Компьютерная математика

УДК 519.711.3, 519.7:004.8

М.І. ОГУРЦОВ, О.М. ХОДЗІНСЬКИЙ

РОЗРОБКА АЛГОРИТМІВ РОЗВ'ЯЗАННЯ ЗАДАЧІ МАРШРУТИЗАЦІЇ ТРАНСПОРТНИХ ЗАСОБІВ З ЧАСОВИМИ ВІКНАМИ

Теория и методы оптимизации

Виконано формалізацію задачі маршрутизації транспортних засобів із часовими вікнами та зведено її до оптимізаційної задачі на просторі перестановок. Розроблені алгоритми розв'язання задачі шляхом побудови початкового розв'язку послідовним алгоритмом та його подальшої оптимізації методом локального пошуку. Наведено результати обчислювального експерименту по визначенню ефективності розроблених алгоритмів.

Вступ. Зростання обчислювальних потужностей сучасних комп'ютерів дозволяє прискорити вирішення багатьох складних проблем, що виникають у різних прикладних областях. Однак математичні моделі значної кількості цих проблем відноситься до класу NP-важких задач [1]. Це означає, що обчислювальна складність задачі залежить від розміру вхідних даних експоненційно. У таких випадках актуальною є розробка нових методів і алгоритмів розв'язування таких задач. Однією з таких задач є задача маршрутизації транспортних засобів при наявності часових вікон [2].

Опис проблеми. Задача маршрутизації транспорту (Vehicle Routing Problems, VRP) полягає у виявленні та оптимізації маршрутів до заданих точок-споживачів для транспортних засобів (ТЗ), що знаходяться в депо [3]. Предметом розгляду в цій статті є маршрутизація ТЗ із часовими вікнами. Маршрутизація транспорту відноситься до комбінаторних задач, які можна подавати як оптимізаційні задачі на графі, де під вершинами графа розуміються споживачі, а під зваженими ребрами – шляхи від одного споживача до іншого. Для кожного споживача задано кількість товарів, які йому треба доставити, та проміжок часу – часове вікно, в середині якого треба цю доставку виконати. Задано також загальну кількість ТЗ та вантажопідйомність кожного з них.

Розв'язком задачі є розбиття множини замовлень по ТЗ та сукупність їх маршрутів. Кожен такий розв'язок характеризується вартістю, яка залежить від витрат на здійснення перевезень та можливих штрафів за невкладання у часові вікна.

Маршрутизація ТЗ із часовими вікнами – це актуальна NP-важка наукова задача [3 – 4].

Мета дослідження. Таким чином, з урахуванням вищезазначеного, метою дослідження є створення математичної моделі формалізація задачі маршрутизації ТЗ із часовими вікнами та розробка методів і алгоритмів її розв'язання для того, щоб мінімізувати значення результуючої цільової функції шляхом мінімізації загальної кількості ТЗ, загального шляху всіх ТЗ і загального часу очікування, необхідних для обробки запитів клієнтів у призначені інтервали часу. Особливістю є те, що ТЗ, який прибув раніше нижньої часової межі, чекає її настання.

Як додаткову мету слід виділити потребу максимально завантажити перші ТЗ.

Задача упаковки у транспортній задачі, що розглядається, не стоїть (вантаж, що буде доставлятись у рамках цієї задачі – дрібнорозмірний).

Основна частина.

Введемо наступні позначення:

$G(V, E)$ – граф;

$V = \{v_0, v_1 ..., v_n\}$ – множина вершин (v_0 – депо, $v_1 ... v_n$ – споживачі);

E – множина ребер $\{(v_i, v_j) \mid i \neq j\}$;

C – матриця невід'ємних відстаней (вартості шляху) c_{ij} між споживачами;

m – кількість ТЗ;

R_i – маршрут i-го ТЗ ($i = 1..m$);

R – сукупність маршрутів R_i ($i = 1..m$);

$C(R_i)$ – вартість маршруту R_i;

q_i – об'єм вантажу, що поставляється i-му споживачеві.

З кожною вершиною V_i асоційована деяка кількість товарів, що мають бути доставлені відповідному споживачеві. Завдання маршрутизації полягає у визначенні такої множини маршрутів m з мінімальною загальною вартістю, щоб кожна вершина множини V обслуговувалась тільки одним ТЗ і тільки один раз. Крім того, всі маршрути мають починатися і закінчуватися в депо (v_0).

Розв'язком задачі є розбиття множини V на підмножини (клієнти одного ТЗ) та виконання порядку обходу на кожній підмножині (перестановка вершин маршруту) [3].

