

## 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 формується 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

© Базилевич Р., Дюпа Р., Кутельмах Р., 2006

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 оточенні.

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:

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

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

Розглядатиметься випадок, для якого точки розподілені у вигляді кластерів та утворюють множину з  $k$  кластерів  $C = C_1, \dots, C_k$ . Задача розв'язується в три етапи:

1. Кластери описують множинами точок. Для кожного кластера знаходять найкоротший маршрут між двома його граничними точками  $\alpha$  і  $\beta$  (точки  $\alpha$  і  $\beta$  з'єднані з двома його сусідніми кластерами).
2. Пошук початкового маршруту. Початковим маршрутом вважають об'єднання маршрутів між кластерами та маршрутів у всіх кластерах.
3. Оптимізація маршруту. Покращують розв'язок шляхом ітераційної оптимізації довжин шляхів на окремих його ділянках та загалом:

$$L^{*0} \rightarrow L^{*1} \rightarrow L^{*2} \rightarrow \dots \rightarrow L^{*k}.$$

Реалізують два типи оптимізації:

- 3а) часткова оптимізація на окремих ділянках шляху;
- 3б) глобальна оптимізація шляхом ітераційного перегляду всього маршруту.

Розглянемо детальніше особливості запропонованих алгоритмів.

### Пошук початкового розв'язку

Обходячи послідовно усі суміжні кластери та з'єднуючи їхні розв'язки, отримаємо повний маршрут. Окремою є задача комівояжера у кластерах, для яких заданими є дві граничні точки. Початкову та кінцеву точки можна вибирати за різними критеріями, зокрема як найближчі точки до сусідніх кластерів. Для цього застосовано новий підхід – метод “умовного ребра”. Маючи початкову та кінцеву точки, з'єднуємо їх ребром, довжину якого штучно задаємо дуже малою величиною. За допомогою базового алгоритму розв'язуємо задачу комівояжера для точок, що входять до кластера. При цьому умовне ребро залишиться в результуючому маршруті і потрібно лише його вилучити. Отримаємо розірваний цикл, який буде частиною загального розв'язку. На рис. 1 показано кластер з граничними точками, результат вирішення задачі комівояжера в ньому та отриманий маршрут.



Рис. 1. Визначення крайніх точок кластера

Маючи розв'язки у кожному кластері та послідовність обходу кластерів, можна створити результуючий маршрут. Для об'єднання кластерів останню точку поточного кластера з'єднуємо з першою точкою наступного кластера і так для всього маршруту. На рис. 2 зображено об'єднання всіх розв'язків у кластерах в єдиний розв'язок задачі комівояжера.

Важливою підзадачею є ідентифікація початкової та кінцевої точок кожного кластера (граничних точок). Найпростіший спосіб вибору цих точок – вибір взаємно найближчих точок між сусідніми кластерами. Проте, оптимальний маршрут далеко не завжди передбачає перехід від кластера до кластера лише за найближчими точками. Можливим є варіант, коли довжина маршруту у кластері різко скоротиться у разі вибору дещо іншої, не найближчої до сусіднього кластера, точки. На рис. 3 наведено приклад такої ситуації; ліворуч зображено вибір крайніх точок як найближчих поміж сусідніми кластерами і результуючий маршрут, праворуч – вибір іншої точки в кластері як першої і результуючий маршрут, який є коротшим.

