УДК 519.163

Євтушенко О. Я.

МЕМЕТИЧНИЙ АЛГОРИТМ ДЛЯ ЕВКЛІДОВОЇ ЗАДАЧІ ШТЕЙНЕРА

Запропоновано евристику для евклідової задачі Штейнера на основі меметичного алгоритму. Описано основні кроки алгоритму з детальним поясненням ключових процедур.

Ключові слова: евклідова задача Штейнера, дерево Штейнера, меметичний алгоритм, мінімальне остове дерево, оптимізація.

Вступ

Евклідова задача Штейнера належить до класу комбінаторних задач мінімізації і є NP-складною. Ця задача полягає у знаходженні мінімального дерева Штейнера для заданої множини точок $t_1, t_2, ...t_n$ в евклідовому просторі.

Дерево з вершинами $t_1, t_2, ...t_n, s_1, s_2, ...s_k$ є деревом Штейнера на $t_1, t_2, ...t_n$, якщо: 1) дерево не має ребер, що перетинаються; 2) $deg(s_i) = 3, 1 \le i \le k$;

3) $deg(t_i) \le 3, 1 \le i \le n$;

4) $0 \le k \le n-2$.

При цьому точки $T = \{t_1, t_2, ...t_n\}$ називаються термінальними, а $S = \{s_1, s_2, ...s_k\}$ – точками Штейнера.

Дерево Штейнера на $T \cup S$ є мінімальним, якщо немає іншого дерева на $T \cup S'$ з меншою сумарною вагою ребер.

Вперше задача про мінімальне з'єднання трьох точок за допомогою додаткової точки на площині була поставлена Ферма та розв'язана геометрично Торрічеллі у XVII столітті. Тоді ж сформульовано основні геометричні властивості задачі. Сімпсон узагальнив цю задачу на k точок у просторі довільної розмірності R^n у XVIII столітті. У 1941 р. задача отримала назву «задача Штейнера», а в 1977 р. доведено її належність до класу NP.

Точні алгоритми для розв'язку евклідової задачі Штейнера запропоновано у другій половині XX ст. з експоненційним часом виконання. Протягом останніх десятиліть описані евристичні алгоритми на основі мінімальних остових дерев, жадібних алгоритмів, методу відпалу, генетичного алгоритму, наближення розв'язку задачі Штейнера на графах та інше.

Запропонуємо новий алгоритм для розв'язку евклідової задачі Штейнера на основі меметичного підходу.

Меметичний алгоритм

Схема меметичного алгоритму

Термін «меметичні алгоритми» введений в кінці 80-х років минулого століття для позначення сімейства метаевристик, орієнтованих на використання популяцій. Меметичний алгоритм використовує популяцію особин для пошуку субоптимальних рішень задачі за заданою функцією здоров'я для оцінки рішень. Дослідники використовували меметичні алгоритми для знаходження наближення NP-повних задач [6].

На вхід нашого меметичного алгоритму подається множина термінальних точок T.

Параметрами алгоритму є кількість ітерацій, розмір популяції, ймовірність використання оператора кооперації p_{cross} та мутації p_{mut}, пропорції використання множин точок ініціалізації особин.

На виході роботи маємо набір точок Штейнера S та ребра дерева на вершинах $T \cup S$.

Роботу алгоритму можна описати такими кроками:

1. ініціалізація популяції;
2. зведення особин популяції до дерева Штейнера;
3. поки не виконується умова зупинки (кількість ітерацій) :

3.1. для кожної особини в популяції:

3.1.1. якщо $\xi > p_{cross}$, то кооперація даної особини і випадково обраної з популяції;

3.1.2. якщо $\xi > p_{mut}$, то мутація особини;

3.1.3. зведення особини-результату 3.1.1–3.1.2 до дерева Штейнера;

3.1.4. застосування локального пошуку до особини для покращення рішення;

3.1.5. додати особину до популяції;

3.2. зменшити популяцію до початкового розміру, залишивши лише найкращі особини;

3.3. якщо алгоритм не генерує кращих особин (популяція незмінна), додати невелику кількість нових, згенерованих особин.

У наступних підрозділах уточнимо основні складові алгоритму.