Цільовою функцією в загальному випадку є вартість розв'язку задачі:

$$F_{VRP} = \div \sum C(R_i), \tag{1}$$

де $C(R_i)$ – сума довжин ребер маршруту R_i, $i = 1,...,m$.

У класичному варіанті без часових вікон потрібно знайти допустимий розв'язок з мінімальною вартістю. Також у даній роботі вводяться наступні обмеження для умов задачі:

– обсяг вантажів на кожному маршруті R_i не має перевищувати заданої величини вантажопідйомності Q;

– маршрутизація часовими вікнами (VRP with Time Windows, VRPTW). Для виконання запиту кожного клієнта v_i задано проміжок часу, визначений як інтервал $[e_i, l_i]$ – заданий горизонт (scheduling horizon) [4]. Тобто реальний момент виконання замовлення має бути всередині інтервалу.

Дані кожного споживача включають розміщення його вершини графу, об'єму вантажу q_i, інтервалу $[e_i, l_i]$, та час, що буде затрачено на обслуговування даного клієнта. Набори вхідних даних були взяті із загальновідомої бібліотеки задач маршрутизації ТЗ із часовими вікнами [4]. Зона доставки може бути подана у вигляді двовимірної координатної площини зі значеннями по осях x та y. Визначення відстані між точками для доставки відбувається за стандартною формулою для прямокутної системи координат:

$$d = \sqrt{(x_2 - x_1)^2} + (y_2 - y_1)^2. \qquad (2)$$

Розглянемо запропонований **алгоритм розв'язання задачі маршрутизації ТЗ із часовими вікнами**. Шукати розв'язок поставленої задачі маршрутизації ТЗ із часовими вікнами пропонується в 2 етапи.

Розроблений алгоритм розв'язання задачі маршрутизації ТЗ із часовими вікнами у вигляді псевдокоду показаний на рис. 1.

```
Best_Solution = 0;
Read_Input (Вхідні дані клієнтів);
Sorted_Data = Sorting_Input(Вхідні дані клієнтів);
Solution = DeterminedConstruction (Sorted_Data);
UpdateSolution (Solution, Best_Solution);
for k = 1,2,..., MAX_ITERATIONS do
   Solution = LocalSearch (Solution);
   if (Solution is better than Best_Solution) then
      UpdateSolution (Solution, Best_Solution);
   endif;
endfor;
```

РИС. 1

Для розв'язання поставленої задачі та мінімізації цільової функції (1) слід на першому етапі побудувати для подальшої оптимізації початковий набір маршрутів, що включатиме маршрут для кожного ТЗ. При цьому пропонується використовувати паралельні обчислення для одночасної побудови сімейства початкових наближених розв'язків у режимі мультистарт. Маршрути з цього сімейства можна представити у наступному вигляді:

$$V_1 = \{\{v_{11} \ldots v_{IN} \ldots\}, \{v_{21} \ldots v_{2D}\}, \ldots, \{v_{m1} \ldots v_{mK}\}\}. \qquad (3)$$

Кожен з векторів відноситься до окремого ТЗ. Разом вони формують загальний вектор маршрутів, що підлягає обрахунку цільової функції та оптимізації.

Заповнення вектора маршрутів пропонується виконувати розробленим **методом детермінованої побудови початкового маршруту**. Згідно нього зі списку клієнтів, відсортованого певним чином, клієнти послідовно додаються до маршрутів ТЗ.

Список клієнтів формується одним з трьох шляхів:

а) випадково;

б) сортуванням за початком часових вікон;

в) сортуванням за обсягом вантажу.

В першу чергу ми намагаємось додати клієнта до першого ТЗ. Якщо після цього маршрут є допустимим (ми не вийшли за межі граничних обсягів вантажу для доставки та всі клієнти з маршруту будуть відвідані вчасно), то залишаємо даного клієнта в цьому ТЗ, якщо ж маршрут стає недопустимим – намагаємось додати клієнта до наступного ТЗ. Якщо всі ТЗ перебрані й до жодного з них ми не можемо додати даного клієнта, додаємо новий ТЗ, якому призначаємо відвідати даного клієнта. Порядок додавання клієнтів до ТЗ показаний на рис. 2.

