

Г. С. Іващенко, О. І. Онищенко, М. Е. Бондаренко, Н. В. Здорик

Харківський національний університет радіоелектроніки, Харків, Україна

МЕТОДИ РІШЕННЯ ЗАДАЧІ КОМІВОЯЖЕРА НА ОСНОВІ ОБЧИСЛЮВАЛЬНОГО ІНТЕЛЕКТУ

Анотація. **Актуальність.** На сьогоднішній день, задача комівояжера зберігає свою актуальність, оскільки потреба у пошуку найкоротших маршрутів наразі зустрічається в багатьох сферах людської діяльності. Із розвитком технологій та зростанням складності виробничих процесів та логістики, підвищуються вимоги до точності та швидкості пошуку, що зумовлює необхідність вибору доцільних методів рішення цієї задачі. **Метою даної роботи** є дослідження ефективності використання алгоритмів рішення задачі комівояжера на основі методів обчислювального інтелекту. **Об'єктом дослідження** є процес пошуку найкоротшого маршруту у графах великої розмірності. **Предметом дослідження** є алгоритми пошуку маршрутів для задачі комівояжера з використанням методів обчислювального інтелекту. **Результати.** У даній роботі розглядаються особливості застосування методів обчислювального інтелекту для вирішення задачі комівояжера, що полягають у використанні варіацій генетичного, мурашиного алгоритмів та штучних нейронних мереж, зокрема, карти Кохонена. Проведено аналіз збіжності генетичних алгоритмів при використанні різновидів генетичних операторів. Отримані результати проведених експериментальних досліджень дозволяють зробити висновки щодо переваг та недоліків окремих алгоритмів. **Висновок.** Запропонований підхід на основі штучної нейронної мережі демонструє найкращу швидкість та найбільшу точність при вирішенні задач розмірністю приблизно 1000 вершин. Для задач з меншим числом вершин, точніші результати забезпечує використання генетичного алгоритму.

Ключові слова: задача комівояжера, маршрут, граф, генетичний алгоритм, кросовер, мутація, селекція, мурашиний алгоритм, штучні нейронні мережі, карта Кохонена

Вступ

Із розвитком технологій, такі галузі людської діяльності, як транспорт і логістика, виробництво, телекомунікації, поступово стають все складнішими, і потребують нових ефективних методів зменшення витрат часу, матеріальних та інтелектуальних ресурсів. З цієї причини, зростає актуальність у більш ефективних методах рішень задачі комівояжера (ЗК) – однієї з найпоширеніших задач комбінаторної оптимізації теорії графів, направленої на пошук найкоротших маршрутів [1].

ЗК відноситься до класу NP-повних задач, тому зі збільшенням кількості вершин, варіативність можливих рішень зростає з факторіальною складністю, і пошук найкращого можливого рішення за прийнятний час унеможливується [2]. В такому випадку доцільним є застосування евристичних алгоритмів [3], завдяки їх здатності знаходити наближені до точних рішення в умовах високої обчислювальної складності задачі. Перспективним напрямком є використання тісно пов'язаних з евристичними алгоритмами методів обчислювального інтелекту, таких як штучні нейронні мережі, еволюційні підходи, ройовий інтелект тощо [4].

Одним з широко використовуваних підходів машинного навчання для пошуку рішень ЗК є генетичні алгоритми (ГА), які дозволяють пришвидшити пошук за рахунок розгляду обмеженої множини можливих рішень замість загального перебору великої кількості маршрутів, мають здатність ітеративно покращувати знайдені варіанти рішення шляхом модифікації наявних та відбору найкращих варіантів маршрутів. Широкі можливості модернізації зумовлюють вибір ГА предметом досліджень багатьох актуальних наукових робіт, зокрема пов'язаних з вирішенням ЗК. В [5, 6] встановлено, що комбінування різних

реалізацій операторів генетичного алгоритму (кросоверу, селекції, мутації, відбору) та конфігурування їх вхідних параметрів забезпечує високу ефективність вирішення завдань комбінаторної оптимізації, але залишається актуальною проблема швидкої (передчасної) збіжності алгоритму.

Іншим напрямком вирішення ЗК є використання мурашиного алгоритму, що являє собою модель багатоагентної системи, де результат роботи кожного агента (поточний шлях обходу графу) впливає на діяльність інших і таким чином поступово надається перевага найкращим маршрутам все більшій кількості агентів. Можливість визначити гетерогенність агентів та налаштувати їхню поведінку, зміни порядку оновлення феромону на маршрутах, а також підбір початкових умов, як запропоновано в [7], дозволяють знаходити кращі рішення ЗК.

Перспективним підходом для вирішення ЗК також є використання штучних нейронних мереж (ШНМ). Здатність ШНМ адаптуватися до складних структур даних зумовлює результативність такої архітектури, як самоорганізуюча карта Кохонена, у вирішенні ЗК. Використання мережі з топологією замкнутого кільця, описане в [8], дає змогу застосовувати алгоритм при пошуку маршрутів для ЗК, але проблема появи окремих відгалужень на маршруті у процесі роботи ШНМ перешкоджає знаходженню найкоротших шляхів.