Попередня обробка

Вхідні дані евклідової задачі Штейнера є набором термінальних точок. У загальному випадку задача визначається на багатовимірному просторі, але в межах цієї роботи для спрощення розрахунків розглянуто двовимірну задачу з точками (x,y). Оскільки для задачі Штейнера не важливо, де саме на площині знаходяться ці точки, лише їх відносне розташування, можна обмежуватись точками у першому квадранті, тобто з невід'ємними координатами.

На кроці попередньої обробки термінальні точки варто впорядкувати за певним правилом для легшого відслідковування зв'язків між термінальними точками і точками Штейнера надалі. Найбільш очевидний шлях – лексикографічне впорядкування вхідних точок, яке здійснюється протягом часу $O(n\log n)$. Попередня обробка виконується лише один раз. Після неї множину термінальних точок можна пронумерувати $t_1,t_2,\ldots t_n$.

Кодування

Враховуючи, що розв'язання нашої задачі полягає в основному в знаходженні точок Штейнера, тоді як термінальні точки мають сталу кількість і розташування, для представлення розв'язку у вигляді особини в популяції достатньо знати місцезнаходження точок Штейнера. Отже, кодування може бути представлено у вигляді набору пар координат (x_i,y_i). Зазначимо, що важливою є фіксація, які саме точки утворюють з'єднання. Справді, одні й ті самі точки результуватимуть у різній сумарній довжині за різного з'єднання (рис. 1).

Рис. 1. Приклади різного з'єднання одного набору точок

Тому кодування здійснюється зі збереженням інформації про з'єднання між точками. Використовуючи властивість точок Штейнера (степінь точки завжди дорівнює три [5]), представимо точку так:

$\{i,(x_i,y_i),(t\vee s,t\vee s,t\vee s),(j_1,j_2,j_3)\}$,

де i – порядковий номер точки, (x_i,y_i) – невід'ємні координати в евклідовому просторі, j_l і $t\vee s$ – порядковий номер і тип (t – термінальна, s – точка Штейнера) точки, з якою зв'язана дана точка.

Оцінка верхнього обмеження на довжину рішення випливає з властивості дерева Штейнера (точок Штейнера має бути не більше *n-2*). Оцінки нижньої межі немає. Тоді порядковий номер належить інтервалу $[1,n-2]$, а особина матиме змінну довжину.

Також особина має містити інформацію про (t,t) ребра, що з'єднують дві термінальні точки, якщо вони присутні в дереві.

Оцінка рішення

Оскільки дерево Штейнера є остовим деревом, то для оцінки рішення є очевидним обрати сумарну довжину ребер. Позначимо одну особину-рішення Chromosome, а процедуру підрахунку евклідової відстані між двома точками euclidean_ distance. Тоді оцінку рішення рахуємо так:

sum ← 0
for $\{i,(x_i,y_i),(t\vee s,t\vee s,t\vee s),(j_1,j_2,j_3)\}$ in Chromosome do
for *l* in 1..3 do
if $mun_l = t$ then
sum ← *sum* + euclidean_distance ((x_i,y_i), j_l in *T*)
else if $j > i$ then
sum ← *sum* + euclidean_distance ((x_i,y_i), j_l in Chromosome)
коментарій: інакше ($j < i$) це ребро вже рахували
endfor
endfor
foreach (t_i,t_j) in Chromosome do
sum ← *sum* + euclidean_distance (*i* in *T*, *j* in *T*)
endforeach

Обчислення функції здоров'я залежить від кількості ребер остового дерева, або ж від кількості точок. Оскільки кількість точок не перевищує *2n-2*, маємо часову оцінку $O(n)$.

Початкова ініціалізація

Одна з характеристик евклідової задачі Штейнера полягає в тому, що всі точки знаходитимуться у межах опуклої оболонки, побудованої на цих точках [5]. Таким чином для початкової ініціалізації можна використовувати будь-які випадкові точки з опуклої оболонки або ж рахувати точки за допомогою додаткових методів, наприклад триангуляції Делоне (рис. 2).

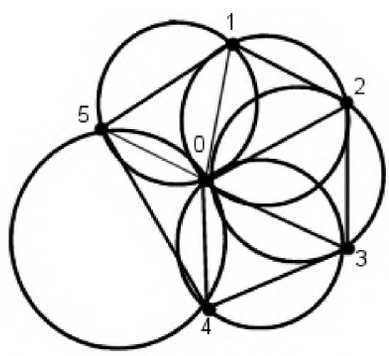

Рис. 2. Ілюстрація методу триангуляції Делоне

Триангуляція Делоне з'єднує відрізками множину точок у трикутники так, що описані кола навколо цих трикутників не містять всередині точок цієї множини. Результатом триангуляції є набір трикутників, причому кількість ребер буде рівна кількості точок. Алгоритм для цього методу виконується за *O (nlogn)*. Для початкової ініціалізації можна взяти точки на трикутниках, координати яких легко порахувати.

