універсальніші інструменти та технології з можливістю гнучкої постановки задач [7–9]. Системи, створювані за такими технологіями, мають володіти великою інформаційною ємністю і завадостійкістю, потребувати малого часу на розроблення, мати здатність до застосування в різних галузях медицини та біології. Такими системами могли б бути системи, здатні налаштовуватися на розв'язання задач.

1. http://misbook.interin.ru/rus.pdf. 2. http://misbook.interin.ru/eng.pdf. 3. http://www.microsoft.com/Ukraine/Government/Health/Default.mspx. 4. http://www.likar-info.com. 5. http://www.med-tech.com.ua. 6. Батюк А.Є., Пасєка М.С., Шимбра Т.П. Розробка інформаційно-аналітичної системи для лікарні швидкої допомоги // Вісн. Нац. ун-ту "Львівська політехніка". – 1999. – № 380. – С. 61–68. 7. Батюк А.Є., Чоп'як В.В., Цмоць І.Г., Леськів М.В., Кирик В.Ю. Інформаційна система автоматизації діяльності медичного центру клітинної імунології та алергології // Моделювання та інформаційні технології: Зб. наук. праць ІПМЕ НАН України. – 2002. – Вип. 14. – С. 175–182. 8. Батюк А.Є. Автоматизація та інтелектуалізація інформаційної підтримки лікаря клініко-діагностичної лабораторії // Інформаційні технології та системи: Наук.-техн. журн. -2005. – Т. 8, № 1. – С. 11–16. 9. Батюк А.Є. Застосування інтелектуальних технологій при виявленні знань і прогнозуванні в медицині // Інформаційні технології та системи: Наук.-техн. журн. – 2005. – Т. 8, № 1. – С. 114–119.

УДК 621.382

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

АЛГОРИТМИ ДИНАМІЧНОГО ФОРМУВАННЯ МОДЕЛІ РОБОЧОГО ПОЛЯ ДЛЯ ЗАДАЧІ КОМІВОЯЖЕРА З КЛАСТЕРНИМ РОЗПОДІЛОМ ТОЧОК

Описано алгоритми формування математичної моделі робочого поля для задачі комівояжера з кластерним розподілом точок, що дають змогу істотно зменшити розмірність задачі. Кластери формуються з груп точок, які знаходяться в близькому околі. Модель робочого поля подають множиною сформованих кластерів.

The algorithms of forming the model of the area for clustered TSP, that allow substantially decrease the size of the problem, are described. Clusters are formed from the groups of neighboring points. The model of the area appears as a set of the clusters.

Вступ

Однією з базових задач комбінаторної оптимізації, що має широке прикладне застосування, є задача комівояжера. Характерною особливістю сьогодення є істотне зростання її розмірності (до мільйонів точок), а також наявність часткових задач зі специфічними властивостями. Це – динамічні транспортні задачі, особливістю яких є поява нових точок обслуговування в процесі реалізації заданого маршруту; системи за викликом (швидка допомога, кур'єрська, пожежна служби, таксі), системи з часовими вікнами, системи постачання та інші. Вони потребують розроблення спеціальних алгоритмів, які б забезпечували отримання якісних результатів в режимі реального часу. В зв'язку з цим актуальним є розроблення ефективних декомпозиційних алгоритмів для задачі комівояжера, які б забезпечили отримання якісних розв'язків для задач великих розмірностей.

Формулювання задачі

За класичним формулюванням заданою вважають множину N з n точок ($|N|=n$), які описані їхніми координатами (x_i, y_i). Необхідно найти маршрут S^*, що проходить по одному разу через кожну точку, довжина якого $L^*(S^*)$ є мінімальною:

$$L^*(S^*) = \Sigma_{ij}\ l_{ij}^* \rightarrow min\Sigma_{ij}\ l_{ij}^* \quad \forall\, l_{ij} \in l_{ij}',$$

де l_{ij}' – деяка з допустимих за заданими обмеженнями ділянка між двома суміжними точками i та j виділеного маршруту.