В ході аналізу результатів існуючих досліджень, зроблено висновок, що результативність описаних алгоритмів наразі є недостатньою, та існують ще не задіяні напрямки вдосконалення поширених підходів, що потребують більш детального експериментального дослідження.

Метою цієї роботи є дослідження особливостей використання підходів машинного навчання, таких як генетичні алгоритми, мурашині алгоритми та

самоорганізуюча карта Кохонена для вирішення задачі комівояжера. Особлива увага має бути приділена дослідженню впливу параметрів налаштувань зазначених методів на ефективність пошуку найкоротшого шляху обходу графів великого розміру.

Постановка задачі

Математична модель задачі комівояжера визначається наступним чином: даний повнозв'язний зв'язаний граф, що має N вершин, та матриця $C = |c_{ij}|$ розмірності $N * N$, яка містить значення вартості (відстані) переходу з вершини i до вершини j . Варіант рішення ЗК – шлях обходу графу, який визначається як матриця $X = |x_{ij}|$, $x_{ij} \in \{0, 1\}$, де значення 1 означає наявність переходу з вершини i до вершини j у знайденому циклі, 0 задається в протилежному випадку. Найкращим рішенням є шлях, який має найменшу можливу вартість

$$\sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min, \quad (1)$$

і кожна вершина включена до маршруту тільки один раз, тобто має тільки одну вхідну, та тільки одну вихідну дуги:

$$\sum_{j=1, j \neq i}^n x_{ij} = \sum_{i=1, i \neq j}^n x_{ij} = 1, (i, j = \overline{1, n}). \quad (2)$$

Крім того, має існувати лише один шлях, що включає одразу всі вершини графа, що визначається умовою

$$u_i - u_j + n x_{ij} \leq n - 1, i, j = \overline{1, n}, i \neq j. \quad (3)$$

Для дослідження методів вирішення ЗК обрані набори вихідних даних, наявні у бібліотеці TSPLIB, яка надає набір готових екземплярів ЗК із заздалегідь визначеними найкоротшими маршрутами, що дозволяє порівнювати точність реалізованих алгоритмів. Для проведення експериментів обрані екземпляри симетричної ЗК, вершини яких задані в декартовій системі координат. Для дослідження результативності обраних у роботі методів вирішення на задачах різної розмірності вирішено розглядати ЗК на графах, що містять від 100 до 1000 вершин.

Рішення задачі комівояжера методами обчислювального інтелекту

Генетичні алгоритми (ГА) засновані на імітації природних еволюційних процесів та інтерпретації явищ генетики та оперують певними об'єктами – особинами, які відрізняються один від одного своїм генотипом (множиною характеристик, що описують рішення та визначають його пристосованість).

В ході дослідження сформульовано ряд умов, які дозволять знайти наближені рішення ЗК за допомогою ГА. Особинами вирішено вважати маршрути, що відносяться до множини можливих рішень ЗК, тобто такі, що відповідають умовам (2) і (3). Для кодування генів обрано шляхове представлення – генотипом буде вважатися перелік вершин маршруту у

порядку обходу, а за показник пристосованості приймається довжина шляху. Відповідно до (1), чим меншою є довжина маршруту – тип більш пристосованою буде вважатися особина.

При пошуці найбільш результативного варіанту ГА досліджуються комбінації різних реалізацій основних операцій – кросовера, мутації та селекції.

Серед кросоверів розглянуті упорядкований (Order Crossover, OX) [9], одноточковий порядковий кросовер (Single Point Ordered Crossover, SPOX), кросовер із частковим відображенням (Partially-Mapped Crossover, PMX) [10], циклічний кросовер (Cycle Crossover, CX) [11], реорганізуючий (Order-based Crossover, OBX) [12] та кросовер з інверсією за опорним елементом (Inver-Over Crossover, IOX) [13]. Одноточковий порядковий кросовер є модифікацією OX, в якій задається тільки індекс кінця фрагменту, а початок фрагменту збігається з початком генотипу.

Задіяні реалізації мутацій включають обмін двох фрагментів генотипу, зсув фрагменту та інверсію фрагменту маршрутів. Використання мутацій сприяє збереженню різноманіття у популяції та запобігає потраплянню у локальні оптимуми.

У якості операторів відбору особин для схрещування для створення нового покоління використовуються варіації турнірного методу та методу рулетки.

В залежності від принципу дії, операції ГА можуть мати власні параметри: номери позицій для визначення фрагментів у випадку кросоверів, параметри для мутацій (довжина фрагменту для обміну, позиція початку фрагменту, його довжина та кількість позицій для зсуву, початкова і кінцева позиції інтервалу для інверсії). З метою дослідження впливу вхідних параметрів операторів ГА на отримувані результати, реалізовані такі способи ініціалізації цих параметрів: «manual» (ручне внесення) – значення параметра задається користувачем один раз і більше не змінюється, «one time» – значення генерується випадково один раз і більше не змінюється, «every generation» – нове випадкове значення задається на кожне нове покоління, «every individual» – нове випадкове значення задається для обробки кожної особини (пари осіб у випадку кросовера) в межах одного покоління.

