

УДК 519.85

С. М. Білан, Н. Р. Кондратенко, О. А. Ткачук

Вінницький національний технічний університет
Хмельницьке шосе, 95, 21021 Вінниця, Україна

Дослідження можливостей генетичного алгоритму в задачах про комівояжера

Розглянуто можливість використання генетичного алгоритму в задачах про комівояжера. Запропоновано методика використання операторів генетичного алгоритму, пристосованого для розв'язання задач великої розмірності.

Ключові слова: комівояжер, вхідні матриці, маркер розбиття, генетичний алгоритм.

Екстремальні задачі комбінаторного типу об'єднують широке коло задач дискретної оптимізації. До цих задач відносяться задачі теорії розкладів, календарного планування, цілочисельного програмування, задачі про комівояжера та інші [1].

Дані задачі мають багато практичних застосувань, наприклад, при проектуванні інформаційних систем та обчислювальних комплексів, у задачах маршрутизації в системах масового обслуговування та комп'ютерних мережах. У цьому плані найбільший інтерес представляє відома «задача комівояжера», суть якої полягає в знаходженні найкоротшого замкнутого шляху обходу вершин графа (міст, станів певної системи і т. п.), які задані своїми координатами [2].

Відомо, що задачі такого типу в загальному випадку мають багатоекстремальний характер і є NP-повними [1, 2]. Наприклад, при розв'язанні задачі комівояжера, як правило, використовуються лише наближені алгоритми розв'язання. Вони мають достатньо високу швидкодію та поліноміальну часову складність, але дають лише наближені до точних результати. Точні методи типу простого перебору, гілок та границь дозволяють ефективно розв'язувати такі задачі, але мають експоненціальну складність, що не дозволяє використовувати їх для задач великої розмірності.

Сучасний підхід до розв'язання такого класу задач пов'язують з використанням методів штучного інтелекту — нейроподібних мереж та еволюційних алгоритмів випадково-детермінованого пошуку, серед яких найбільш поширеними є генетичні [3–7]. Відомо, що такі структури моделюють роботу кори мозку людини

та еволюційний процес розвитку живої природи, мають можливість паралельної обробки даних, і, за рахунок цього, забезпечують суттєвий вигравш часу при розв'язанні задач великої розмірності.

У даній статті ставиться задача провести дослідження можливостей генетичного алгоритму при розв'язанні задач про комівояжера, складність яких змінюється за рахунок збільшення загальної розмірності задачі (кількості вершин графа для обходу) та кількості комівояжерів.

Постановка задачі

Поставимо задачу розробки структури генетичного алгоритму, який буде пристосований для розв'язання задач про комівояжера великої розмірності, в постановці яких враховується змінна кількість комівояжерів.

Припустимо, є певна кількість міст m і задані матриці

$$D^r = [d_{ij}^r], \quad i, j = 1, \dots, m; \quad r = 1, 2, \dots, n,$$

де d_{ij}^r час переходу r -го комівояжера з міста i в місто j .

Необхідно за допомогою генетичного алгоритму знайти таке розбиття множини міст M на підмножини M_1, M_2, \dots, M_n , де $M_n \subseteq M$, $\bigcup_{r=1}^n M_r = M$, щоб для замкнутих гамільтонових контурів r -го комівояжера в підмножині M_r виконувалась умова оптимальності у вигляді:

$$\sum_r L_r = \sum_r \sum_{(i,j) \in L_r} d_{ij}^r \rightarrow \min, \quad (1)$$

де L_r — довжина замкнутого гамільтонового контуру в підмножині M_r .

У випадку $r = 1$, постановка задачі зводиться до однієї вхідної матриці і має традиційний вигляд [1, 2].

Для розв'язання задач про комівояжера в даній постановці розглянемо генетичний алгоритм, який буде їх розв'язувати за такими вхідними даними: число комівояжерів $r = 1, 2, 3$ та розмірність вхідних матриць $m = 6, 10, 20, 100$.