Розглядатиметься частковий випадок, для якого розподіл точок має групово-зосереджений характер та утворює множину з k кластерів $C = C_1,\ldots,C_k$. Кожен кластер $C_i \in C$ описується макромоделлю у вигляді однієї точки з умовним центром x_i, y_i. Кластери формуються за допомогою декомпозиційних алгоритмів, що повинно істотно зменшити час обчислень. Задача з n точками замінюється задачею з k точками. Зазвичай $k<<n$, що істотно спрощує задачу. Деякою вибраною базовою процедурою (алгоритмом) P_0 розв'язується класична задача комівояжера для кластерів, яка визначає послідовність обходу кластерів (макромаршрут). Отримується маршрут $S^*(C) = \cup\, S^*_{ij}$, де S^*_{ij} – ділянка циклу між виділеними суміжними кластерами C_i та C_j, який вважаємо оптимальним для цієї множини кластерів. Довжина цього маршруту $L^*(S^*,C) = \Sigma_{ij}\ l_{ij}^*(C)$, де $l_{ij}^*(C)$ – довжини окремих ділянок між умовними центрами пар для визначеного оптимального макромаршруту.

Декомпозиційні алгоритми для формування кластерів передбачають розбиття поля задачі на підмножини за заданими критеріями з метою прискорення пошуку кластерів замість розгляду цілої області загалом. Об'єднання локальних розв'язків у підмножинах становитиме загальний розв'язок задачі пошуку кластерів. Кожен кластер розглядають як окрему точку та розв'язують задачу комівояжера для цих точок. Утворена модель робочого поля складатиметься зі сформованих кластерів. На такій моделі визначається макромаршрут, що задає порядок обходу кластерів.

Декомпозиційні алгоритми для формування моделі робочого поля

Запропоновані декомпозиційні алгоритми формування моделі робочого поля містять декілька етапів:

1) розбиття повної моделі поля задачі на підмножини за заданими критеріями;
2) формування кластерів у підмножинах за заданими критеріями;
3) опис кластерів математичними моделями;
4) розв'язання класичної задачі комівояжера для визначених моделей кластерів. Окремо для задачі загалом визначають послідовність обходу кластерів – макромаршрут; а також для кожного кластера зокрема;
5) об'єднання часткових розв'язків, отриманих на попередньому етапі, в єдиний початковий розв'язок;
6) оптимізація початкового розв'язку.

Декомпозиція повної моделі поля задачі на частини необхідна для забезпечення швидкого пошуку кластерів. Обмеженням під час декомпозиції є максимальна кількість точок, що може бути у кожній підмножині. Поле задачі розбивають на підмножини, кількість точок у кожній з яких не повинна перевищувати задане число. На рис. 1 показано поле задачі з кластерним розподілом точок.

Алгоритм розбиття поля задачі на підмножини виконується так. Спочатку визначають мінімальну область, до якої потрапляють усі точки і циклічно виконують поділ області на дві, якщо кількість елементів у області перевищує задану. В процесі поділу для кожної підобласті формується інформація про її суміжні підобласті. На рис. 2 показано розбиття поля задачі на підмножини із заданою максимальною кількістю елементів у кожній з них – 10 (створено 44 підмножини з кількістю точок від 0 до 10).

Наступний етап – формування кластерів за цими критеріями. Критерієм формування кластерів є відстань від точки до кластера. Відстань від точки до кластера – це відстань до найближчої до неї точки кластера. Якщо ця відстань не перевищує задану, точка належить до кластера. Алгоритм формування кластерів використовує інформацію, отриману на попередньому етапі – послідовно розглядаються усі створені підмножини, в яких перевіряються відстані між певною точкою та всіма іншими точками із підмножини чи кластерами, утвореними раніше. Якщо відстань від точки до іншої точки чи кластера підмножини не перевищує задану, то дві точки об'єднуються в новий кластер або точку додають до існуючого кластера відповідно. Це повторюється доти, поки не буде розглянуто усі підмножини поля задачі. На рис. 3 показано створені кластери моделі робочого поля задачі (сформовано 10 кластерів для заданого цілого поля задачі розмірністю 200 точок).