Однією з головних проблем при використанні ГА є передчасна збіжність, яка полягає у поступовому зменшенні різноманіття генотипу популяції протягом роботи алгоритму. Поява особин з однаковим генотипом значно гальмує чи зовсім зупиняє пошук нових, можливо, кращих рішень. Швидкість розвитку цього явища може залежати від розміру популяції, числа поколінь та виконання операцій ГА (зокрема, ймовірності та ступеню мутації). З метою аналізу та пошуку методів запобігання передчасної збіжності, запропоновано критерій оцінки – показник виродження популяції D , який вираховується як лінійна (4) або квадратична функція (5):

$$D = \frac{A-1}{N-1}, A = \overline{1, N}, D \in [0;1], \quad (4)$$

$$D = \frac{A^2-1}{N^2-1}, A = \overline{1, N}, D \in [0;1], \quad (5)$$

де A – кількість унікальних генотипів у популяції, N – загальний розмір популяції. Виродження складає 0, якщо $A = N$, тобто коли кожна особина у популяції має унікальний генотип, і збільшується до 1 зі зростанням кількості повторюваних рішень, що описуються варіантами маршрутів, представлених у особинах популяції (рис. 1).

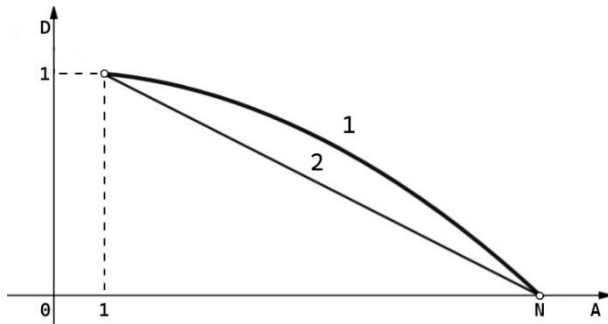


Рис. 1. Графіки квадратичної (1) та лінійної (2) функцій оцінки виродження популяції ГА

У реалізованих ГА передбачено першочергові умови зупину за відсутністю покращень рішень (якщо найкращий результат не змінюється протягом заданого числа поколінь), та за виродженням популяції (при досягненні показником виродження певного вказаного значення). Для обох варіантів також вказується максимальне число поколінь, після якого алгоритм припинить роботу, якщо першочергова умова зупинки не виконається до того часу.

В роботі також розглянуто використання мурашиного алгоритму (МА), в основі якого поведінка однійменних комах, які здатні ефективно знаходити найкоротші шляхи до джерел їжі при колективній взаємодії. Важливу роль в цьому процесі має позначення найкоротших маршрутів спеціальною речовиною – феромоном, концентрація якої впливає на вибір шляху іншими агентами (мурахами). В процесі формування рішення, агент прокладає шлях по переходах між вершинами, ймовірність вибору яких в класичному варіанті визначається як

$$p_{ij} = \frac{t_{ij}^{\alpha} + \frac{1}{w_{ij}^{\beta}}}{\sum_{l \in S_i} (t_{il}^{\alpha} + \frac{1}{w_{il}^{\beta}})}, \quad (6)$$

де i – поточна вершина, в якій розташований агент, j – вершина, що розглядається для вибору, S_i – множина доступних для вибору вершин, t – кількість феромону на переході, w – довжина переходу, α та β – регульовані параметри, що при виборі переходу визначаються пріоритетність феромону чи вартості переходу відповідно.

Враховуючи здатність феромону до випаровування, кількість речовини після проходження однією комахою залишається вищою на більш коротких шляхах, що зумовлює поступове виділення найкоротших шляхів більшістю особин колонії [14].

Дослідження спирається на реалізації МА на основі таких підходів, як алгоритм Ant System [15] і

виділення класу елітарних агентів для закріплення найкращих шляхів [16]. Наявність пам'яті в кожного агента, яка зберігає інформацію про вершини, пройдені протягом однієї ітерації, дозволяє визначати ще не відвідані вершини, що задовольняє умовам (1-3) та дозволяє використовувати розроблену реалізацію МА для вирішення ЗК.

Розглянуто використання штучних нейронних мереж (ШНМ), як широко розповсюдженого засобу для вирішення задач, що оперують складними структурами даних [17–20]. Серед архітектур ШНМ для вирішення задачі на графах найбільш доцільним є використання самоорганізуючої карти Кохонена (Self-Organizing Map – SOM), що використовує підхід «навчання без вчителя» – налаштування мережі без використання тренувальних даних, направлене на виявлення неочевидних взаємозв'язків та закономірностей даних [21].

При реалізації алгоритму побудови SOM, вжито ряд заходів, направлених на його адаптацію до вирішення ЗК. Топологія мережі обрана у вигляді замкнутого кільця, згідно якої кожен нейрон має зв'язок тільки з двома сусідніми [22]. Пошук зупиняється, коли кожна вершина буде мати принаймні один асоційований нейрон мережі. Таким нейроном вважається той, що є певною мірою наближений до вершини (відстань не перевищує вказаний поріг). При цьому, кожен нейрон не може бути асоційований більш ніж з одною вершиною ЗК. Якщо нейрон вже є наближеним до вершини, він виключається із процесу пошуку ВМУ (Best Matching Unit – найближчий нейрон до вектору даних, що розглядається протягом поточної ітерації), а вершина, відповідно, виключається з перебору в наступних ітераціях.