Кількість трикутників і ребер в триангуляції залежить від загальної кількості точок *n* та кількості точок *chp* ($chp \leqslant n$), що утворюють опуклу оболонку. Наведемо оцінку кількості основних геометричних складових триангуляції: $2n-2-chp$ трикутників, $3n-3-chp$ ребер, *chp* ребер опуклої оболонки і $3n-3-2chp$ внутрішніх ребер.

Оскільки s_i для будь-яких трьох точок буде знаходитись всередині трикутника на цих точках, то в роботах [1; 7], як додаткові точки, використано центроїди трикутників – точки перетину медіан.

Координати центроїда визначаються так:

$$x=\frac{1}{3}(x_a+x_b+x_c),\ y=\frac{1}{3}(y_a+y_b+y_c)$$

Кожен трикутник має лише один центроїд, тобто всього центроїдів $2n-2-chp$. Враховуючи обмеження на *chp*, ця величина лежатиме у межах $[2n-5,n-2]$.

Як додаткові точки, також можна використати середини сторін трикутників, які знаходяться всередині опуклої оболонки. В цьому випадку координати визначаються так:

$$x=\frac{x_a+x_b}{2},\ y=\frac{y_a+y_b}{2}.$$

Точка середини внутрішніх ребер одна для кожної пари таких трикутників, їх кількість дорівнює кількості внутрішніх ребер – $3n-3-2chp$. Враховуючи обмеження на *chp*, ця величина лежатиме у межах $[3n-9,n-3]$.

Маємо три множини точок: точки-центроїди, точки-середини внутрішніх ребер та випадкові точки з опуклої оболонки. Для створення диверсифікованої початкової популяції обираємо точки з однієї з цих множин у визначеній вхідним параметром пропорції, доки довжина хромосоми не стане *n-2*.

Процедура зведення до дерева Штейнера

Кожну особину у популяції слід звести до дерева Штейнера. Для цього застосовуємо таку процедуру.

Особина має *n-2* точок Штейнера та n термінальних. На них будуємо повний граф, ваги ребер якого є довжинами шляхів між відповідними точками-вершинами. На цьому графі знаходимо мінімальне остове дерево за допомогою відомого алгоритму Крускала, складність якого $O(n\log n)$.

Далі перевіряємо кожну s_i :

1. якщо s_i має степінь 1, видалити її разом з відповідним ребром;
2. якщо s_i має степінь 2, видалити її та інцидентні їй ребра, з'єднавши її сусідів одним ребром;
3. якщо s_i має степінь 3, залишити її та інцидентні ребра;
4. якщо s_i має степінь 4:

4.1 визначаємо дві множини інцидентних вершин залежно від того, в якій півплощині відносно *x*-координати s_i знаходиться вершина;

4.2 якщо в кожній множині по 2 вершини, видаляємо s_i , додаємо дві нові точки Штейнера

$$v=\left(x_{s_i}-\frac{x_{s_i}-x_{min}}{2},y_{s_i}-\frac{y_{s_i}-y_{min}}{2}\right)\text{ та}$$

$$u=\left(x_{s_i}+\frac{x_{max}-x_{s_i}}{2},y_{s_i}+\frac{y_{max}-y_{s_i}}{2}\right)$$

зі степенем три, де x_{s_i} , y_{s_i} – координати s_i , x_{min} , y_{min} – найменші значення координат точок у лівій півплощині, x_{max} , y_{max} – найбільші значення координат точок у правій півплощині (рис. 3) ;

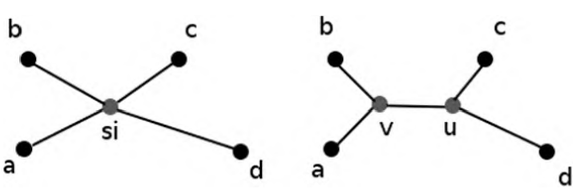

Рис. 3. Ілюстрація обробки точки Штейнера зі степенем чотири, коли визначені у п. п. 4.1 множини інцидентних вершин містять по дві вершини