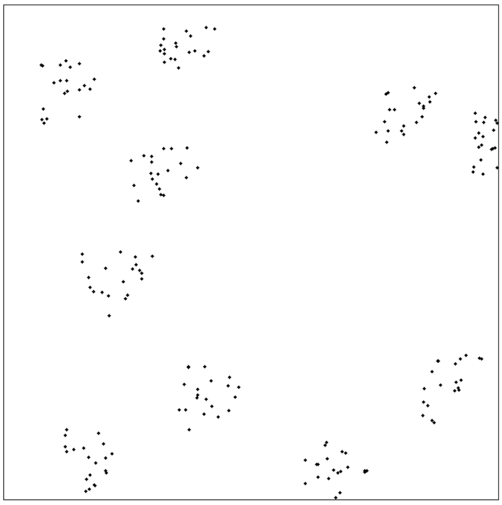

Рис. 1. Поле задачі з кластерним розподілом точок

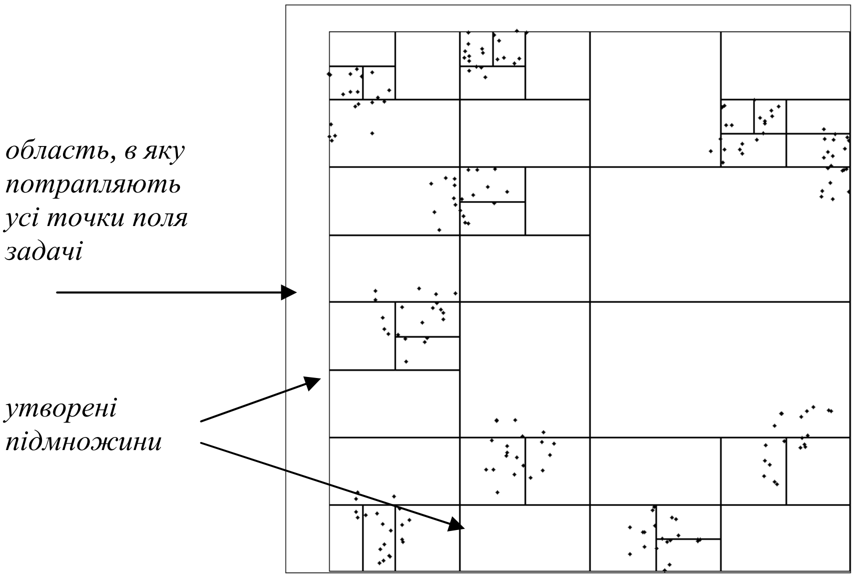

Рис. 2. Розбиття поля задачі на підмножини

Наступним етапом є формування математичних моделей кластерів. Можливими є два підходи, перший з яких передбачає утворення та використання на наступному етапі точкової математичної моделі для кожного кластера, другий – повної математичної моделі. За точкової моделі кластерів кожен кластер описується деяким умовним центром, наприклад "центром ваг", у другому випадку беруть до уваги усі точки кластера. В наших дослідженнях умовний центр кластера визначають як середнє значення координат всіх його точок – окремо для кожної з осей. Віддалі між кластерами визначають як відстані між умовними їхніми центрами. У випадку повних моделей кластерів віддалі між ними визначають як відстані між найближчими точками для кожної пари кластерів. У цьому випадку забезпечується більша точність визначення макромаршруту (це маршрут, що визначає послідовність обходу кластерів), проте цей підхід вимагає більших обчислювальних затрат.