В якості функції сусідства, що визначає порядок зсуву нейронів залежно від їх розташування відносно ВМУ, обрана функція Гауса наступного вигляду:

$$h_{i,BMU} = e^{\frac{-D(\bar{n}_i, \bar{n}_{BMU})^2 * L(\bar{n}_i, \bar{n}_{BMU})}{\sigma}}, \quad (7)$$

де $L(\bar{n}_i, \bar{n}_{BMU})$ – евклідова відстань від нейрону \bar{n}_i до ВМУ, а $D(\bar{n}_i, \bar{n}_{BMU})$ – відстань між цими нейронами, що виражається в кількості переходів між ними по мережі. Використання відстані D дозволяє враховувати топологію мережі та при переміщенні нейронів віддавати більшу перевагу тим вузлам, що знаходяться ближче до ВМУ по порядку слідування в мережі, а не лише за відстанню на площині. Параметр σ визначає «еластичність» мережі – його значення прямо пропорційно впливає на кількість сусідніх з ВМУ нейронів, що змінять своє положення, і на відстань їх зсуву. В ході роботи алгоритму σ зменшується із темпами, які задаються окремим параметром $\Delta\sigma$, поступово зменшуючи значення функції сусідства, і роблячи кожен наступний змін у мережі більш локальною. Після обчислення функції Гауса, відбувається зсув нейронів в бік поточної вершини, що описується як

$$\bar{n}_i' = \bar{n}_i + \mu h_{i,BMU} (\bar{x} - \bar{n}_i), \quad (8)$$

де \bar{n}_i – нейрон, що переміщується, \bar{x} – вершина, в бік якої відбувається зміщення, $\mu \in (0;1]$ – коефіцієнт навчання, що вповільнює збіжність алгоритму, забезпечуючи точніші рішення.

На останніх ітераціях алгоритму роботи ШНМ, коли локальність змін є максимальною (відбувається пересування переважно тільки ВМУ), може виникнути явище, яке має негативний вплив на остаточний результат. Якщо один і той же нейрон обирається як ВМУ для двох чи більше вершин ЗК, його наближення до однієї вершини частково компенсується переміщенням в бік іншої, що призводить до уповільнення процесу асоціації. Для запобігання цьому явищу, вибір найближчого до вершини нейрона визначено як:

$$\bar{n}_j : L(\bar{n}_j, \bar{d}_i) * p_j \rightarrow \min, \quad (9)$$

де p_j є штрафним коефіцієнтом, що має завадити занадто частому вибору нейрону як найближчого та прискорити пересування в бік однієї вершини. Початкове значення коефіцієнту дорівнює 1 для всіх нейронів, і після кожного вибору елемента як ВМУ, його штрафний коефіцієнт збільшується на значення, що задається окремим параметром Δp .

Оскільки розподілення вершин ЗК у графі не є рівномірним, в процесі формування карти можуть залишатися нейрони, чиї всі сусіди вже асоційовані з вершинами, але які не мають поблизу доступних вершин для асоціації з ними. Алгоритм роботи карти Кохонена буде рухати такі нейрони до найближчих доступних вершин, які можуть знаходитися на віддаленій ділянці простору графу ЗК, що породжує відгалуження на фрагментах маршруту (рис. 2), які роблять кінцевий шлях обходу графа значно довшим.

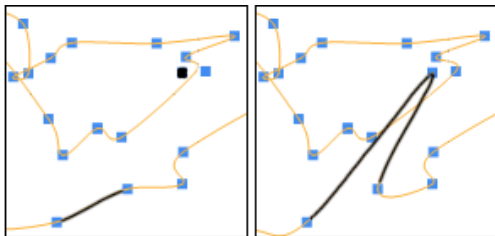


Рис. 2. Поява відгалужень при нерівномірному розподіленні точок мережі

З метою уникнення появи відгалужень вирішено формувати мережу зі збільшеною кількістю нейронів. Після виконання умови зупини, для отримання остаточного маршруту, нейрони, які не мають асоційованих вершин ЗК, видаляються з мережі. Доцільне значення коефіцієнта збільшення кількості нейронів N визначається експериментально для конкретного графа.

Результати експериментальних досліджень

Для проведення експериментів обрані графи tsb225, pcb442 та pr1002 (кількість вершин відповідає номеру у назві) з бібліотеки TSPLIB.

Дослідження використання ГА починаються з перевірки темпів збіжності алгоритму для різних

комбінацій генетичних операторів, із задіянням методів ініціалізації параметрів, що включають генерацію випадкових значень (One Time – «ОТ», Every Generation – «ЕГ», Every Individual – «ЕІ»).

Оскільки оператор мутації покликаний підвищувати різноманіття генотипу та зменшувати темпи збіжності, вирішено не використовувати його в даній серії експериментів, встановивши ймовірність мутації на 0.

Розмір популяції складає 250 особин, умовою зупини обрано досягнення коефіцієнтом виродження D величини 0,8, із максимально можливою кількістю поколінь 5000. Експерименти проведені для обох варіантів розрахунку показника виродження популяції (4-5).