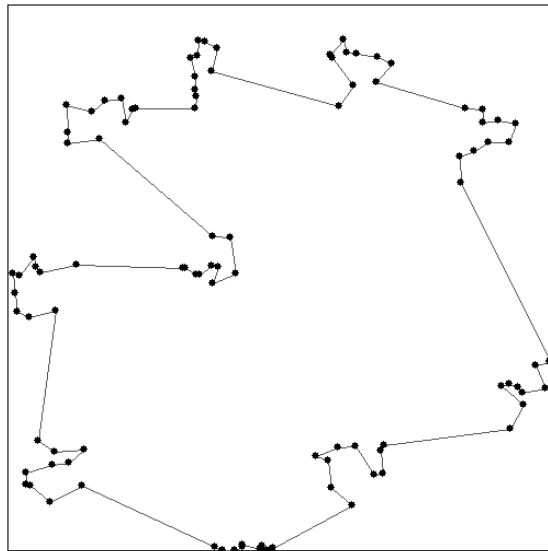


Рис. 2. Результат розв'язання задачі комівояжера

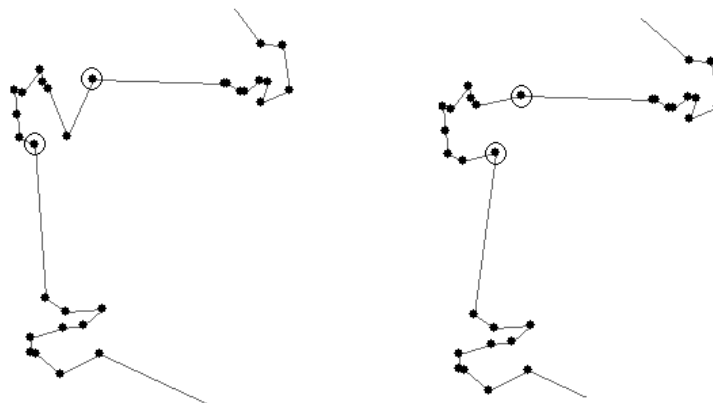


Рис. 3. Вибір крайніх точок кластера впливає на загальний результат

Довжина загального маршруту зменшиться лише в тому випадку, якщо різниця в довжині між різними маршрутами в кластері буде більшою, ніж різниця в довжині між ребрами, що з'єднують кластер з сусідніми. Досліджено низку стратегій для пошуку граничних точок у кластері та створення повного маршруту. Кожна стратегія передбачає початкову ідентифікацію взаємно найближчих точок між кластерами як крайніх, а на наступних кроках їх ітераційну зміну. Кожній стратегії властива певна елементарна область сканування, тобто область, де визначатимуться граничні точки кластерів та маршрути руху в кластерах. Опишемо основні стратегії.

### Алгоритми локальної оптимізації

**Стратегія ESA1.** Елементарною областю сканування є кластер та початкова точка наступного кластера (рис. 4).

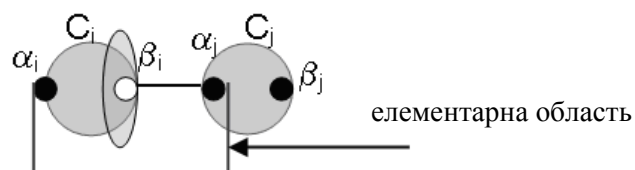


Рис. 4. Елементарна область стратегії ESA1:  
 $\alpha_i, \alpha_j$  – фіксовані точки,  $\beta_i$  – нефіксована точка

Ця стратегія має декілька варіантів. Перший (рис. 5, *a*) – обхід усіх кластерів від першого до останнього із встановленням кінцевої точки кожного поточного кластера і використання її як фіксованої під час сканування наступного кластера (алгоритм EAS1-L). Також можливий інший підхід (рис. 5, *b*) – перебір усіх кластерів в іншому напрямку – від останнього до першого, тоді плаваючою буде початкова точка кожного кластера (алгоритм EAS1-R). Третій підхід (рис. 5, *c*) – комбінація першого і другого підходів – обходимо кожен кластер в одному напрямку, а потім у зворотному. При цьому кожен кластер буде сканований двічі (алгоритм EAS1-LR).