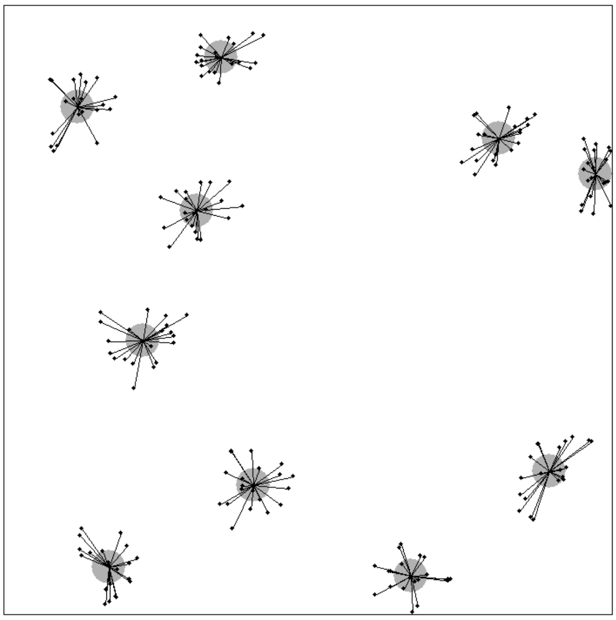

Рис. 3. Формування кластерів

Результати визначення макромаршрутів для двох запропонованих підходів можуть бути різними. Можливий випадок, коли за точкової математичної моделі кластерів для кластера C_2, наприклад, найближчим є кластер C_1, хоча реально ближчим є кластер C_3, якщо брати до уваги повну математичну модель (рис. 4). Відповідно макромаршрути в обох випадках можуть бути різними.

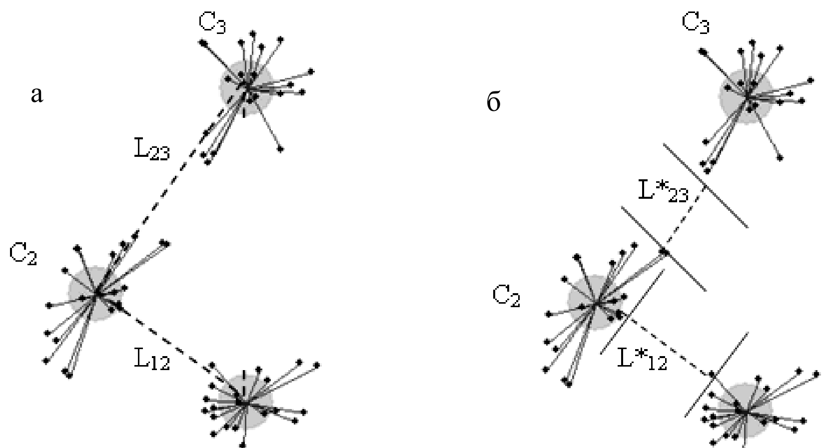

Рис. 4. Математична модель:
а – $L_{12} < L_{23}$ – для точкової математичної моделі;
б – $L^*_{12} > L^*_{23}$ – для повної математичної моделі

Останній етап формування моделі робочого поля задачі комівояжера – розв'язання класичної задачі комівояжера для кластерів. На цьому етапі кожен кластер розглядається як окрема точка. Класичну задачу комівояжера розв'язують за допомогою добре відомої та ефективної базової процедури (алгоритму). У нашому випадку це – 2-opt. Опишемо алгоритм 2-opt [1].

Алгоритм 2-opt – це добре відомий алгоритм для розв'язування задачі комівояжера, що ґрутується на локальній оптимізації попередньо знайденого маршруту. Обчислювальна складність алгоритму – $O(n^2)$ [2]. За цим алгоритмом початковий маршрут знаходять за допомогою будь-якого швидкого алгоритму (жадібного, алгоритму найближчого сусіда, алгоритму Борувки або ін. [2]), причому допустимим є навіть випадковий маршрут. Наступним етапом є його оптимізація за допомогою зміни двох існуючих ребер, що з'єднують точки маршруту, на два інші ребра, причому так, щоб отриманий маршрут став коротшим. Оптимізація триває, доки вона є можливою. На рис. 5 показано приклад такої заміни.

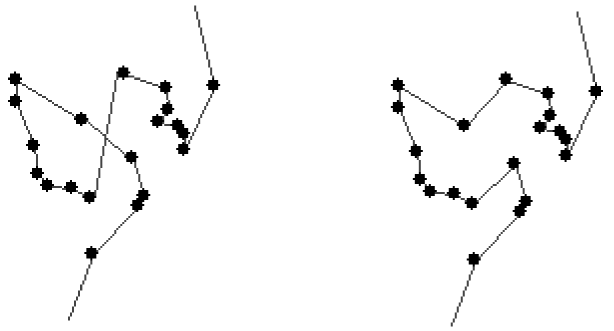

Рис. 5. Оптимізація маршруту за допомогою алгоритму 2-opt

Алгоритм 2-opt ґрунтується на відомому алгоритмі α-opt [3], де попередньо обчислений початковий маршрут оптимізують за допомогою зміни α існуючих ребер на α нових з меншою довжиною (існують алгоритми 3-opt, 4-opt, 5-opt [2]). Також необхідно згадати, що один з найкращих сьогодні алгоритм Ліна–Кернігана [2] та певні його модифікації [3] також базується на алгоритмі α-opt. В алгоритмі Ліна–Кернігана число α змінюється динамічно в процесі роботи алгоритму.