Методика досліджень

Відомо, що згідно еволюційної теорії, кожен біологічний вид розвивається і змінюється для того, щоб найкращим чином пристосуватися до навколишнього середовища. Тобто еволюція, в певному розумінні, є процесом оптимізації всіх живих організмів. У природі задача оптимізації розв'язується шляхом природного відбору, суть якого полягає в тому, що більш пристосовані особини мають більше можливостей для виживання та розмноження. При цьому, завдяки передачі генетичної інформації, нащадки наслідують від батьків їх основні властивості. В результаті еволюційного процесу, після зміни десятків або сотень поколінь, середня пристосованість особин даного виду збільшується. Всі ці відомі біологічні еволю-

ційні підходи до оптимізації знайшли відображення в структурі генетичного алгоритму. Поширеним є підхід до використання даного алгоритму в задачах оптимізації, який наведений у [5–7], зміст якого полягає в наступному. Нехай дано певну складну цільову функцію, яка залежить від декількох змінних, і потрібно знайти такі значення змінних, при яких значення функції максимальне або мінімальне. Тоді розв’язок цієї задачі за допомогою генетичного алгоритму складається з таких кроків. Кожен варіант (набір значень змінних) розглядається як особина, а значення цільової функції для цього варіанту — як пристосованість даної особини. В процесі еволюції пристосованість особин буде зростати. Це означає, що будуть з’являтися більш кращі оптимальні варіанти. Зупинивши еволюцію в певний момент та вибравши кращий варіант, можна отримати достатньо прийнятний розв’язок задачі.

Таким чином, щоб розв’язати задачу в даній постановці, потрібно розробити послідовність операцій, які моделюють еволюційні процеси на основі аналогів механізмів генетичного наслідування і природного відбору.

Представимо загальну схему базового генетичного алгоритму [5], як ітераційний процес, що складається з кількох етапів:

- 1) генерація початкової популяції;
- 2) створення нащадків:
 - а) вибір батьківської пари і здійснення схрещування;
 - б) внесення мутаційних змін у популяцію;
- 3) відбір і формування нового покоління;
- 4) якщо не виконується умова зупинки — перехід до п. 2.

Опишемо більш детально основні етапи генетичного алгоритму, який буде пристосований для розв’язання задач про комівояжера в даній постановці. Генетичний алгоритм починає роботу зі створення початкової популяції. Наведемо приклад однієї хромосоми з популяції для задачі комівояжера $r = 1$ (рис. 1).

2 (ген 1)	5 (ген 2)	4 (ген 3)	3 (ген 4)	6 (ген 5)	1 (ген 6)
-----------	-----------	-----------	-----------	-----------	-----------

Рис. 1. Загальний вигляд однієї хромосоми

Дана хромосома описує замкнутий шлях обходу комівояжером 6 міст. Хромосома складається з 6 комірок, кожна з яких містить номер міста (номери в хромосомі не повторюються). В початковій популяції хромосоми генеруються випадково.

Узагальнимо процедуру генерування початкової популяції хромосом та всі інші етапи генетичного алгоритму відповідно до схеми для базового алгоритму для кількості комівояжерів $r \geq 1$. Етапи схрещування та мутації однакові для будь-якої кількості комівояжерів.

Розглянемо роботу алгоритму для двох комівояжерів. Спочатку відбувається зчитування матриць суміжності для кожного комівояжера. Потім генерується початкова популяція, яка представляє собою набір з t хромосом. Кожна хромосома складається з t комірок, у кожній комірці зберігається номер міста, а хромосома визначає послідовність обходу міст. Додатково кожна хромосома має ще $(r - 1)$

комірок для індексів маркерів розбиття. Для двох комівояжерів це одна комірка, що містить маркер розбиття, який ділить хромосому на 2 підмножини, кожна підмножина визначає шлях обходу для кожного комівояжера, що зображено на рис. 2.

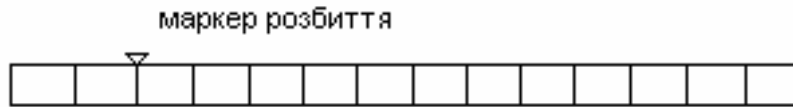


Рис. 2. Вигляд хромосоми з маркером розбиття

Матрицю, в якій зберігаються покоління «батьків» та «нащадків», назвемо робочою матрицею, її зображено на рис. 3.