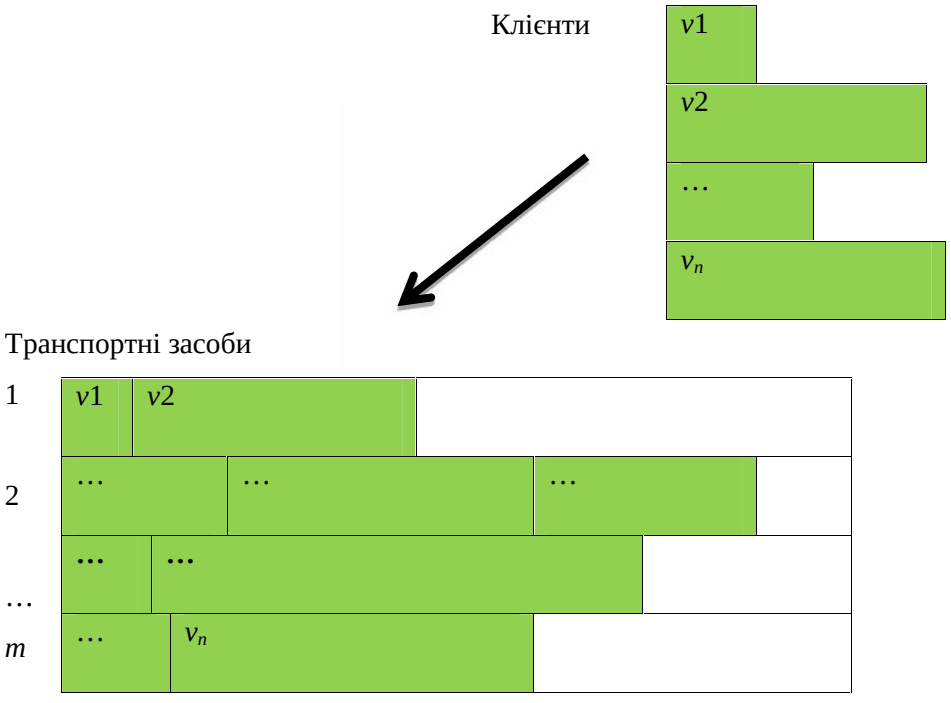

РИС. 2

При цьому окремо розглядався наступний варіант: на кожному кроці при додаванні нового клієнта до ТЗ застосовувався розроблений **метод жадібного сортування частково сформованого маршруту**. Тобто після кожного додавання клієнта ми розв'язували задачу комівояжера із часовими вікнами для окремого ТЗ та підмножини клієнтів, доданих до нього.

Для кожного елемента маршруту (3) можна розрахувати його вартість:

$$C(R_i) = \begin{cases} d, T_c \geq T_0 \\ d + (T_0 - T_c), T_c < T_0 \end{cases}, \quad (4)$$

де T_0 – значення початку часового вікна для даного клієнта; T_c – поточний час на момент початку етапу маршруту.

Сума вартостей відповідно до формули (1) складе потрібну нам цільову функцію. Вид формули пов'язаний з тим, що згідно Solomon [4] час подорожі між клієнтами прийнятий таким, що дорівнює відстані між ними.

На другому етапі роботи розробленого алгоритму маршрутизації ТЗ із часовими вікнами виконується перевірка оптимальності отриманого початкового наближення та його оптимізація на кожному процесорі окремо. Результат, отриманий на етапі 1, має форму вектора (3), а на ньому можна виконувати ітеративний оптимізаційний процес, шукаючи кращі розв'язки в околі шляхом застосування локального пошуку. В результаті ми розвиваємо проблему, що розглядається, одразу у двох напрямах – це й задача маршрутизації транспорту з часовими вікнами і класична оптимізаційна задача на просторі перестановок. Для класичної задачі відомі ефективні та швидкі алгоритми пошуку наближених та оптимальних розв'язків, що дозволить значно спростити розв'язок VRPTW.

Виконана формалізація задачі маршрутизації ТЗ із часовими вікнами була застосована для побудови алгоритму розв'язання відомих задач [4] на практиці. Як описано вище, на першому етапі для побудови початкового маршруту був застосований детермінований алгоритм. На другому етапі в рамках виконання роботи було розроблено та впроваджено до програмної реалізації задачі маршрутизації ТЗ за наявності часових вікон метод оптимізації наближеного розв'язку, отриманого на етапі 1 шляхом застосування локального пошуку. Для цього множину маршрутів наближеного розв'язку, отриману на етапі 1 ми представляємо у вигляді вектора, шукаючи кращі розв'язки в околі **методом локального пошуку**. Формалізовано цей варіант оптимізації можна представити як локальний пошук шляхом простої перестановки випадкового елемента маршруту (4), що не є депо, як з елементами маршрутів інших ТЗ, так і з фіктивним клієнтом – тобто просто перемістити цього клієнта до маршруту іншого ТЗ. В результаті ми отримуємо нові варіанти маршруту Y в околі поточного маршруту X такі що:

$$O(X) = \{Y \in P_n : \omega^{sj} X\}. \quad (5)$$

Всі ці розбиття утворюють окіл (диз'юнктний) з блоків розбиття.

