— продаж незавершеного будівництва;
— продаж законсервованих потужностей;
— продаж невикористовуваних видів основних засобів та нематеріальних активів;
— інвестиції в нове обладнання.

Підсумовуючи, можна сказати, що оптимізація грошових потоків підприємства має своє відображення в системі формування планів і може використовуватись при складанні прогнозних документів.

1.Бланк И. А. Финансовий менеджмент. – К, 2000. 2. http://www.chat.ru/~saisa/index.html 3. http://www.ai.tsi.lv/ru/ga/ga_intro.html 4. Ротштейн А.П., Интеллектуальные технологии идентификации: нечеткие множества, генетические алгоритмы, нейронные сети. – Винница: УНІВЕРСУМ–Вінниця, 1999. – 320 с. 5. Генетические алгоритмы, искусственные нейронные сети и проблемы виртуальной реальности / Г.К. Вороновский, К.В. Махотило, С.Н. Петрашев, С.А. Сергеев. – Харьков: Основа, 1997. – 212 с. 6. Михалевич В.С., Волкович В.Л., Яценко Ю.П. Многокритериальный анализ темпов конверсии на базе интегральных моделей. Кибернетика и системный анализ, 1993. – С.36–46.

УДК 004.93’14

Р. Мельник, Р. Тушницький
Національний університет “Львівська політехніка”,
кафедра програмного забезпечення

КЕРУВАННЯ ТОЧНІСТЮ ТА СКЛАДНІСТЮ АЛГОРИТМУ КЛАСТЕРИЗАЦІЇ ДАНИХ ВЕЛИКОЇ РОЗМІРНОСТІ ДОПУСКОМ НА ФУНКЦІЮ ПОДІБНОСТІ ТА ДЕКОМПОЗИЦІЄЮ

Розглянуто агломеративний ієрархічний алгоритм кластеризації даних. Запропоновано коефіцієнт швидкості для зменшення алгоритмічної складності без втрат точності алгоритму. Приведено результати із якісними характеристиками кластеризації тестових даних.

The clustering agglomerative hierarchical algorithm for data grouping is considered. To reduce algorithmic complexity without accuracy losses an approach with the speed and accuracy coefficient is proposed. Some results with quality characteristics of clustered test data are presented.

Вступ

Методи кластерного аналізу широко використовуються для декомпозиції, дослідження та розпізнавання зображень [1 – 4]. У представленій роботі розглянуто інструмент для управління складністю та точністю алгоритму кластеризації. Проведені дослідження підтверджують, що із збільшенням коефіцієнта швидкості у визначених межах, складність алгоритму зменшується без втрат точності кластеризації.

1. Керування алгоритмічною складністю та точністю

Практичні класи задач кластеризації даних є часозалежними завдяки обсягам даних і складності алгоритму, щоб досягти відповідних характеристик класифікації. Для скорочення втрати часу, одночасно задовольняючи вимоги точності, пропонуємо підхід, подібний, але простіший за розглянутий [4].

Традиційний агломеративний ієрархічний алгоритм для кластеризації даних має наступні кроки:

S0. Для всіх точок x_i, $x_j \in X$.

S1. Пошук пар кандидатів за функцією подібності:

$$\forall (x_i, x_j\ (j > i)) \quad \text{підрахунок} \quad F(x_i, x_j).$$

S2. Пошук пар, що мають найменше значення відстані

$$F^*(x_i,x_j)=\min\ F(x_i,x_j), \qquad i,j\in I,$$

та об'єднання точок x_i, x_j, створення нової точки (кластера) x_{n+1}.

S3. Видалення точок x_i, x_j зі списку кандидатів.

S4. Кінець (для всіх x_i, $x_j \in X$).

Для найкращої точності універсальний ієрархічний алгоритм характеризується поліноміальною складністю $O(N^3)$. Для зменшення складності до $O(N^2)$ автори в [4] пропонують знайти так званих друзів, об'єднаних на однаковому рівні дерева. Зауважимо, що автори не пояснюють витрати, необхідні для знаходження найменших відстаней для всіх друзів на кожному рівні дерева.