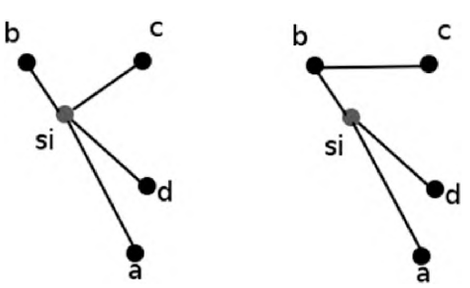

Рис. 4. Ілюстрація обробки точки Штейнера зі степенем чотири, коли визначені у п. п. 4.1 множини інцидентних вершин містять по одній і три вершини з ребром (s_i,c) більшим, ніж відстань від c до b

4.3 якщо в першій множині одна точка b, а в другій три a,c,d, то визначимо, яка з цих трьох точок з'єднана з s_i найменшим ребром (s_i,c). Якщо (s_i,c) більше, ніж відстань від c до b, то видаляємо (s_i,c) і додаємо (b,c) (рис. 4). Інакше залишаємо ребро (s_i,c), рахуємо точку Торрічеллі [8] для s_i та a,d, яка буде новою точкою Штейнера v з ребрами $(s_i,v),(a,v),(d,v)$ (рис. 5).

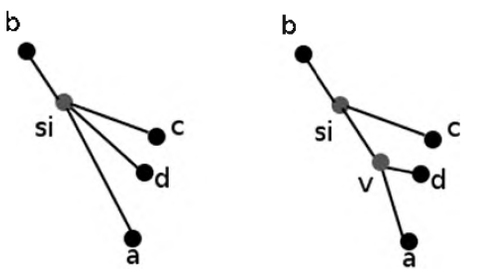

Рис. 5. Ілюстрація обробки точки Штейнера зі степенем чотири, коли визначені у п. п. 4.1 множини інцидентних вершин містять по одній і три вершини з ребром (s_i,c) меншим, ніж відстань від c до b

Зазначимо, що дві множини, утворені у п. п. 4.1 процедури зведення, завжди будуть непорожні, оскільки всі точки знаходяться у межах опуклої оболонки на T. Доведення від супротивного: нехай усі чотири точки знаходяться з одного боку від s_i. Оскільки всі точки особини-рішення належать опуклій оболонці на T, то з другого боку s_i також мають бути термінальні точки, а якщо вони є, то алгоритм мінімального остового дерева мав знайти інше мінімальне дерево за їхньої участі. Тому обидві множини, утворені півплощинами, будуть непорожні.

У межах меметичного алгоритму процедура зведення відіграє роль локального пошуку в ініціалізації. Завдяки застосуванню процедури алгоритм починає роботу не з випадковою популяцією, представники якої не обов'язково є деревом Штейнера, а вже з покращеними рішеннями-деревами.

Складність процедури зведення: *O (n)*.

Генетичні оператори

Кооперація двох особин є головним генетичним операторам для еволюції; також її називають оператором рекомбінації.

Для застосування оператора рекомбінації обираємо дві особини, що є представленнями двох можливих рішень. Ці особини поєднуємо одна з одною, використовуючи пошук мінімального остового дерева та утворюючи нове рішення. Оскільки обидві особини є мінімальними остовими деревами на $T \cup S$, то пошук остового дерева на об'єднанні їх ребер із застосуванням процедури зведення до дерева Штейнера результуватиме в іншому коректному рішенні задачі. Результатом кооперації може бути особина довжиною більше, ніж *n-2*, але завдяки відбору лише найкращих представників для наступного покоління (крок 3.2 алгоритму) такі рішення відсіюються.

Іншим генетичним оператором є мутація, що полягає у випадковому виборі точки Штейнера і пересуненні її у випадкове місце в межах опуклої оболонки. Частота застосування такого оператора може сильно впливати на різноманіття згенерованих рішень, адже він змінює топологію дерева Штейнера.

Частота застосування генетичних операторів регулюється вхідними параметрами алгоритму.

Локальний пошук для покращення рішення

Оскільки пошук ведеться в евклідовому просторі, використовуємо невеликі зміни для випадково обраних точок Штейнера:

1. пересунення точки Штейнера на маленьку відстань у випадковому напрямку;
2. зсув точки Штейнера від чи до сусідньої для усереднення довжин ребер, інцидентних даній точці;
3. зсув точки Штейнера до термінальної.