Далі обчислювалась вартість нового маршруту за формулою (1) та за формулою (6) виконувалась перевірка, чи є новий варіант Y ближчим до оптимального ніж X:

$$\Delta = F_{Y-} F_X . \qquad (6)$$

Якщо $\Delta < 0$, то в цьому випадку відбувається міграція до нового маршруту. Пошук зупиниться, якщо буде досягнуто локального мінімуму.

При застосуванні методу локального пошуку була допустима міграція клієнтів між ТЗ (рис. 3), що може призвести до зміни кількості клієнтів у їх маршрутах та навіть до усунення одного чи декількох ТЗ з розв'язку.

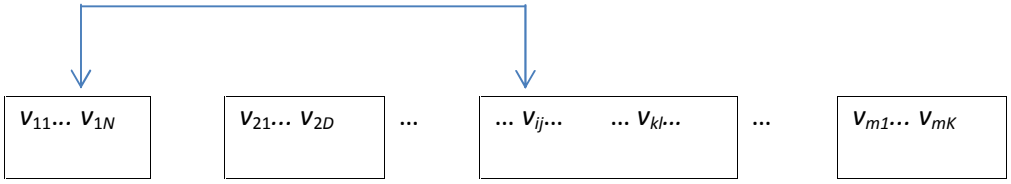

РИС. 3

Тестування виконувалось для розмірності 100 клієнтів, результати роботи алгоритму вже після етапу 1 були в деяких випадках усього лише менш ніж у два рази гіршими, ніж кращі відомі розв'язки. Після етапу 2 результати наблизились до оптимальних.

Як набір клієнтів із загальновідомої бібліотеки задач маршрутизації ТЗ із часовими вікнами [4] було обрано набір *C*2, розв'язок для якого слід шукати для перших 25, 50 або 100 клієнтів, яких треба відвідати. При цьому реалізація алгоритму на мові *C* була виконана на основі використання паралельних обчислень на кластері СКІТ (як на СКІТ-3, так і на СКІТ-4 в залежності від розмірності задачі) для одночасного пошуку декількох варіантів наближеного розв'язку в режимі мультистарт. Для перевірки роботи розробленого алгоритму було обрано тестову задачу Соломона *C*201 [4], для якої найкращим з відомих маршрутів, побудованих у світі для розмірності 100 є варіант з кількістю ТЗ – 3, загальною відстанню, яку вони долатимуть – 589.1.

При початковому сортуванні клієнтів за обсягом вантажу, що необхідно їм доставити, кількість ТЗ після виконання етапу 1 склала 18, загальна відстань – 4323.52, час виконання – в середньому 2 сек. При додаванні жадібного сортування на кожному кроці формування маршруту (розв'язанні задачі комівояжера для кожного ТЗ) кількість ТЗ скоротилася до 6, загальна відстань до – 2920.95, час виконання – в середньому 15 сек. Тобто ми отримали значне покращення (втричі за кількістю ТЗ, у півтора рази – за відстанню) ціною незначних часових затрат на обчислення. При початковому сортуванні клієнтів за початком часово-

го вікна ми отримали кількість ТЗ 6, загальна відстань склала 891.66, час виконання – в середньому 2 сек. При додаванні жадібного сортування на кожному кроці формування маршруту час виконання склав 21 сек., а відстань та кількість ТЗ не змінились.

При випадковому початковому сортуванні клієнтів загальна кількість ТЗ складала 15 – 24, загальна відстань перебувала в межах 3600 – 4600. Тобто вона була наближена до результатів сортування за обсягом вантажу. Час виконання операції коливався в районі 2 – 3 сек. При додаванні жадібного сортування на кожному кроці формування маршруту загальна кількість ТЗ завжди дорівнювала 6 – 7, загальна відстань значно коливалася, в середньому вона також була наближена до результатів сортування за обсягом вантажу після жадібного сортування і коливалася в межах 2510 – 3340 (приклади – у таблиці), час виконання операції коливався від 45 до 75 сек.