Пропонується подібна ідея для зменшення алгоритмічної складності без обчислення відстаней друзів, але накладанням в алгоритмі допусків на значення відстані для об'єднання точок на певному рівні дерева. На кроці S3 класичного алгоритму ми об'єднаємо ті пари вершин (кластерів), що задовольняють таку умову:

$$F(x_i, x_j) \geq F_0(1- k_v), \tag{1}$$

де F_0 – мінімальне значення відстані на рівні згортання, k_v ($k_v < 1$) – коефіцієнт допуску, що вказує на відстань між кандидатами для об'єднання на поточному рівні дерева (назвемо k_v коефіцієнтом швидкості та точності).

Функцію F утворимо як зважену суму модулів різниць (Манхеттенської відстані):

$$F_{ij} = \{w_1|a_i - a_j| + w_2|b_i - b_j| + w_3|c_i - c_j| + ...\} / k \cdot r, \tag{2}$$

або як зважену суму квадратів (евклідова відстань):

$$F_{ij} = \{w_1[a_i - a_j]^2 + w_2[b_i - b_j]^2 + w_3[c_i - c_j]^2 + ...\} / N, \tag{3}$$

де a, b, c – характеристики властивостей, що формують точку чи кластер в просторі і виражаються числом; k, r – кількість точок в i *–му* та j*-му* кластерах дерева згортання. Сумування йде по всіх точках, N – кількість вхідних точок.

Для нового кластера характеристики формуються усередненням характеристик двох вхідних кластерів:

$$Q_l = Q_l((a_i + a_j) / 2, (b_i + b_j) / 2, (c_i + c_j) / 2, ...), \tag{4}$$

або зваженим усередненням двох компонент характеристик:

$$Q_k = Q_k((k \cdot a_i + r \cdot a_j)/(k + r), (k \cdot b_i + r \cdot b_j)/(k + r),\ ...), \tag{5}$$

Для якісної оцінки кластерів, сформованих різними рівнями дерева, використовуємо критерій Варда (Ward) відхилень квадратів:

$$E=\{\sum E_k\}/M=\{\sum\sum d_j\}/M=\{\sum[\sum(a_i^*-a_j)^2+(b_i^*-b_j)^2+...]/m_j\}/M, \quad k\in\overline{1,M}, \quad i,j\in J \tag{6}$$

де a_i^*, b_i^*, c_i^* – координати центру кластера, a_i, b_i, c_i – координати точок кластера, E_k – девіація кластера, E – інтегральна девіація, d_j – девіація точок кластера відносно центру кластера, M – кількість кластерів на контролюючому рівні ієрархічного дерева.

Щоб скоротити втрати часу, пов'язані із оцінкою якісної функції, пропонуються простіші функції – питома густина для даного рівня ієрархічного дерева [5, 6]:

$$S = \{\sum L_k\ / D_k\} / M, \tag{7}$$

де L_k, D_k є кількістю точок кластерів і його діаметр, або оберненою характеристикою є питомий об'єм на точку:

$$V = \{\sum D_k / L_k\} / M. \quad (8)$$

У формулах (7,8) для оцінки простору кластерів пропонується використати діаметр кластера D_k як максимальну віддаль між точками кластера.

На рис. 1 зображено орієнтовні залежності складності та точності від коефіцієнта швидкості: А.С. – алгоритмічна складність, А.А. – точність алгоритму.

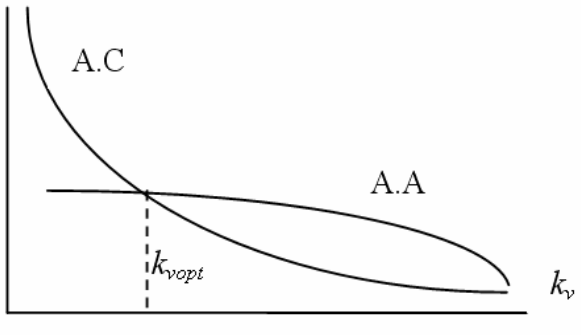


Рис. 1. Складність та точність за коефіцієнтом швидкості

Задача полягає в знаходженні найкращого значення коефіцієнта швидкості k_{vopt} , яке можна використати в алгоритмі для конкретної вибірки даних.

