обчислень OGR [9], результатом багаторічної роботи якого є виявлення всіх оптимальних лінійок з числом позначок до 27 включно.

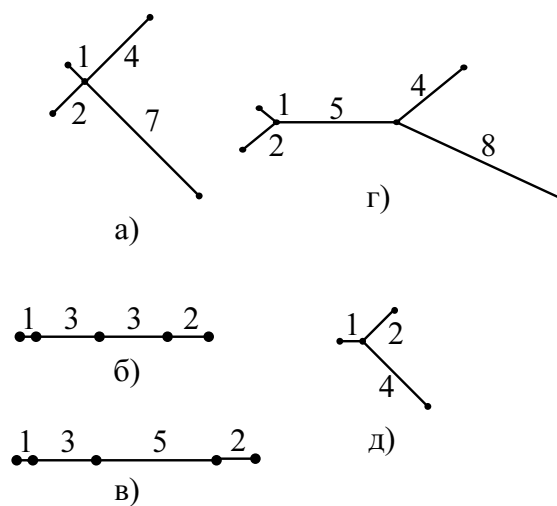


Рис. 1. Приклади оптимальних дерев:
а) необмежене оптимальне за Лічем ($S_1=9$; $S_2=11$);
б) обмежене оптимальне за Лічем ($S_1=9$);
в) оптимальне за Голомбом ($S_2=11$);
г) ідеальне за Лічем і Голомбом ($S_1=S_2=15$);
д) ідеальне ($S_1=S_2=6$).

Метою досліджень є створення програмної системи для пошуку дерев, які будуть оптимальними за вказаними критеріями для заданих топологій різної розмірності.

Аналіз методів досліджень

На сьогодні відомими є прямі методи пошуку лише для деяких різновидів дерев. Для знаходження лінійок Голомба використовується апарат циклічних різницевих множин [10]. Для лінійок, оптимальних за Лічем аналітичний спосіб побудови запропонував Віхман [8, 11]. Для дерев топології “штанга” (рис. 1г) досить вдалий підхід до маркування запропонував сам Дж. Ліч [6].

За участі одного з авторів проводились дослідження можливих способів пошуку оптимальних дерев. Зокрема, розглядався повний перебір варіантів маркування для заданої топології за допомогою системи розподілених обчислень [12, 13]. Такий підхід дає змогу гарантовано відшукати оптимальні дерева кожного типу, проте через експоненційну складність повний перебір при кількості ребер від 10 і більше є практично неможливим.

Тому доцільним є використання евристичних методів локального пошуку. У попередніх дослідженнях [14, 15] проаналізовано та реалізовано з використанням розподілених обчислень підхід на основі генетичних алгоритмів, який показав досить високі практичні результати при кількості ребер від 3 до 11. До певних недоліків можна віднести складність програмної реалізації та досить значну кількість параметрів налаштування. Тому вважаємо за доцільне розглянути також альтернативні евристичні підходи.

Алгоритм пошуку оптимальних дерев

Пропонується застосувати для пошуку оптимальних дерев алгоритм на основі *методу імітації відпалу* [16, 17].

Цей метод використовується для розв'язання задач глобальної оптимізації, в якому процедура пошуку розв'язку імітує фізичний процес отримання низькоенергетичних станів твердої речовини у термостаті. Тверду речовину спочатку нагрівають до температури плавлення, а потім повільно охолоджують до стану теплової рівноваги.

За допомогою моделювання такого процесу шукається точка або множина точок, на яких досягається мінімум певної числової функції.

При цьому проводиться аналогія між станами речовини та допустимими розв'язками, енергією стану та вартістю розв'язку (значенням цільової функції), станом за температури плавлення і початковим розв'язком (можливо, генерованим випадковим чином), малою зміною стану і елементарною зміною розв'язку, температурою і параметром керування, який задає розмір околу для пошуку наступного розв'язку.

Метод імітації відпалу на відміну від багатьох інших методів оптимізації намагається уникнути потрапляння у локальні оптимуми за допомогою можливих випадкових кроків у напрямку розв'язків з гіршим значенням цільової функції.

Для задачі пошуку оптимальних дерев допустимий розв'язок можна зобразити у вигляді одновимірного масиву додатних цілих чисел, які задають довжини ребер, у порядку, що визначається батьківським кодом відповідної топології дерева [18]. У батьківському коді кожен елемент відповідає певній вершині дерева, а його значення задає номер вершини-сусіда серед попередніх елементів. Наприклад, лінійка, що зображена на рис. 1б, 1в., має батьківський код 0,1,2,3,4, а "зірка" на рис. 1д. – код 0,1,1,1. Для оптимальних розв'язків значення у масивах довжин ребер будуть, відповідно, 1,3,5,2 та 1,2,4.