		1... m	$m + 1$	$m + 2$
1	Батьки (початкова популяція)			
...				
m				
$m + 1$	Хромосоми, отримані в результаті схрещування			
...				
$2m$				
$2m + 1$	Хромосоми, отримані в результаті мутації			
...				
$3m$				

Рис. 3. Робоча матриця

Операція схрещування реалізується як двоточкове схрещування. Тобто, для кожної хромосоми генеруються два випадкові цілі числа з діапазону $[1...m]$ ($k_1 \neq k_2$), і потім виконується перестановка отриманих частин, так як зображено на рис. 4.

До схрещування:

1	1	k_1	$k_1 + 1$	2	k_2	$k_2 + 1$	3	m
---	----------	-------	-----------	----------	-------	-----------	----------	-----

Після:

$k_2 + 1$	3	m	$k_1 + 1$	2	k_2	1	1	k_1
-----------	----------	-----	-----------	----------	-------	---	----------	-------

Рис. 4. Операція двоточкового схрещування

Мутація реалізується як перестановка значень двох випадково обраних генів. При такому перетворенні шлях проходження міняється тільки в двох місцях.

Операція відбору реалізується окремо для кожного варіанту задачі (для одного, двох та трьох комівояжерів). Для одного комівояжера підраховується замкнутий шлях за усіма отриманими хромосомами та відбираються m найкращих. Для двох комівояжерів при реалізації відбору в роботі алгоритму використовується

маркер розбиття (рис. 4), який розділяє випадковим чином отримані хромосоми на дві підмножини, кількість елементів яких не менша 2-х. У результаті перша підмножина визначає шлях першого комівояжера, друга — другого. Для кожного комівояжера підраховується довжина шляху, і по сумі обох шляхів проводиться відбір хромосом у наступне покоління, причому маркер розбиття зберігається в хромосомі в комірці з індексом $m + 1$.

Для трьох комівояжерів використовується вже 2 маркери розбиття. Відбір реалізується аналогічно відбору для двох комівояжерів, тільки хромосома розбивається на 3 частини, кожна з яких — шлях одного з комівояжерів.

Найкращі варіанти хромосом зберігаються і стають батьками в наступному поколінні. Цикл повторюється поки результат — найкращий шлях (розбиття) змінюється (покращується). Таким чином, відбір, описаний вище, працює як елітний. Аналогічний підхід може використовуватись для задач, де кількість комівояжерів більше трьох.

Для реалізації описаних генетичних операцій була створена програма мовою Pascal на основі алгоритму, узагальнену схему якого представлено на рис. 5.