Найкраще значення коефіцієнта k_{vopt} таке, для якого алгоритмічна складність набуває мінімального значення, а точність алгоритму не має втрат. За цим коефіцієнтом результуючі кластери є такі сам, як при точній кластеризації з параметром $k_v = 0$.

Для знаходження цього коефіцієнта для вибірки даних пропонуємо такий наближений підхід. Маючи визначені відстані матриці Z (її розмірність є N×N) для N точок ми шукаємо вектор найменших різниць між елементами матриці:

$$G = G(g_k = z_{ij} - z_{ik}),\ k \in \overline{1,N},\quad i, j \in J \quad (9)$$

Для оцінки k_{vopt} приймемо відношення $k_v \approx g_k / z_{ij.}$

2. Експериментальні дослідження

Вхідні дані нормалізовані за формулою:

$$\frac{V - \min V}{\max V - \min V}. \quad (10)$$

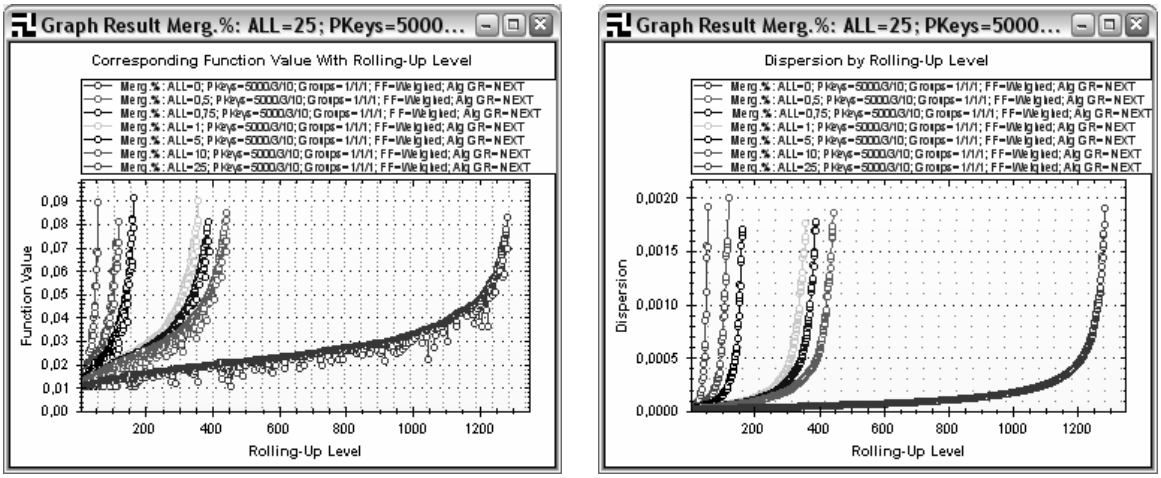
Спочатку експерименти були проведені для підтвердження адекватності функції до оцінювання точності алгоритму. Рис. 2 демонструє характер зміни функції подібності (*а*) та питомої дисперсії (*б*) під час процесу кластеризаціїї для значень коефіцієнта k_v: 0, 0.5%, 0.75%, 1%, 5%, 10%, 25% за рівнями дерева згортання.

Рис. 4 демонструє характер зміни функції подібності (*а*) та питомої дисперсії (*б*) під час процесу кластеризаціїї із каскадною декомпозицією.

Рис. 5 ілюструє залежність характеристики питомого об’єму (*а*) та зміну питомої густини (*б*) під час процесу згортання із каскадною декомпозицією.

Рис. 6 ілюструє залежність характеристики питомого об’єму (*а*) за кількістю кластерів на рівні дерева згортання та зміну функції на ділянці 0-60 кластерів на рівні (*б*) під час процесу згортання.