На рис. 6 показано розв'язок задачі комівояжера між кластерами. Множина кластерів та послідовність їхнього обходу формує модель для наступного кроку задачі комівояжера з кластерним розподілом точок.

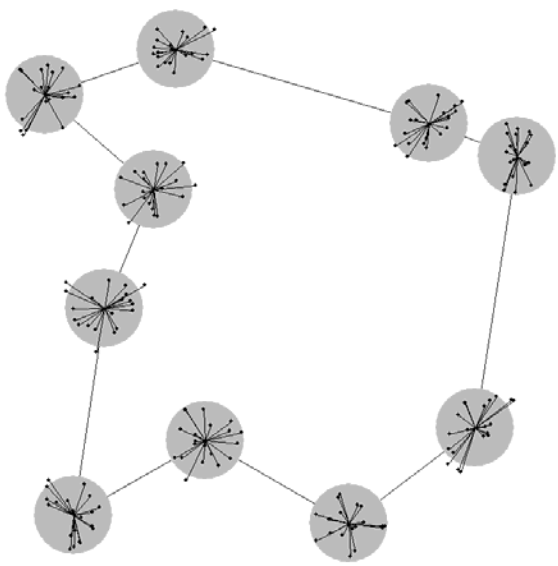

Рис. 6. Модель робочого поля для задачі комівояжера

У результаті описаних декомпозиційних алгоритмів сформовано кластери та послідовність їхнього обходу, що становить початковий етап розв'язування задачі комівояжера за кластерного розподілу точок.

Експерименти

Проведено тести для задач з різними розмірностями – 1000, 5000, 10000, 50000, 100000, 500000 та 1000000 точок. В усіх тестах згенеровані кластери з однаковою кількістю елементів – 100. Під час декомпозиції поля задачі на підмножини задавали максимальну кількість точок у підмножині – 100. Центри кластерів вибирали як середні значення координат точок кластера. Під час генерації кластерів задавали радіус кластера, тобто максимальну відстань точки кластера від його центру, а також мінімальну відстань між кластерами. Усі тести проводилися на ПК з процесором Pentium IV – 3000 МГц та об’ємом оперативної пам’яті 512 Мб. На рис. 7 показано графіки залежностей часу обчислень від розмірності задачі для етапів розбиття поля задачі на підмножини, формування кластерів та розв’язання класичної задачі комівояжера між кластерами алгоритмом 2-opt.