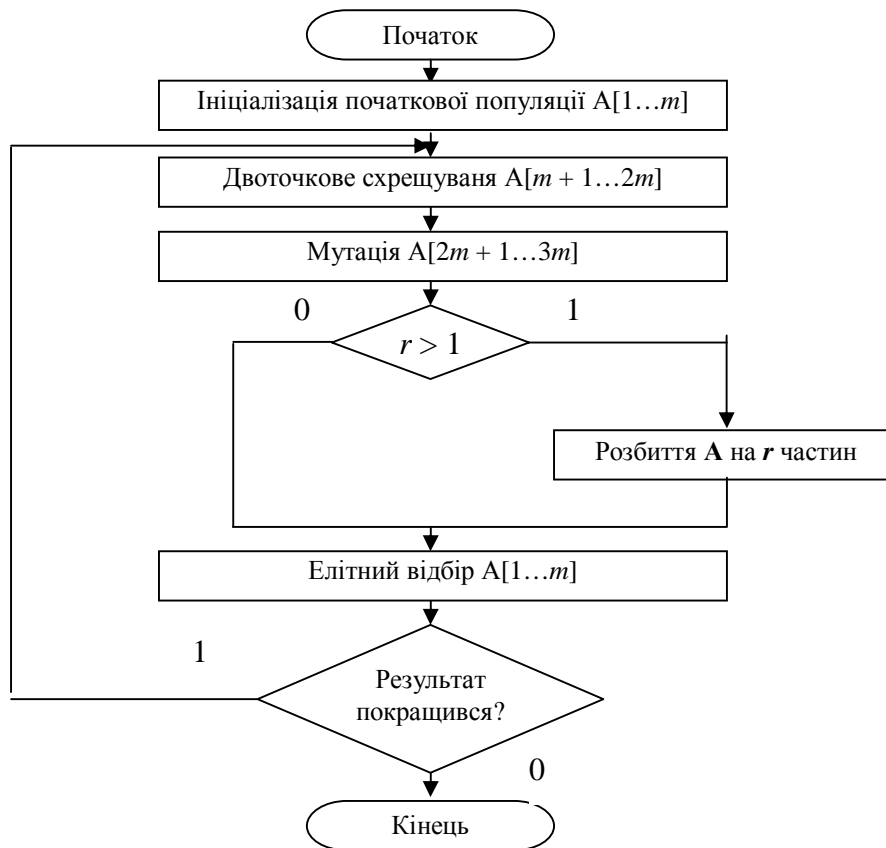


Рис. 5. Блок-схема алгоритму

Комп'ютерний експеримент

Експериментальні дослідження проводились за допомогою єдиної програми, яка дозволила розв'язати поставлену задачу для одного, двох, трьох комівояжерів та кількості міст $m \leq 100$.

Для дослідження оптимізаційних можливостей запропонованого генетичного алгоритму проведено серію з 30 експериментів для різної кількості комівояжерів та різної загальної розмірності задачі. Нижче наведені результати роботи програми для декількох тестових задач. Вхідні дані для першої тестової задачі ($r = 2$, вхідні матриці 6×6) наведено в табл. 1.

Таблиця 1. Вхідні дані для першої тестової задачі

–	5	6	1	7	8	–	5	7	9	23	10
5	–	13	9	7	4	5	-	5	6	1	4
1	34	–	5	6	5	8	9	-	5	5	7
5	5	1	–	8	5	7	5	5	-	45	5
7	4	6	5	–	5	6	7	4	8	-	1
6	3	8	5	9	–	15	1	8	9	19	-

У результаті роботи програми було знайдено такі найкращі варіанти розв'язку:

1) 1-й комівояжер: 6,1,4,3 — 13; 2-й комівояжер: 2,5 — 8 (відповідно критерію (1) з сумою — 21);

2) 1-й комівояжер: 3,1,4,5 — 16; 2-й комівояжер: 2,6 — 5 (відповідно критерію (1) з сумою — 21).

Результати комп'ютерних експериментів розв'язання задачі комівояжера для $r = 1, 2, 3$ та $m = 6$ зведено в табл. 2, де наведено середнє число ітерацій, максимальне, мінімальне та середнє значення цільової функції (1).

Таблиця 2. $r = 1, 2, 3, m = 6$

	1 комівояжер		2 комівояжера		3 комівояжера	
	Ітерацій	Розв'язок	Ітерацій	Розв'язок	Ітерацій	Розв'язок
Середнє значення	2721,4	21,6	2422,9	7,7	2139,9	21,1
Максимальне	3835	22	5207	20	2957	29
Мінімальне	2002	21	2002	6	2002	20

Нижче вказані аналогічні дані для $r = 1, 2, 3$ та $m = 10$ (табл. 3), $m = 20$ (табл. 4)

Таблиця 3. $r = 1, 2, 3, m = 10$

	1 комівояжер		2 комівояжера		3 комівояжера	
	Ітерацій	Розв'язок	Ітерацій	Розв'язок	Ітерацій	Розв'язок
Середнє значення	2795,9	101,3	3699,0	102,2	3891,4	100,6
Максимальне	6841	120	7374	132	8072	135
Мінімальне	2002	86	2020	76	2007	72

Таблиця 4. $r = 1, 2, 3, m = 20$

	1 комівояжер		2 комівояжера		3 комівояжера	
	Ітерацій	Розв'язок	Ітерацій	Розв'язок	Ітерацій	Розв'язок
Середнє значення	3929,4	193,8	5459,7	201,1	5163,4	201,1
Максимальне	8010	221	15116	233	11334	232
Мінімальне	2076	174	2023	175	2197	175

Запропонований генетичний алгоритм найбільш ефективно діє в задачах великої розмірності, що підтверджують результати роботи програми, які були отримані для $r = 1, 2, 3$ та $m = 100$ (табл. 5). Для перевірки якості роботи алгоритму була згенерована матриця відстаней, де значення відстаней між містами випадковим чином встановлювались у діапазоні значень від 1 до 50, тобто наближено можна вважати максимальною границею маршруту значення 5000, мінімальною — 50. Для порівняння з наведеними вище границями маршрутів, генетичний алгоритм дав найбільш прийнятний розв'язок у випадку найскладнішої задачі для $r = 3$ та $m = 100$, що відображено в табл. 5.