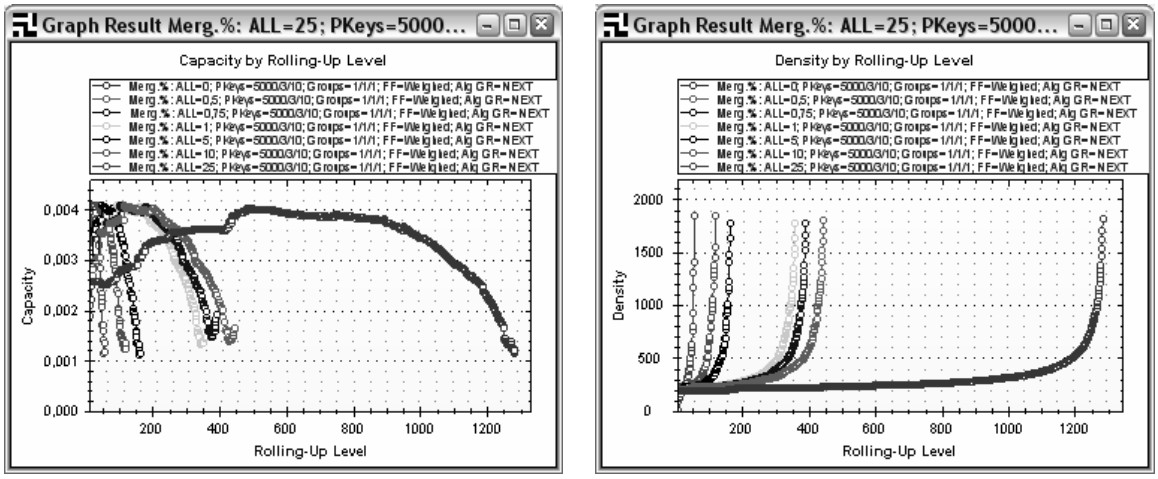
У табл. 1 показано зміну часової залежності (алгоритмічну складність) від коефіцієнту швидкості (0, 1%, 5%, 10%) та алгоритму згортання без та з використанням каскадної декомпозиції [5–6].



а *б*

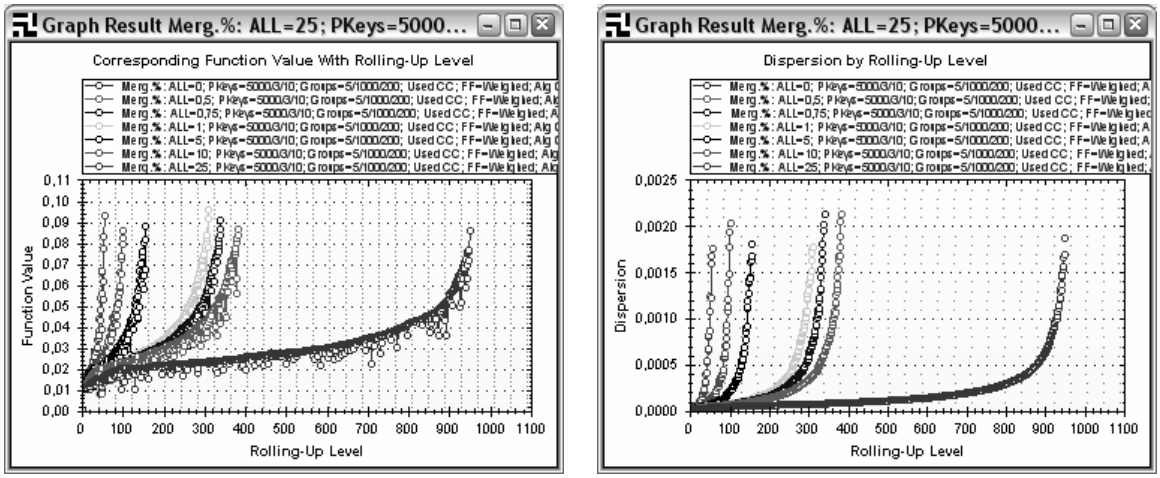
Рис. 2. Функція відстані та питомої дисперсії за рівнями дерева згортання

Рис. 3 ілюструє залежність характеристики питомого об'єму (*а*) та зміну питомої густини (*б*) під час процесу згортання.