В ході серії експериментів проведених для графа pr1002 (табл. 1), виявлено, що найменшу схильність до передчасної збіжності забезпечує кросовер Inver-over (IO) із методом ініціалізації початкових параметрів «ЕІ», який має приблизно однакові показники для обох варіантів розрахунку показника виродження D при використанні обох методів відбору – турнірної селекції (Т) та рулеткою (Р).

Таблиця 1 – Збіжність різновидів ГА

№ п/п	Кросовер	Селекція	Ініціалізація	Кільк. ітерацій	Виродження
Лінійна функція D					
1	IO	Т	ЕІ	5000	0,0120
2	IO	Р	ЕІ	5000	0,0201
3	IO	Т	ЕГ	5000	0,0375
4	IO	Р	ЕГ	5000	0,0924
5	OBX	Т	ЕІ	2536	0,8393
6	OBX	Т	ЕГ	2306	0,8755
7	IO	Т	ОТ	434	0,8568
Квадратична функція D					
1	IO	Р	ЕІ	5000	0,0186
2	IO	Т	ЕІ	5000	0,0343
3	IO	Р	ЕГ	5000	0,0396
4	IO	Т	ЕГ	5000	0,1505
5	OBX	Р	ЕІ	2548	0,9050
6	OBX	Т	ЕГ	2181	0,9048
7	IO	Р	ОТ	597	0,8318

Наступним етапом дослідження є порівняння швидкості роботи алгоритму та довжин знайдених маршрутів. З цією метою, для всіх варіантів ГА встановлено єдину умову зупини – досягнення 500 поколінь. Згідно результатам експериментів (рис.3) найкоротші шляхи знаходяться при селекції методом рулетки та кросовері ОХ з ініціалізацією способом ЕІ (3880514 на графі pr1002), наступним за довжиною знайденого маршруту є результат при IO із тим же типом селекції та порядком задання параметрів (4028028).

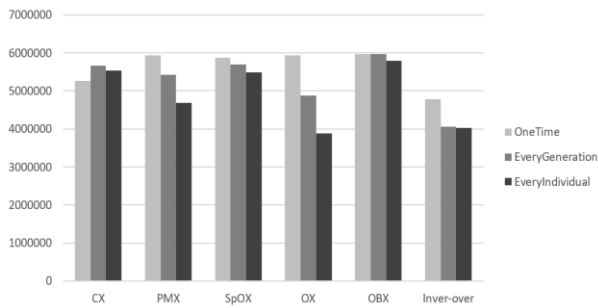


Рис. 3. Довжини маршрутів, знайдених ГА, для різних кросоверів та селекції методом рулетки

Але в той же час, конфігурація з кросовером ІО виявилася у 12 разів швидшою за варіант з ОХ (рис. 4) – 26 та 321 секунда відповідно, що зумовлює вибір цього варіанту генетичного алгоритму для подальших експериментів.

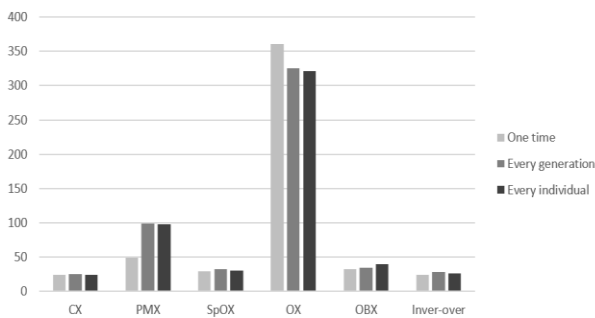


Рис. 4. Час роботи ГА для різних кросоверів та селекції методом рулетки

Проведені експерименти також показали, що обрана конфігурація (кросовер ІО, ініціалізація параметрів для кожної особи окремо на кожному поколінні, стратегія відбору рулеткою) відрізняється стійкістю до передчасної збіжності, що дає змогу не використовувати оператор мутації для підвищення різноманітності генотипу і дозволяє тим самим зменшити час роботи алгоритму.

Проведена серія експериментів, в ході якої визначався необхідний розмір популяції та число поколінь для отримання якомога коротших маршрутів за прийнятний час. Результати використання ГА для графів різного розміру представлені у зведених даних за усіма розглянутими методами у табл. 3.

При дослідженні мурашиного алгоритму, важливим аспектом є підбір параметрів α та β , тому дослідження роботи МА починаються з встановлення їх найкращого співвідношення. Початкові умови проведення експериментів: кількість агентів – 25, кількість одиниць феромону, що агент залишає на ділянці – 100, кількість ітерацій – 25, коефіцієнт випаровування – 50%, α змінюється від 0,5 до 5 з кроком 0,5, β – від 1 до 5 з кроком 1. Результати експерименту, представлені на рис. 5, показують, що кращі результати досягаються при α від 1,5 та вище, та β від 2 та вище. Найкоротший маршрут на графі pr1002 (365244) отриманий при $\alpha=2,5$ та $\beta=5$. Згодом, цей результат було покращено (361885) при збільшенні кількості агентів та ітерацій до 35.