Таблиця 5. $r = 1, 2, 3, m = 100$

	1 комівояжер		2 комівояжера		3 комівояжера	
	Ітерацій	Розв'язок	Ітерацій	Розв'язок	Ітерацій	Розв'язок
Середнє значення	6614,7	374,9	13060,0	402,9	10228,2	508,9
Максимальне	11997	422	22723	478	30766	607
Мінімальне	3663	329	3842	340	4270	412

Потрібно зазначити, що остання постановка задачі є дуже складною для розв'язання її за допомогою стандартних методів. До останніх можна віднести два основних підходи для розв'язання такого типу задач — переборний та локально-градієнтний, кожен з яких має свої переваги та недоліки, але вже для $m = 30$ та $r = 1$ пошук оптимального шляху є складною задачею при розв'язанні її вказаними методами [1, 3]. Відомо, що для задач, які розглядаються в даній роботі, універсальних методів, що дозволяють швидко знайти абсолютно точні рішення, не існує. В роботах [3–6] показано, що комбінацією переборного та градієнтного методів, можна отримати наближені розв'язки задач комбінаторної оптимізації, точність яких буде зростати зі збільшенням часу розрахунків; також указано на особливості генетичного алгоритму, які пов'язані з тим, що цей алгоритм саме і є комбінацією стандартних підходів до оптимізації. А саме — механізми схрещування та мутації, в певному розумінні, реалізують переборну частину методу, а відбір кращих розв'язків — градієнтний спуск. Таке сполучення забезпечує достатньо стійку ефективність генетичного пошуку для різної складності оптимізаційних задач.

У даній роботі для розв'язання поставленої задачі завдяки запропонованим операціям схрещування та мутації і процедурі елітного відбору, генетичний алгоритм дуже швидко виходить на один з кращих результатів, а саме для останньої задачі, коли $r = 3$ та $m = 100$, найкращий результат був отриманий через 10 хв. Зауважимо, що для проведення експерименту був використаний звичайний комп'ютер (Intel Celeron 1,3 GHz, 256 MB RAM).

Висновки

Дослідним шляхом, за допомогою обчислювальних експериментів, було встановлено, що запропонований генетичний алгоритм є достатньо потужним при розв'язанні задач про комівояжера у відповідній постановці. Основною перевагою запропонованого алгоритму є його спроможність розв'язувати задачі про комівояжера з великою розмірністю за припустимий час.

Подальші дослідження будуть пов'язані з використанням апарату нейромереж для розв'язання задач про комівояжера з розширенням постановки задач за рахунок введення певних обмежень на тип та кількість міст у маршрутах.

1. *Реклейтис А., Рейвиндран А. и др.* Оптимизация в технике. В 2-х кн. — М.: Мир, 1988. — 667 с.
2. *Кристофидес Н.* Теория графов. Алгоритмический подход. — М.: Мир, 1978. — 427 с.
3. *Круглов В.В., Борисов В.В.* Искусственные нейронные сети. Теория и практика. — М.: Горячая линия – Телеком, 2001. — 382 с.
4. *Наренков И.П.* Эвристики и их комбинирование в генетических методах дискретной оптимизации // Информационные технологии. — М., 1999. — № 1. — С. 2–7.
5. *Ротштейн О.П.* Інтелектуальні технології ідентифікації: нечіткі множини, генетичні алгоритми, нейронні мережі. — Вінниця: УНІВЕРСУМ–Вінниця, 1999. — 320 с.
6. Генетические алгоритмы, искусственные нейронные сети и проблемы виртуальной реальности / *Вороновский Г.К., Махотило К.В., Петрашев С.Н., Сергеев С.А.* — Харьков: Основа, 1997. — 212 с.
7. *Кондратенко Н.Р., Куземко С.М.* Оптимізація багато екстремальних функцій за допомогою одного генетичного алгоритму // Наукові вісті Національного технічного університету України «Київський політехнічний інститут». — 2002. — № 1. — С. 42–47.

Надійшла до редакції 30.05.2004