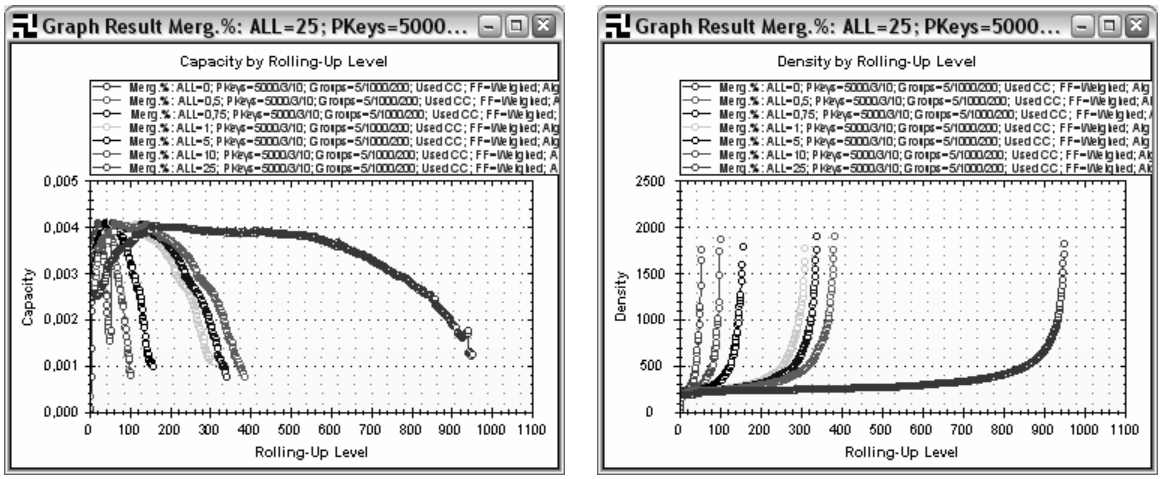
а *б*

Рис. 3. Питомі об'єм та густина за рівнями дерева згортання



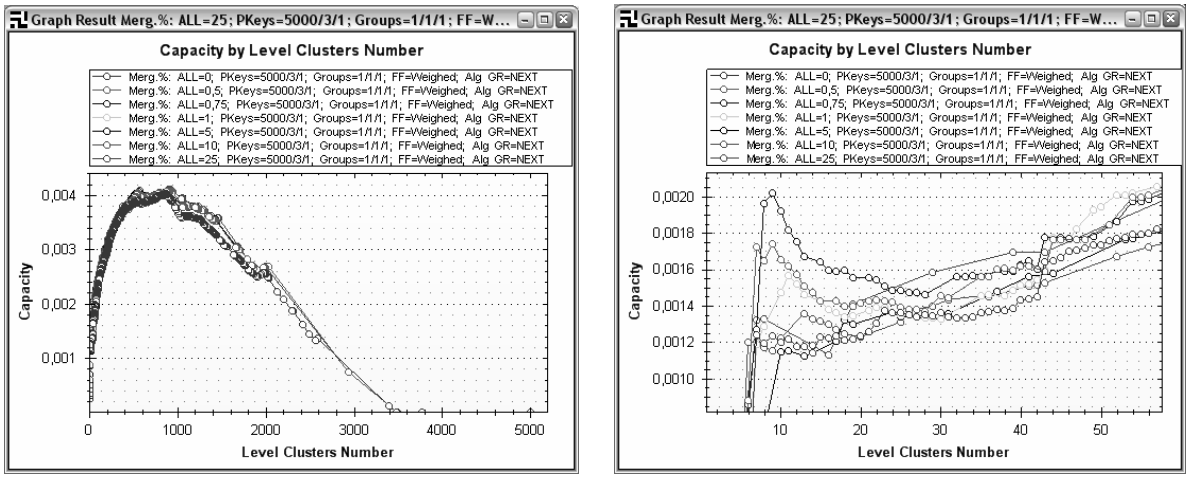
а *б*

Рис. 4. Функція відстані та питомої дисперсії за рівнями дерева згортання із каскадною декомпозицією