Метод локального пошуку показав результативну оптимізацію в усіх трьох варіантах початкового сортування. Для пошуку в околі 1 при початковому сортуванні клієнтів за обсягом вантажу відстань з 2920.95 скорочувалася до 2400.22, кількість ТЗ не змінилася, загальний час виконання алгоритму складав 50 – 70 сек. У процентному співвідношенні найкращий з отриманих на другому етапі результатів для випадкового сортування дорівнював 66,4 % від результатів першого етапу, тобто ми скоротили маршрут етапу 1 на 33,6 % – див. таблицю. При початковому сортуванні клієнтів за початком часового вікна відстань з 891.66 скорочувалася до 745.76; кількість ТЗ – до 5; загальний час виконання складав у середньому 55 – 60 сек. У процентному співвідношенні отриманий на другому етапі результат дорівнював 83,6 % від результатів першого етапу.

ТАБЛИЦЯ

Довжина маршруту після етапу 1	Довжина маршруту після етапу 2	Відсоток від початкового маршруту, %	Кількість транспортних засобів	Затрачений час, с
2516.88	2393.17	95,1	6	70
2770.34	2151.85	77,7	5	72
2880.90	2755.79	95,7	7	97
2923.31	2849.48	97,5	6	62
3007.45	1997.68	66,4	5	71
3134.32	3017.44	96,3	6	66
3220.65	2644.62	82,1	7	63
3242.71	3082.32	95,1	7	55

Висновки. В результаті виконаної формалізації задачі маршрутизації ТЗ із часовими вікнами її було зведено до класичної оптимізаційної задачі на просторі перестановок. Це дозволило значно спростити розв'язок задачі маршрутизації ТЗ із часовими вікнами. Аналіз числових експериментів на суперкомп'ютері СКІТ показав досить високу ефективність алгоритму за невеликої обчислювальної складності. Початкове сортування клієнтів за часом початку часового вікна дало істотний кількісний приріст отриманого наближеного розв'язку при незначних часових затратах, тому його рекомендується застосовувати для побудови початкового наближення. Практична реалізація показала високу ефективність, знаходячи досить гарні розв'язки для розмірності 100 клієнтів за час, що не перевищував 100 с.

В подальшому планується впровадити інші методи та алгоритми комбінаторної оптимізації на просторі перестановок, включаючи метод імітаційного відпалу, для порівняльного аналізу ефективності з уже впровадженими методами, й алгоритмами та пошуку кращих розв'язків VRPTW.

М.И. Огурцов, А.Н. Ходзинский

РАЗРАБОТКА АЛГОРИТМОВ РЕШЕНИЯ ЗАДАЧ МАРШРУТИЗАЦИИ ТРАНСПОРТНЫХ СРЕДСТВ С ВРЕМЕННЫМИ ОКНАМИ

Выполнена формализация задачи маршрутизации транспортных средств с временными окнами; задача была приведена к форме классической оптимизационной задачи на пространстве перестановок. Разработаны алгоритмы решения задачи маршрутизации транспортных средств с временными окнами путем детерминированного построения начального решения и его дальнейшей оптимизации методом локального поиска. Приведены результаты вычислительного эксперимента по определению эффективности разработанных алгоритмов.

M.I. Ogurtsov, A.N. Khodzinsky

DEVELOPMENT OF METHODS AND ALGORITHMS FOR SOLVING VEHICLE ROUTING PROBLEM WITH TIME WINDOWS

Formalization of vehicle routing problem with time windows is proposed; vehicle routing problem with time windows is transformed to the form of the classic optimization problem on the space of permutations. Methods and algorithms for solving vehicle routing problem with time windows based on determined construction of the initial solution and its further optimization by the local search method are developed. Results of computational experiment are provided to determine the effectiveness of the developed methods and algorithms.

1. *Leeand E.K.*, Mitchell J. E. Branch-and-bound methods for integer programming. Encyclopedia of Optimization. – Kluwer Academic Publishers, 1998.
2. *Гуляницкий Л.Ф., Самусь А.В.* Решение *H*-методом задачи оптимизации маршрутов транспортных средств с временными окнами // Компьютерная математика. – 2012. – № 2. – С. 147 – 155.
3. *Rahimi-Vahed, Alireza, Teodor Gabriel Crainic, Michel Gendreau, and Walter Rei.* A path relinking algorithm for a multi-depot periodic vehicle routing problem // Journal of Heuristics. – 2013. – 19(3). – P. 497 – 524.
4. *Solomon M., Marius M.* Algorithms for the vehicle routing and scheduling problems with time window constraints // Operations research. – 1987. – 35(2). – P. 254 – 265.

Одержано 06.11.2015

Про авторів:

Огурцов Максим Ігорович,
молодший науковий співробітник
Інституту кібернетики імені В.М. Глушкова НАН України,

Ходзінський Олександр Миколайович,
старший науковий співробітник
Інституту кібернетики імені В.М. Глушкова НАН України.