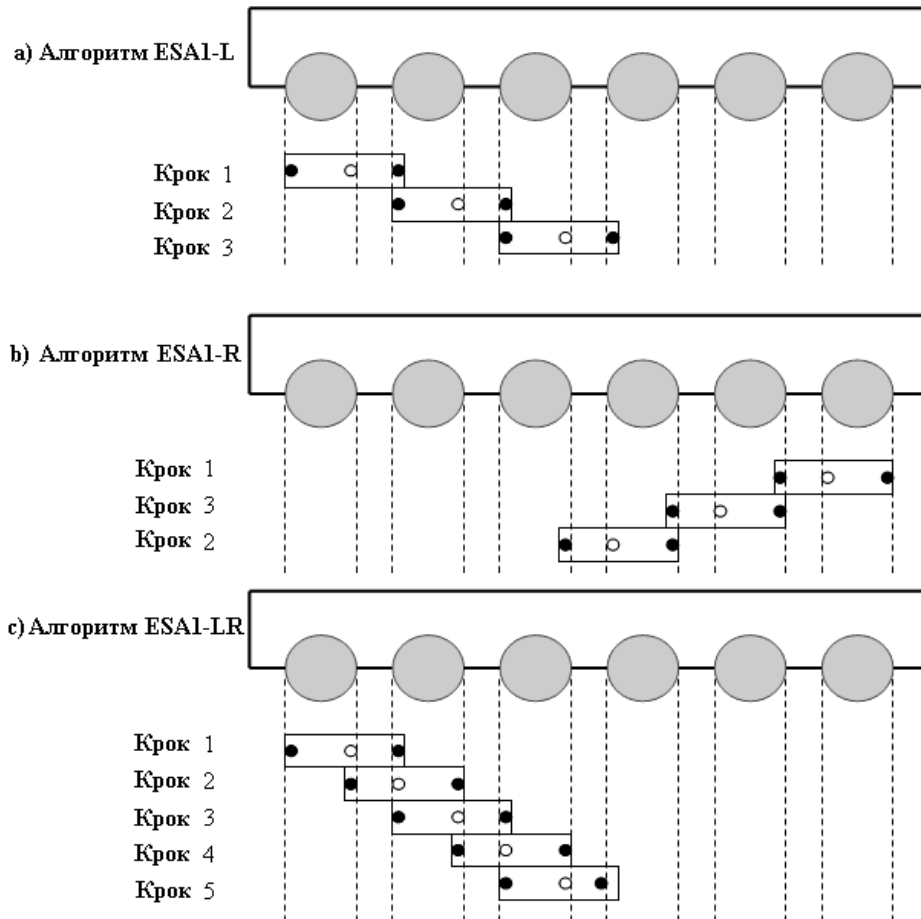


Рис. 5. Стратегія EAS1

Стратегія EAS1 є найпростішою та найекономнішою процедурою сканування. У ній сканувальну область збільшено лише на одну точку від розміру кластера. Проте тут лише половину крайніх точок розглядають як нефіксовані. Підхід EAS1-LR передбачає сканування кожної елементарної області двічі, він вимагатиме вдвічі більше часу на сканування, проте забезпечується вища точність, оскільки кожен граничну точку кожного кластера розглядають як нефіксовану.

**Стратегія ESA2** (Elementary scanning area 2). Тут елементарною областю є кластер та граничні точки двох суміжних кластерів. На рис. 6 зображено область сканування для стратегії ESA2.

Для стратегії запропоновано два підходи до сканування кластерів. Перший (рис. 7, *a*) – сканування по черзі кожного кластера та встановлення в кожному плаваючих граничних точок. Визначену граничну точку у кластері на наступному кроці використовуватимуть під час сканування наступного кластера як першу точку елементарної області (алгоритм ESA2-1). Другий підхід (рис. 7, *b*) – сканують спочатку парні за порядком обходу кластери, в яких встановлюють плаваючі граничні точки, тобто кластер 2, кластер 4, кластер 6 і т. д., а потім скануються непарні за порядком обходу кластери – кластер 1, кластер 3, кластер 5 і т. д., до елементарних областей яких входять встановлені під час першого проходу крайні точки парних кластерів.

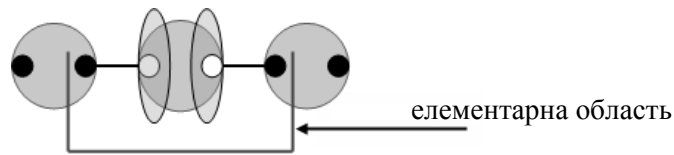


Рис. 6. Елементарна область стратегії ESA2

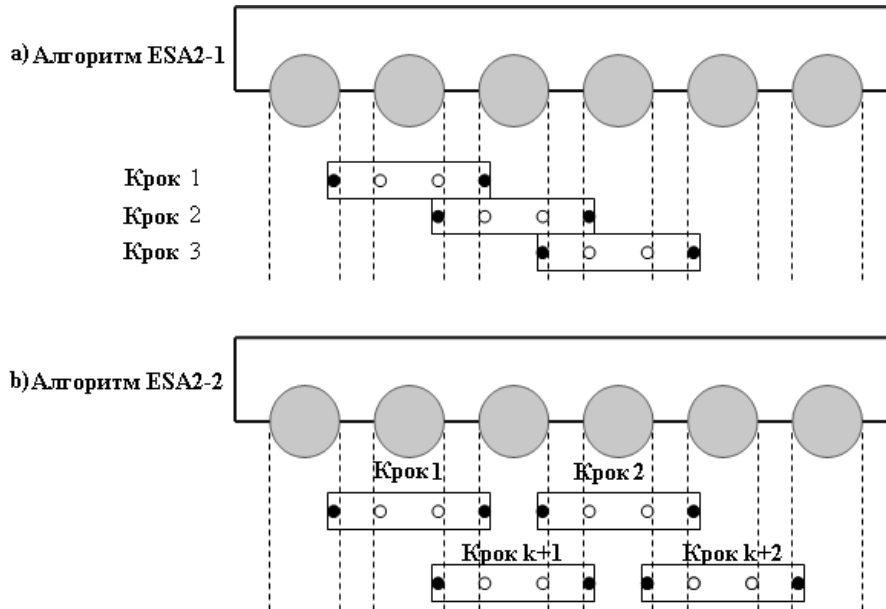


Рис. 7. Стратегія ESA2

Стратегія ESA2, як і стратегія ESA1, є економною, розмірність елементарної області порівняно з першою стратегією збільшено лише на одну точку. Кожен кластер сканується лише один раз.