а *б*

Рис. 5. Питомі об'єм та густина за рівнями дерева згортання із каскадною декомпозицією



а *б*

Рис. 6. Питомий об'єм за кількістю кластерів на рівні та ділянка 0–60 кластерів на рівні

Таблиця 1

Часові залежності алгоритму

Кількість точок	Коефіцієнт швидкості, %	Час без каскадінгу, с	Час із каскадінгом, с	Параметри декомпозиції
5000	0	43,281	11,0625	5 груп по 1000 елементів, 200 результуючих з групи, використано 1 каскад
	1	11,312	3,64	
	5	6,765	2,3125	
	10	5,203	1,73437	
10000	0	230,250	27,765	10 груп по 1000 елементів, 200 результуючих з групи, використано 2 каскади
	1	52,765	8,75	
	5	25,593	5,203	
	10	20,468	3,734	
50000	0	не обчислено	405,640	50 груп по 1000 елементів, 200 результуючих з групи, використано 3 каскади
	1		113,625	
	5		67,312	
	10		52,234	

Залежність алгоритмічної точності було визначено експериментально для значення коефіцієнту швидкості k_v: 0, 0.5%, 0.75%, 1%, 5%, 10%, 25% (кількість точок – 5000). У табл. 2 наведено зміну значень (отриманих на рівні дерева згортання R із кількістю кластерів 10) питомих дисперсії, густини та об'єму для основної процедури алгоритму та залежності для випадку каскадної декомпозиції.

Таблиця 2

Точність алгоритму (R = 10)

Коефіцієнт швидкості, %	Без каскадної декомпозиції			Із каскадною декомпозицією		
	Питомий об'єм	Питома густина	Питома дисперсія	Питомий об'єм	Питома густина	Питома дисперсія
0	0,0011986	1813,7595	0,00189974	0,0012511	1816,4741	0,0018677
0,5	0,0016516	1807,6029	0,00186511	0,0007661	1903,2552	0,0021244
0,75	0,0019217	1770,0324	0,00177828	0,0007661	1903,2552	0,0021244
1	0,0014725	1767,4991	0,00176427	0,0010792	1766,1849	0,0017718
5	0,0011485	1768,4239	0,00170764	0,0009931	1794,187	0,0018032
10	0,0012218	1845,1773	0,00199867	0,0008379	1734,461	0,0019268
25	0,0013259	2147,6669	0,00222705	0,0036906	1633,1251	0,0016681

За результатами табл. 2 можна зробити висновок, що для цієї вибірки (5000 3-вимірних точок, згенерованих за рівномірним законом розподілу у межах значень від 1 – 100, нормалізованих) оптимальний коефіцієнт швидкості k_{vopt} є у проміжку 5–10 %. Значення k_v, більші за 5%, алгоритмічну складність зменшують, але й точність алгоритму зменшується.

3. Програмний пакет

Розроблено експериментальний програмний пакет з інтерфейсом користувача, що контролює всі етапи процесу: введення даних, параметрів, проміжні та кінцеві результати, параметри кластеризації (повна, часткова), звіти виконання алгоритму, результуючі параметри.

На рис. 7 зображено розфарбовані результуючі кластери для згортання вибірки на 5000 точок.