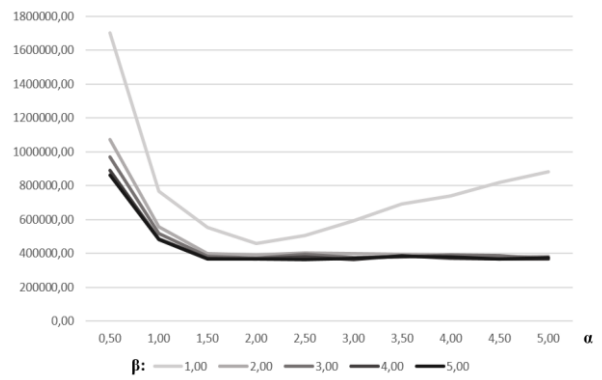


Рис. 5. Впливу параметрів α та β на довжину маршруту, знайденого мурашиним алгоритмом

Проведено дослідження доцільності застосування концепції елітних агентів. Частина агентів наданий відповідний клас, що обумовлює кількість феромону, що додається ними (змінюється від 125 до 350 одиниць). Згідно рис. 6, найкращий результат отримано при кількості феромону 225 одиниць. Надалі це значення використовувалося для отримання шляхів з мінімальною довжиною (табл. 3).

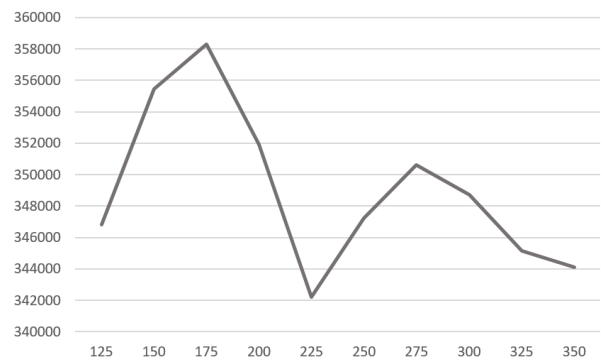


Рис. 6. Залежність довжини мінімального маршруту від кількості феромону, що залишають елітні агенти

При розгляданні самоорганізуючої карти Кохонена для пошуку найкоротших шляхів на графах, налаштування мережі необхідно проводити окремо для кожного з представлених графів. Не виявлено закономірності щодо отриманого результату та значень вхідних параметрів мережі для графів різної розмірності. Суттєвий вплив на довжину знайденого маршруту обходу на графі pr1002 помічено для штрафного коефіцієнту Δp (рис. 7).

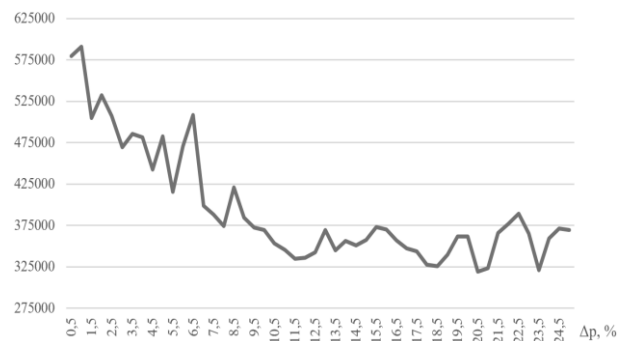


Рис. 7. Залежність довжини маршруту від значення штрафного коефіцієнту

Після серій експериментів щодо підбору параметрів (Δp , множник розміру мережі N та коефіцієнт швидкості навчання μ), були визначені їх значення для отримання найкращих результатів, представлених в табл. 2.

Таблиця 2 – Значення параметрів Δp , N , μ для обраних досліджень графів

Граф	Δp , %	μ , %
tsp225	18	35
pcb442	7,5	35
pr1002	20,5	60

Значення інших параметрів роботи штучної нейронної мережі для пошуку найкоротшого шляху обходу графу обрані наступні: множник розміру мережі дорівнює 3, еластичність мережі $\sigma = N2$, де N – кількість вершин на графі, темп зменшення еластичності мережі $\Delta\sigma$ дорівнює 0,1%.

В табл. 3 представлені зведені результати досліджень для всіх розглянутих алгоритмів та трьох графів ЗК, де L_{best} – довжина відомого найкращого маршруту, що надається авторами TSPLIB, L позначає довжину найкоротшого шляху, знайденого відповідним підходом, t – загальний час роботи алгоритму в секундах.

Таблиця 3 – Зведені результати пошуку найкоротших шляхів методами на основі підходів машинного навчання

Граф	L_{best}	ГА		МА				SOM	
		L	t, c	Ant System		Elitist Algorithm		L	t, c
				L	t, c	L	t, c		
tsp225	3919	4252	9	4968	8	4754	6	5514	3
pcb442	50778	58076	68	66595	32	66646	23	60681	8
pr1002	259045	475204	194	361885	144	349750	127	309255	24

Результати дослідження, зібрані в табл. 3, показали, що найкращі результати для графів розміром у 225 та 442 вершини забезпечує генетичний алгоритм (4252 та 58076 відповідно), але довжина знайденого шляху для графу на 1002 вершини поступається результатам роботи МА та карті Кохонена, причому найкращий результат має ШНМ – 309255 при відомій оптимальній довжині в 259045.