### Експерименти

Було проведено 5 тестів для задачі розмірністю 10000 точок та один тест для задачі розмірністю 75000 точок. У кожному тесті розмір кластера був фіксованим – 10 точок. Результати розв'язування задачі порівнювалися з результатами базового алгоритму – 2-opt. Усі тести проводилися на ПК з процесором Pentium IV – 3000 МГц та об'ємом оперативної пам'яті 512 Мб.

Характеристики за часом та якістю усіх тестів наведено в таблиці:

#### Результати тестування задач розмірностями 10000 та 75000 точок

	Розмірність задачі – 10000	Розмірність задачі – 75000
Алгоритм	Час виконання, с	Час виконання, с
2-opt	251,9	17613
ESA1-L	2,81	133
ESA1-R	2,81	133
ESA1-LR	2,95	134
ESA2-1	2,9	133
ESA2-2	2,9	133
Алгоритм	Довжина маршруту відносно базового алгоритму	Довжина маршруту відносно базового алгоритму
2-opt	0,00%	0,00%
ESA1-L	-0,20%	0,10%
ESA1-R	-0,25%	0,10%
ESA1-LR	-1,50%	-1,30%
ESA2-1	-0,50%	-0,20%
ESA2-2	-0,50%	-0,20%

## Висновки

Алгоритми локальної оптимізації дають змогу істотно зменшити час розв'язування задачі комівояжера, що особливо важливо у випадку їхньої великої розмірності. Стратегія ESA1-LR демонструє покращення якості обчислень в порівнянні з базовим алгоритмом і приблизно 3% оптимізацію порівняно з початковим розв'язком та приблизно в 100 разів менший час обчислень для задачі розмірністю 10000 точок. Ця стратегія, за якою кожен кластер сканують двічі, забезпечує вибір кращого маршруту всередині кластера.

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 621.397.3

О. Березький, Ю. Батько

Тернопільський державний економічний університет

## АНАЛІЗ АЛГОРИТМІВ ВИДІЛЕННЯ КОНТУРІВ ЗОБРАЖЕНЬ

© Березький О., Батько Ю., 2006

**Досліджено алгоритми виділення контурів зображень, запропоновано та проаналізовано алгоритм визначення зовнішнього контуру зображень, а також наведено приклад програмної реалізації запропонованого алгоритму та результати його роботи.**

**Algorithms of selection of contours of images are explored, algorithm of determination of external contour of images is offered and analyzed, and also an example of program realization of offered algorithm and results of its work is introduced.**

### Вступ

Одним із актуальних напрямків розвитку сучасної прикладної науки є широке впровадження комп'ютерних засобів у медицину. Зокрема комп'ютерні технології з набором спеціалізованих програмних засобів використовують для діагностики пацієнтів, швидкого оброблення інформації та обміну даними, навчання майбутніх спеціалістів тощо [1]. На сучасному етапі поєднання прогресивних інформаційних технологій, нових методів та алгоритмів оброблення, аналізу та синтезу зображень і медицини привели до народження нової області – телемедицини, що передбачає встановлення діагнозу на відстані на основі аналізу та оброблення зображень клітин органів людини. Як відомо [2, 3], кожен тип клітин має свої ознаки: відповідну геометричну форму та характерне комбіноване забарвлення. Виділення цих ознак в автоматичному режимі є одним з основних завдань, які мають вирішити розробники алгоритмічного та програмного забезпечення у цій галузі.

Завданням статті є аналіз алгоритмів проходження зовнішнього контуру зображення, а також розроблення алгоритму виділення контурів зображень пухлинних клітин.

### Огляд літературних джерел

Одним із важливих етапів аналізу зображення є коректне та швидке виділення та проходження контуру зображення. Проаналізуємо відомі алгоритми визначення зовнішнього контуру зображення. Перевагами алгоритму “Square Tracing” [6] є простота реалізації та швидкодія, недоліками – неефективне опрацювання зображень із діагональними сторонами і зображень, які