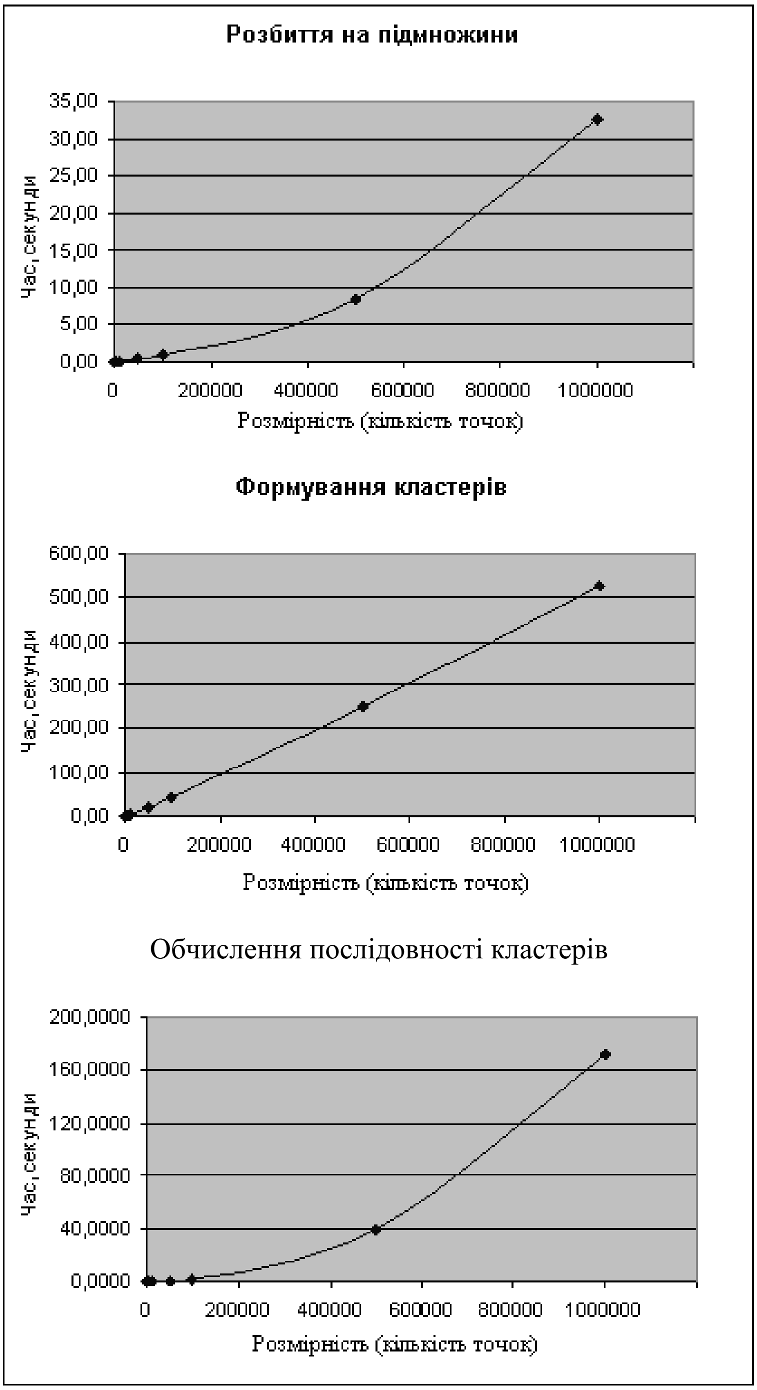

Рис. 7. Графіки залежностей часу обчислень від розмірності задачі для усіх етапів формування моделі робочого поля для задачі комівояжера

На рис. 8 показано графік залежності часу обчислень від розмірності задачі для усіх етапів формування робочого поля.

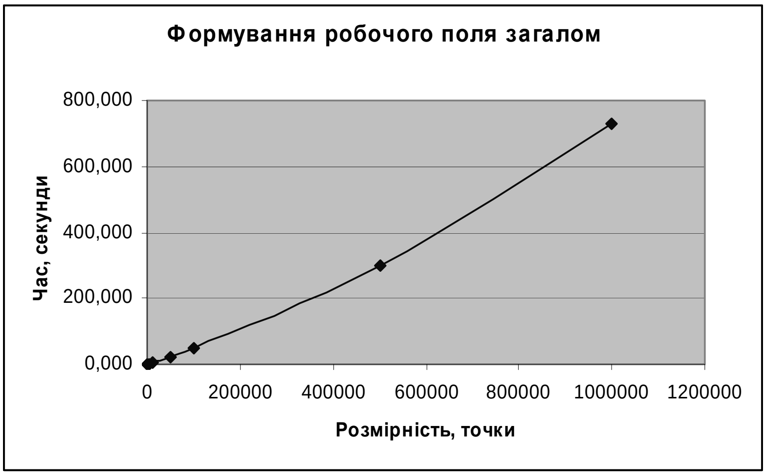

Рис. 8. Залежність часу від розмірності для усіх етапів разом

Результати усіх тестів наведено у таблиці.