У залежності від топології дерева і значень довжин ребер для кожного допустимого розв'язку можна обчислити кількість пропусків та повторень значень у масиві відстаней між вершинами. Тоді цільову функцію (енергію стану) в загальному вигляді можна зобразити як

$$E = K_{gaps} \cdot n_{gaps} + K_{reps} \cdot n_{reps} \Rightarrow \min ,$$

де n_{gaps} – кількість пропусків, n_{reps} – кількість повторень; K_{gaps} , K_{reps} – коефіцієнти ваги пропусків та повторень, відповідно.

Очевидно, що для обмежених дерев Ліча $S_1 = n \cdot (n+1)/2 - E$, а для дерев Голомба – $S_2 = n \cdot (n+1)/2 + E$, тобто від значення енергії безпосередньо залежить значення відповідного критерію оптимальності S_1 або S_2 .

Підібрані експериментально значення коефіцієнтів K_{gaps} , K_{reps} для різних типів дерев подані в табл. 1.

Таблиця 1

Значення коефіцієнтів ваги пропусків та повторень для дерев Ліча та Голомба при пошуку за методом імітації відпалу

Тип дерева	Коефіцієнт ваги пропусків K_{gaps}	Коефіцієнт ваги повторень K_{reps}
Голомба (G)	1	1000
Обмежене Ліча (RL)	1000	1
Необмежене Ліча (UL)	1	1

Пропонується використовувати почергово елементарні зміни двох типів:

- незалежне збільшення/зменшення випадково обраного елемента розв'язку на випадкове значення, що не перевищує 1/2 суми всіх елементів;
- одночасне збільшення одного випадково обраного елемента розв'язку на випадкове значення, що не перевищує 1/2 суми всіх елементів і зменшення другого елемента на таке саме значення.

При цьому передбачається заборона появи серед елементів від'ємних значень та забезпечення наявності одиниці, як додаткова евристика для досягнення кращих значень енергії та критерію оптимальності.

Для реалізації алгоритму було обрано мову програмування Java. До її переваг, зокрема, можна віднести широкий набір засобів для розробки клієнт-серверних додатків.

Програмна система є веб-проектом, який надає користувачам наступні можливості, доступні через браузер на сторінці проекту:

- задання кількості вершин у дереві;
- задання топології дерева через його батьківський код;
- вибору типу дерев (необмежених та обмежених дерев Ліча, дерев Голомба);
- задання значень параметрів алгоритму (початкової та кінцевої температури, коефіцієнта її зміни, кількості кроків пошуку на кожній температурі);
- пошуку оптимальних дерев із заданими характеристиками;
- обчислення енергії для отриманого розв'язку та значення цільової функції відповідно до типу дерева;
- візуалізації заданого батьківським кодом дерева та його маркування;
- сортування, пошуку та фільтрації результатів експериментів за всіма названими ознаками.

Результати досліджень

Дослідження проводились для різних топологій дерев з 10-12 ребрами. Виконано пошук оптимальних дерев усіх трьох типів: Голомба (G), необмежених (UL) та обмежених (RL) дерев Ліча різних топологій, зокрема, “зірки” та “лінійки”, а також “гусениці” (рис. 2).

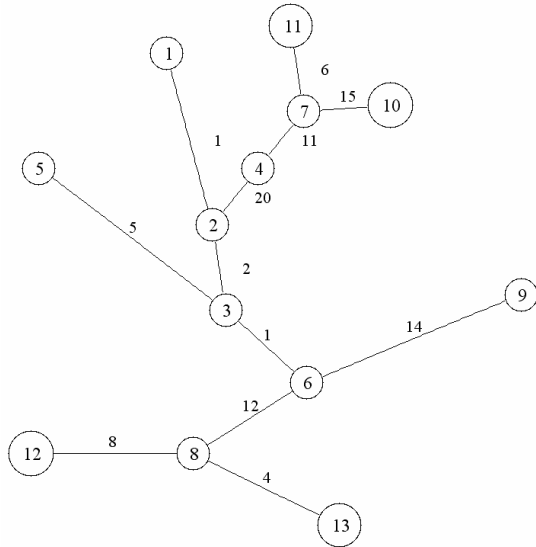


Рис. 2. Близьке до оптимального дерево Ліча з 12 ребрами ($S_1=52$).

Деякі результати проведених експериментів подано в табл. 2.