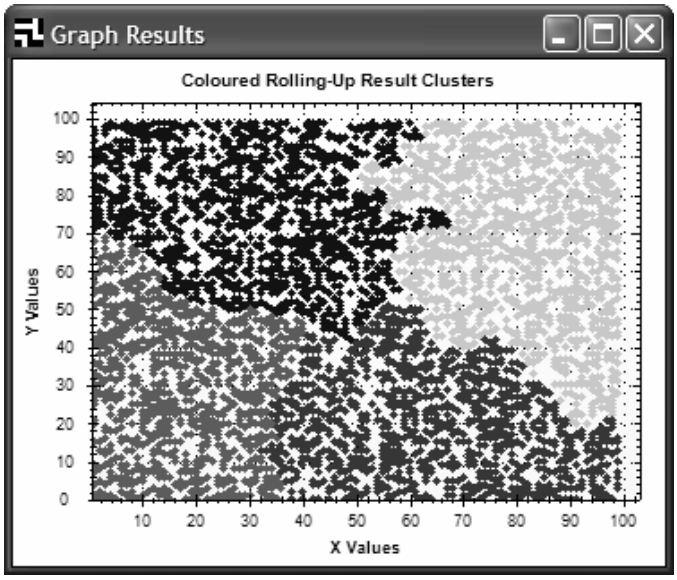


Рис. 7. Розфарбовані результуючі кластери для згортання вибірки на 5000 точок

Висновки

Розроблено алгоритм для кластеризації даних, що ґрунтуються на агломеративному ієрархічному підході. Реалізовано нову ідею для контролю алгоритмічної точності та складності за коефіцієнтом швидкості. Створено програмний пакет для кластеризації даних. Проведені експериментальні дослідження для великих вибірок даних підтверджують ефективність запропонованої ідеї: зменшення кількості об'єднань кластерів на рівнях дерева згортання без втрат алгоритмічної точності. Зменшення алгоритмічної складності дає змогу застосовувати алгоритм для великих груп вибірок, таких як візуальні образи, гени або тексти.

1. Andy M Yip, Chris Ding, Tony F.Chan. Dynamic Cluster Formation Using Level Set Methods. – IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.28, n. 6, pp.877-889, June, 2006. 2. Leo Grady, Eric L.Schwartz. Isoperimetric Graph partitioning for Image segmentation. – IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.28, n. 3, pp.469-475, March, 2006. 3. M. Pavan, M. Pelillo. Dominant sets and Pairwise Clustering. – IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.29, n. 1, pp.167-172, January, 2007. 4. C. Ding, X. He. Cluster Aggregate Inequality and Multilevel Hierarchical Clustering, Proc. 9th European Conf. Principles of Data Mining and Knowledge Discovery (2005). p. 71−83. 5. Мельник Р.А., Алексєєв О.А. Кластеризація ключів образів на основі декомпозиції їх множини // Відбір і обробка інформації. – 2006. – Вип. 24(100). – С.110–114. 6. Р. Мельник, Р. Тушницький. Каскадна декомпозиція множин великої розмірності при кластеризації ключів образів // "Комп'ютерні науки та інформаційні технології". – 2008. – № 604. – С. 249–254.

УДК 519.16

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

ОПТИМІЗАЦІЯ РОЗВ'ЯЗКІВ ЗАДАЧІ КОМІВОЯЖЕРА МЕТОДОМ ПОСЛІДОВНОГО СКАНУВАННЯ

Запропоновано новий метод оптимізації розв'язків задачі комівояжера. Метод може бути застосований для оптимізації початкового розв'язку задачі, отриманого за допомогою декомпозиції чи для покращення маршруту, отриманого будь-яким алгоритмом. Вхідними даними є маршрут, який необхідно покращити.

New approach for Traveling Salesman Problem(TSP) solutions optimization is proposed. Approach can be applied for initial solution optimization, calculated with the help of decomposition algorithm or for route optimization, calculated by any classic algorithm. Route to be improved is an input data for algorithm.

Вступ

Задача комівояжера – одна із основних задач комбінаторної оптимізації, що має широке прикладне застосування [1,2]. Існує небагато алгоритмів, що забезпечують одержання якісних розв'язків задачі комівояжера, особливо при малих часових затратах [3]. Для розв'язування задачі комівояжера алгоритм Ліна–Кернігана є одним з найефективніших [4,5]. Його обчислювальна складність – $O(n^2)$. Одержані результати – в межах 1-3% від оптимального. Впродовж останніх років було запропоновано нову версію алгоритму Ліна–Кернігана – алгоритм Ліна–Кернігана–Гельсгауна [6], який забезпечує отримання оптимального розв'язку задачі для 7397 точок із