Результати тестування задач розмірностями 1000–1000000 точок

Тест 1	*Розмірність - 1000*	
	Розбиття на підмножини, с	0,01
	Формування кластерів, с	0,32
	Обчислення послідовності обходу кластерів, с	0,0002
	Загальний час, с	**0,323**
Тест 2	*Розмірність - 5000*	
	Розбиття на підмножини, с	0,03
	Формування кластерів, с	1,83
	Обчислення послідовності обходу кластерів, с	0,002
	Загальний час, с	**1,865**
Тест 3	*Розмірність - 10000*	
	Розбиття на підмножини, с	0,07
	Формування кластерів, с	3,98
	Обчислення послідовності обходу кластерів, с	0,01
	Загальний час, с	**4,058**
Тест 4	*Розмірність - 50000*	
	Розбиття на підмножини, с	0,45
	Формування кластерів, с	21,80
	Обчислення послідовності обходу кластерів, с	0,32
	Загальний час, с	**22,570**
Тест 5	*Розмірність - 100000*	
	Розбиття на підмножини, с	1,04
	Формування кластерів, с	44,75
	Обчислення послідовності обходу кластерів, с	1,30
	Загальний час, с	**47,09**
Тест 6	*Розмірність - 500000*	
	Розбиття на підмножини, с	8,38
	Формування кластерів, с	249,44
	Обчислення послідовності обходу кластерів, с	38,80
	Загальний час, с	**296,62**
Тест 7	*Розмірність - 1000000*	
	Розбиття на підмножини, с	32,60
	Формування кластерів, с	524,98
	Обчислення послідовності обходу кластерів, с	171,20
	Загальний час, с	**728,78**

Висновки

Алгоритми формування робочого поля для задачі комівояжера з кластерним розподілом точок дають змогу істотно зменшити розмірність задачі за рахунок застосування кластерів як груп близько розміщених точок. Визначений порядок обходу кластерів призначений для подальшого розв'язання класичної задачі комівояжера всередині кожного кластера та об'єднання локальних розв'язків у загальний розв'язок. Як показали результати тестування, робоче поле формують за лінійний час, що вказує на доцільність використання запропонованих декомпозиційних алгоритмів для задачі комівояжера великих розмірностей – мільйонів точок.

1. Fredman M.L., Johnson D.S., McGeoch L.A., Ostheimer G. Data Structures for Traveling Salesmen // J. ALGORITHMS 18. – 1995. – P. 432–479. 2. Johnson D.S., McGeoch L.A. Experimental analysis of heuristics for the STSP. – 2002. 3. Helsgaun K. An effective implementation of the Lin-Kernighan Traveling Salesman Heuristic. – 2002.

УДК 621.382

Р. Базилевич, Р. Дюпа*, Р. Кутельмах
Національний університет "Львівська політехніка",
кафедра програмного забезпечення
*Університет Бордо (Франція)

ВИКОРИСТАННЯ АЛГОРИТМІВ ЛОКАЛЬНОЇ ОПТИМІЗАЦІЇ ДЛЯ РОЗВ'ЯЗУВАННЯ ЗАДАЧІ КОМІВОЯЖЕРА З КЛАСТЕРНИМ РОЗПОДІЛОМ ТОЧОК

Описано алгоритми локальної оптимізації початкового розв'язування задачі комівояжера з кластерним розподілом точок. Початковий розв'язок складається з об'єднання часткових маршрутів між кластерами та маршрутів всередині кластерів. Кластери формуються з груп точок, що знаходяться в близькому околі.

The local optimization algorithms of initial solution of the clustered TSP are described. The initial solution is determined as concatenation of initial partial routes between clusters and routes in clusters. The clusters are formed from the groups of neighboring points.

Вступ

Задача комівояжера є однією з найкраще досліджених задач комбінаторної оптимізації. Сьогодні існує багато алгоритмів для її розв'язання. Задача комівояжера з кластерним розподілом точок – це окремий вид задачі комівояжера, що має широке прикладне застосування. Значне зростання її розмірності в останні роки та застосування в системах реального часу вимагає ефективних підходів, які б забезпечили отримання якісних розв'язків та мали б малу обчислювальну складність.

Формулювання задачі

У класичному формулюванні заданими вважають множину N з n точок ($|N|=n$), описаних їхніми координатами (x_i, y_i). Необхідно знайти маршрут S^*, що проходить по одному разу через кожну точку, довжина якого $L^*(S^*)$ є мінімальною:

$$L^*(S^*) = \Sigma_{ij}\, l_{ij}^* \rightarrow min\Sigma_{ij}\, l_{ij}^* \quad \forall\, l_{ij} \in l_{ij}',$$