Таблиця 2

Значення енергії та критерію оптимальності для дерев, знайдених з використанням методу імітації відпалу

Тип дерева	Кількість ребер	Батьківський код	Маркування	Енергія	Значення критерію оптимальності
G	10	0,1,1,1,1,1,1,1,1,1	1,18,4,75,77,24,46,39,9,12	98	153
UL	10	0,1,1,1,1,1,1,1,1,1	2,1,5,14,7,12,24,20,23,4	20	34
RL	10	0,1,1,1,1,1,1,1,1,1	1,6,15,3,12,1,19,17,9,18	17	38
G	11	0,1,2,3,4,5,6,7,8,9,10,11	4,20,9,6,1,27,26,12,13,19,18	89	155
UL	11	0,1,2,3,4,5,6,7,8,9,10,11	2,2,3,2,8,8,8,6,5,1	15	41
UL	11	0,1,2,3,4,5,6,7,8,9,10,11	2,2,6,2,5,9,5,11,3,1,3	19	47
RL	12	0,1,2,2,3,3,4,6,6,7,7,8,8	1,2,20,5,1,11,12,14,15,6,8,4	26	52

Як видно з п'ятого та шостого експериментів для необмежених дерев Ліча немає прямої відповідності між значеннями енергії та критерію оптимальності, тобто вище значення критерію S_1 може досягатись за гіршого значення енергії. Для п'ятого експерименту відстані між вершинами будуть мати значення від 1 до 41 та від 43 до 53 з 1 пропуском (42) та 14 повтореннями; для шостого відстані матимуть значення від 1 до 47 та 49 з 1 пропуском (48) та 18 повтореннями. Тому вибір вигляду цільової функції з урахуванням критерію оптимальності для необмежених дерев Ліча є предметом подальших досліджень.

Висновки

Розроблено тестову модель веб-орієнтованої програмної системи для пошуку близьких до оптимальних дерев на основі методу імітації відпалу.

Проведено експерименти щодо пошуку оптимальних дерев з числом ребер від 10 до 12, які підтвердили коректність розроблених алгоритмів і програмних модулів та показали перспективність застосування методу імітації відпалу для даної задачі.

[1]. Colbourn C.J., Dinitz J.H. Handbook of Combinatorial Designs (2 ed.). – Chapman & Hall - CRC, 2007. – 1018 p.

[2]. Bloom G.S., Golomb S.W. Applications of Numbered Undirected Graphs // Proceedings of the IEEE. – 1977. – Vol. 65, No. 4, pp. 562-570.

[3]. Bloom G.S., Golomb S.W. Numbered Complete Graphs, Unusual Rulers, and Assorted Applications // Theory and Applications of Graphs. – Springer, 1978, pp. 53-65.

[4]. Різник В.В. Синтез оптимальних комбінаторних систем. – Львів: Вища школа. Вид-во при Львів. ун-ті, 1989. – 168 с.

[5]. Наукова школа професора Володимира Різника <http://iknit.lp.edu.ua/riznyk>.

[6]. Leech J. Another Tree Labelling Problem // The American Mathematical Monthly. – 1975. – Vol. 82, No. 9, pp. 923-925.

[7]. Golomb Ruler <http://mathworld.wolfram.com/GolombRuler.html>.

[8]. Luschny P. Perfect And Optimal Rulers <http://www.luschny.de/math/rulers/introe.html>.

[9]. Project OGR <http://www.distributed.net/OGR/>.

[10]. Drakakis K. A Review of the Available Construction Methods for Golomb Rulers // Advances in Mathematics of Communications. – 2009. – Vol. 3, No. 3, pp. 235-250.

[11]. Wichmann B. A Note on Restricted Difference Bases // Journal of the London Mathematical Society. – 1963. – Vol. 38, pp. 465-466.

[12]. Пазюк О.В., Цимбал Ю.В. Система розподілених обчислень для пошуку оптимальних числових лінійок // Комп'ютерні науки та інформаційні технології: Матеріали 4-ї МНТК CSIT-2009. – Львів: Вид-во ПП “Вежа і КО”, 2009. – с. 214-216.

[13]. Цимбал Ю.В., Пазюк О.В. Пошук оптимальних комбінаторних структур методом розподілених обчислень // Комп'ютерні науки та інформаційні технології: Вісник Національного університету “Львівська політехніка”, № 694. – Львів: Вид-во Львівської політехніки, 2011. – с. 62-66.

[14] Різник В.В., Цимбал Ю.В., Пазюк О.В., Коліда В.С., Олійник Д.В. Інтелектуальна система розподілених обчислень для пошуку оптимальних комбінаторних структур // Обчислювальний інтелект (результати, проблеми, перспективи): Матеріали 1-ї МНТК ОІ-2011. – Черкаси: Вид-во Черкаського державного технологічного університету, 2011. – с. 214-216.

[15] Цимбал Ю., Пазюк О., Коліда В., Олійник Д. Паралельний генетичний алгоритм для знаходження близьких до оптимальних дерев Ліча // Технічні вісті, № 2012/1(35), 2(36). – с. 83-85.

[16] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines // Journal of Chemical Physics. – 1953. – Vol. 21, pp. 1087-1092.

[17] D.S. Johnson, C.R. Aragon, L.A. McGeoch, C. Schevon. Optimization by Simulated Annealing: an Experimental Evaluation: Part 2, graph-coloring and number partitioning // Operations Research, Vol. 39, No.3, May-June 1991, pp. 378-406.

[18]. Ruskey F. Combinatorial Generation, 2003. <http://www.1stworks.com/ref/RuskeyCombGen.pdf>.