Виявлено, що карта Кохонена дозволяє знайти наближені рішення значно швидше за інші розглянуті алгоритми, при чому ця різниця зростає із збільшенням кількості вершин графу – SOM швидший за інші алгоритми у 2-3 рази для задачі на 225 вершин, у 3-8 разів для 442 та у 5-7 разів для 1002 вершин. Подібне спостереження дозволяє зробити висновок про доцільність використання даного алгоритму для вирішення ЗК високих розмірностей.

Висновки

В даній роботі розглянуті підходи до вирішення задачі комівояжера з використанням методів обчислювального інтелекту, такі як генетичний алгоритм, мурашиний алгоритм та штучна нейронна мережа на прикладі самоорганізуючої карти Кохонена. Найбільшу точність рішень для задач розмірністю у 225 та 442 вершини показав генетичний алгоритм, а для задачі на 1002 міста найкоротший шлях був знайдений за допомогою ШНМ.

Також мережа Кохонена виявилася найшвидшим методом пошуку наближених рішень, що має важливе значення при практичному використанні задачі комівояжера.

Проведений аналіз збіжності генетичного алгоритму при використанні різних реалізацій генетичних операторів та способів їх налаштувань, найкращі результати отримано для конфігурації із кросовером типу Inver-Over, та методом ініціалізації його параметрів, що включає встановлення нових випадкових значень для кожної особи популяції на кожному поколінні.

Для подальших досліджень рішення ЗК на основі підходів машинного навчання є доцільним розгляд наступних питань:

- пошук швидких автоматизованих методів підбору початкових значень параметрів для самоорганізуючих карт Кохонена;
- можливість використання інших архітектур ШНМ для вирішення ЗК;
- аналіз більшої кількості існуючих та розробка нових модифікацій мурашиного алгоритму;
- дослідження нових реалізацій генетичних операторів для ГА.

Проведені дослідження показують ефективність використання підходів на основі засобів машинного навчання для знаходження прийнятних маршрутів обходу графів за короткий час.

СПИСОК ЛІТЕРАТУРИ

1. Davendra D. (2010), "Travelling Salesman Problem, Theory and Applications", InTech, P. 338, ISBN 978-953-307-426-9
2. Dahiya C., Sangwan S. (2018), "Literature Review on Travelling Salesman Problem", International Journal of Research, Vol. 5(16), pp. 1152-1155, e-ISSN: 2348-6848, p-ISSN: 2348-795X
3. Toaza B, Esztergár-Kiss D. (2023), "A review of metaheuristic algorithms for solving TSP-based scheduling optimization problems", Applied Soft Computing, Vol. 148, pp. 1-24, doi: <https://doi.org/10.1016/j.asoc.2023.110908>
4. Crişan G. C., Iantovics L. B., Nechita E. (2019), "Computational Intelligence for Solving Difficult Transportation Problems", Procedia Computer Science, Vol. 159, pp. 172-181, doi: <https://doi.org/10.1016/j.procs.2019.09.172>
5. Eskandari S., Rafsanjani M. K. (2023), "Two new selection methods and their effects on the performance of genetic algorithm