Який з трьох варіантів покращення застосовувати обираємо випадковим чином.

На противагу генетичним операторам локальний пошук виконуємо на кожній ітерації для локального уточнення місцезнаходження точок Штейнера, але зміни приймаються лише тоді, коли сумарна довжина ребер, інцидентних даній точці, не стає більшою.

Реалізація

Для реалізації та візуалізації використано проекти з відкритим кодом CGAL [3], Gnuplot [4].

Представлення порядкового номера і для термінальних точок, і для точок Штейнера як unsigned int, що має інтервал значень $\left[0;2^{32}-1\right]$, є достатнім для експериментального дослідження.

Приклад роботи алгоритму на 10 точках, 8 ітераціях та з популяцією в 5 представників представлено на рис. 6.

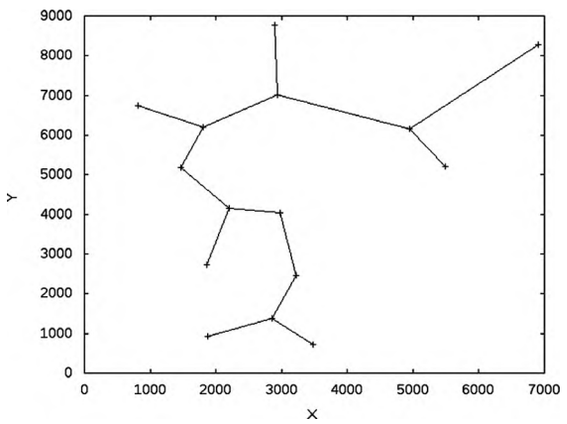

Рис. 6. Приклад роботи алгоритму на 10 точках

Висновки

Запропонований алгоритм поєднує кілька різних підходів: комбінування різноманітних точок (випадкових і отриманих з триангуляції Делоне) під час генерації популяції та контроль за перебігом еволюції через додавання нових особин забезпечує диверсифікацію дослідження пошукового простору, застосування локального пошуку на кожному кроці – інтенсифікацію отриманого рішення. Час роботи алгоритму в найкращому випадку обмежується $O(nlogn)$, і показав непогані результати на тестах з [2]. Просторова складність алгоритму $O(n^2)$.

Надалі слід оптимізовувати структури даних для збереження дерев і процедури.

Список літератури

1. Beasley J. E. A Delaunay triangulation-based heuristic for the Euclidean Steiner problem / J. E. Beasley, F. Goffinet // Networks. – 1994. – Vol. 24, № 4. – P. 215–224.
2. Beasley J. E. Operations Research library [Електронний ресурс] / J. E. Beasley. – Режиму доступу: http://people.brunel.ac.uk/~mastjjb/jeb/info.html. – Назва з екрана.
3. Computational Geometry Algorithms Library [Електронний ресурс] / CGAL Open Source Project. – Режим доступу: http://www.cgal.org/. – Назва з екрана.
4. Gnuplot library [Електронний ресурс]. – Режим доступу: http://www.gnuplot.info/. – Назва з екрана.
5. The Steiner Tree Problem / F. Hwang, D. S. Richards, P. Winter. – Elsevier Science Publishers B. V. – 1992. – P. 338. – (Annals of Discrete Mathematics, Vol. 53).
6. Handbook of Memetic Algorithms / F. Neri, C. Cotta, P. Moscato. – Springer-Verlag Berlin Heidelberg, 2012. – P. 368. (Studies in Computational Intelligence, Volume 379).
7. Van Laarhoven J. W. Exact and heuristic algorithms for the Euclidean Steiner tree problem: PhD diss. / Jon William Van Laarhoven ; University of Iowa, 2010.
8. Weisstein Eric W. Fermat Points / Eric W. Weisstein // MathWorld – Wolfram Web Resource. – Режим доступу: http://mathworld.wolfram.com/FermatPoints.html. – Назва з екрана.

O. Yevtushenko

MEMETIC ALGORITHM FOR EUCLIDEAN STEINER PROBLEM

The paper proposes heuristic for Euclidean Steiner problem based on memetic algorithm. Main steps of the algorithm are described and crucial procedures are explained.

Keywords: Euclidean Steiner problem, Steiner tree, memetic algorithm, minimal spanning tree, optimization.

Матеріал надійшов 13.09.2013