- in solving supply chain and travelling salesman problems”, International Journal of Bio-Inspired Computation (IJBIC), Vol. 22(3), pp. 176-184, doi: <https://doi.org/10.1504/IJBIC.2023.135464>
6. Іващенко Г. С., Склярів А. С., Барковська О. Ю. (2023), “Гібридний метод рішення задачі маршрутизації транспорту з урахуванням додаткових обмежень”, Системи управління, навігації та зв'язку, №1, с. 75-79, doi: <https://doi.org/10.26906/SUNZ.2023.1.075>
 7. Wang, Y., Han, Z. (2021), “Ant colony optimization for traveling salesman problem based on parameters optimization”, Applied Soft Computing, Vol. 107, pp. 1-11, doi: <https://doi.org/10.1016/j.asoc.2021.107439>
 8. Zhang J., Feng X., Zhou B., Ren D. (2012), “An overall-regional competitive self-organizing map neural network for the Euclidean traveling salesman problem”, Neurocomputing, Vol. 89, pp. 1-11, doi: <https://doi.org/10.1016/j.neucom.2011.11.024>
 9. Shafie M. F., Ahmad F., Osman M. K., Ismail A.P., Ahmad K. A., Yahaya S. Z. (2023), “Optimization of Saleman Travelling Problem Using Genetic Algorithm with Combination of Order and Random Crossover”, 2023 IEEE 13th International Conference on Control System, Computing and Engineering (ICCSCE), pp. 255-258, doi: <https://doi.org/10.1109/ICCSCE58721.2023.10237137>
 10. Deep K., Mebrahtu H. (2012), “Variant of partially mapped crossover for the Travelling Salesman problems”, International Journal of Combinatorial Optimization Problems and Informatics, Vol. 3(1), pp. 47-69, ISSN: 2007-1558
 11. OuYang Q., Xu H. (2015), “The study of comparisons of three crossover operators in genetic algorithm for solving single machine scheduling problem”, International Conference on Manufacturing Science and Engineering (ICMSE 2015), Atlantis Press, pp. 293-297, doi: <https://doi.org/10.2991/icmse-15.2015.55>
 12. Kumar R., Memoria M., Thapliyal M., Kirola M., Ahmad I., Gupta A., Tyagi S., Ansari N. (2022), “Analyzing The Performance Of Crossover Operators (OX, OBX, PBX, MPX) To Solve Combinatorial Problems”, 2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON), pp. 817-821, doi: <https://doi.org/10.1109/COM-IT-CON54601.2022.9850689>
 13. Singh D. R., Singh M. K., Singh T. (2016), “A Hybrid Algorithm with Modified Inver-Over Operator and Genetic Algorithm Search for Traveling Salesman Problem”, Advanced Computing and Communication Technologies, pp. 141-150, doi: https://doi.org/10.1007/978-981-10-1023-1_14
 14. Chopard B., Tomassini M. (2018), “The Ant Colony Method”, An Introduction to Metaheuristics for Optimisation, pp. 81-96, doi: https://doi.org/10.1007/978-3-319-93073-2_5
 15. Colomi A., Dorigo M., Maniezzo V. (1991), “Distributed Optimization by Ant Colonies”, Proceedings of ECAL91 – European Conference on Artificial Life, Elsevier Publishing, pp. 134-142
 16. Dorigo M., Maniezzo V., Colomi A. (1996), “The Ant System: Optimization by a colony of cooperating agents”, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), Vol. 26(1), pp. 29-41, doi: <https://doi.org/10.1109/3477.484436>
 17. Yaloveha, V., Hlavcheva, D., Podorozhniak, A. and Kuchuk, H. (2019), “Fire hazard research of forest areas based on the use of convolutional and capsule neural networks”, 2019 IEEE 2nd Ukraine Conference on Electrical and Computer Engineering, UKRCON 2019 – Proceedings, DOI: <http://dx.doi.org/10.1109/UKRCON.2019.8879867>
 18. Yaloveha, V., Hlavcheva, D. and Podorozhniak, A. (2019), “Usage of convolutional neural network for multispectral image processing applied to the problem of detecting fire hazardous forest areas”, *Сучасні інформаційні системи*, Vol. 3, No 1, pp. 116–120, DOI: <https://doi.org/10.20998/2522-9052.2019.1.19>
 19. Datsenko, S., and Kuchuk, H. (2023), “Biometric authentication utilizing convolutional neural networks”, Advanced Information Systems, vol. 7, no. 2, pp. 67–73. Doi: <https://doi.org/10.20998/2522-9052.2023.3.10>
 20. Yaloveha, V., Podorozhniak, A., Kuchuk, H. (2022), “Convolutional neural network hyperparameter optimization applied to land cover classification”, *Radioelectronic and Computer Systems*, No. 1(2022), pp. 115–128, DOI: <https://doi.org/10.32620/reks.2022.1.09>
 21. Sarikiyriakidis S., Goulianas K., Margaris A. I. (2023), “Using Self-organizing Maps to Solve the Travelling Salesman Problem: A Review”, WSEAS transactions on systems, Vol. 22, pp. 131-159, doi: <https://doi.org/10.37394/23202.2023.22.14>
 22. Faigl J., Hollinger, G. A. (2018), “Autonomous Data Collection Using a Self-Organizing Map”, IEEE Transactions on Neural Networks and Learning Systems, Vol. 29(5), pp. 1703-1715, doi: <https://doi.org/10.1109/TNNLS.2017.2678482>

Received (Надійшла) 07.02.2024

Accepted for publication (Прийнята до друку) 24.04.2024

Methods of solving the traveling salesman problem based on computational intelligence

Heorhii Ivashchenko, Oleksandr Onyshchenko, Maksym Bondarenko, Nikita Zdoryk

Abstract. Topicality. Today, the traveling salesman's problem remains actual, as the need to find the shortest routes is currently found in many areas of human activity. With the development of technologies and the increase in the complexity of production processes and logistics, the requirements for accuracy and speed of search are increasing, which makes it necessary to choose appropriate methods for solving this problem. **The goal of this work** is to study the effectiveness of algorithms for solving the traveling salesman problem based on computational intelligence methods. **The object of research** is the process of finding the shortest route in large-dimensional TSP. **The subject of research** is route search algorithms for the traveling salesman problem using methods of computational intelligence, in particular, the analysis of the convergence of genetic algorithms when using different types of genetic operators. **Results.** This paper examines the features of the application of computational intelligence methods for solving the salesman's problem, which involves the use of variations of genetic algorithms, ant algorithms and artificial neural networks, in particular, Kohonen's self-organizing map. The obtained results of the performed experimental studies allow to draw conclusions about the advantages and disadvantages of individual algorithms. **Conclusions.** The proposed algorithm based on an artificial neural network has the highest speed, as well as the highest accuracy when problem graph contains approximately 1000 vertices. For problems with a smaller number of vertices, the genetic algorithm provides more accurate results.

Keywords: traveling salesman problem, route, graph, genetic algorithm, crossover, mutation, selection, ant algorithm, artificial neural networks, Kohonen's self-organizing map