МОДЕЛЮВАННЯ, ІМІТАЦІЯ ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ В ЕКОНОМІЦІ Й УПРАВЛІННІ

УДК 656.073+339.14 **DOI: 10.26565/2311-2379-2020-98-12**

О.В. Масленнікова
доцент
Харківський національний університет імені В.Н. Каразіна
майдан Свободи, 4, м.Харків, 61022, Україна
E-mail: 21041950EM@gmail.com, ORCID: https://orcid.org/0000-0002-7870-5743
Д.О. Осташев
студент
Харківський національний університет імені В.Н. Каразіна
майдан Свободи, 4, м.Харків, 61022, Україна
E-mail: ostashev@tutanota.com

ГЕНЕТИЧНИЙ АЛГОРИТМ ДЛЯ РОЗПОДІЛУ МАРШРУТІВ ДОСТАВКИ

Інформаційні технології стали невід'ємною частиною життя сучасного суспільства. Саме тому їх активно застосовують для вирішення складаних завдань, що пов'язані з оптимізацією діяльності в економіці, менеджменті, фінансах, соціальній та інших сферах. Логістика не стала винятком, а інформаційні технології використовують для автоматизації процесу доставки, що сприяє ефективній та рентабельній роботі підприємств. Це можливо за умови раціонального використання ресурсів постачальника та забезпечення його ефективним способом розподілу маршрутів доставки. Саме цим пояснюється актуальність даної роботи. Метою статті є розробка генетичного алгоритму для визначення оптимального плану доставки, а також порівняння його роботи з деякими відомими методами оптимізації доставки. Об'єктом дослідження є маршрути доставок. Предмет дослідження – алгоритм розподілу маршрутів доставки. В статті запропоновано генетичний алгоритм для розв'язку задачі комівояжера з декількома маршрутами. Представлено постановку задачі визначення оптимального плану доставки, розглянуто можливість застосування генетичного алгоритму для поставленої задачі, описано вид генетичного алгоритму та представлено блок-схему його реалізації. Підготовлено програмну реалізацію генетичного алгоритму та інших методів розподілу маршрутів доставки, що здійснено за допомогою мови програмування *C++* та фреймворку *Qt*. Для роботи алгоритмів потрібно згенерувати замовлення, транспортні засоби та депо, а також увести необхідні параметри алгоритмів. Отримано результати роботи алгоритму та інших обраних методів за допомогою програми, що дозволило оцінити їх ефективність на основі експериментальних даних при заданих параметрах та порівнювати їх роботу. Експериментально встановлено перевагу запропонованого генетичного алгоритму над відомим мурашиним на випадково згенерованих тестах. Це свідчить про прикладну цінність отриманих результатів та можливості застосування алгоритму у реальних умовах розподілу маршрутів доставок.

Ключові слова: розподіл маршрутів доставки, генетичній алгоритм, мурашиний алгоритм.
JEL Classification: R40, C02, C61, C63.

Постановка проблеми. У сучасному суспільстві інформаційні технології знайшли широке застосування в усіх сферах життя. Вони допомагають у вирішенні багатьох складних завдань, що мають соціальну, економічну, фінансову або управлінську направленість. Задача оптимізації бізнес-процесів у логістичній сфері за допомогою інформаційних технологій є актуальною. Зокрема, їх застосування для автоматизації процесу доставки, що впливає на ефективність роботи підприємства та отримання ним прибутку. Для раціонального використання ресурсів постачальника необхідно забезпечити його ефективним способом розподілу маршрутів доставки. Ефективний план доставок повинен відповідати загальній політиці компанії-постачальника та мати певний баланс між затратами на доставку та часом очікування замовлень.

© Масленнікова О.В., Осташев Д.О., 2020

Аналіз останніх досліджень і публікацій. Проблема диспетчеризації вантажівки є актуальною для дослідників та знайшла своє відображення в наукових статтях, зокрема в працях (Dantzig & Ramser, 1959), (Stodola & Mazal, 2015), (Chitty et al., 2019). Задача розподілення маршрутів між водіями полягає в складанні планів перевезення вантажів із декількох депо до пунктів призначень. При цьому цільова функція направлена на максимізацію вигоди постачальника. Функція вигоди у нашій моделі буде залежати від таких параметрів, як сумарна відстань, пройдена транспортними засобами компанії-постачальника, та середнього часу очікування замовлень клієнтом.

Мета статті та завдання. Метою статті є розробка генетичного алгоритму для визначення оптимального плану доставки, а також порівняння його роботи з деякими відомими методами оптимізації доставки.

Для досягнення даної мети необхідно виконати такі завдання: представити постановку задачі розподілу маршрутів доставки; розглянути можливість застосування генетичного алгоритму для поставленої задачі; охарактеризувати вид генетичного алгоритму, що буде використовуватися; представити блок-схему роботи алгоритму; розробити програму для демонстрації роботи алгоритмів та деяких відомих методів оптимізації доставки, що дозволяє експериментально оцінювати їх можливості та ефективність.

Основні результати дослідження. У процесі виконання роботи використовуються основні положення теорії алгоритмів та теорії графів. Алгоритми та система демонстрації реалізовані за допомогою мови програмування *C++* та фреймворку *Qt*.

Постановка задачі. Представимо нашу задачу в термінах теорії графів (Diestel, 2005). Дано неорієнтований зважений граф. У певних його вершинах розміщуються депо, а в деяких знаходяться транспортні засоби, є вершини, відмічені як пункти призначення.

Позначимо W_i – *i*-е депо, O_i – *i*-е замовлення, D_i – *i*-ий транспортний засіб. Нехай $N_{warehouses}$ – кількість депо, $N_{drivers}$ – кількість транспортний засобів та N_{orders} – кількість пунктів призначення. Будемо вважати, що в усіх депо достатньо товару для того, щоб задовольнити всі замовлення.

Планом перевезення назвемо сукупність маршрутів транспортних засобів. Кожен маршрут є шляхом у графі, що вказаний у формулі (1).

$$D_i \to W_{warehouse_i} \to O_{ord_{i,0}} \to O_{ord_{i,1}} \to \cdots \to O_{ord_{i,cou_i-1}} \quad (1)$$

Відзначимо, що кожний маршрут починається з положення якогось транспортного засобу та спочатку проходить через депо. Після цього немає сенсу відвідувати інші депо. Закінчується маршрут у пункті призначення. Тоді маршрут кожного використаного транспортного засобу будемо задавати на депо, через яке він проходить, та упорядкованим списком пунктів призначення.

Нехай cou_i – кількість замовлень *i*-го транспортного засобу. Для кожного маршруту визначимо $cost_i$ – вартість його проходження та $sumtime_i$ – сумарний час виконання замовлень на цьому маршруті (формула 2).

$$sumtime_i = time_{ord_{i,0}} + time_{ord_{i,1}} + \cdots + time_{o_{i,cou_i-1}} \quad (2)$$

де $time_{ord_{i,j}}$ – час виконання *j*-го замовлення транспортного засобу *i*.

По всім можливим планам перевезення *P* будемо оптимізувати наступну функцію (формула 3).

$$f(P) = \alpha * C + \beta * T; \quad \alpha + \beta = 1 \quad (3)$$

де C – сумарні кошти, витрачені на перевезення вантажів, T – середній час очікування замовлень, *α, β* – заздалегідь відомі коефіцієнти. Ці величини визначаються за формулами (4) та (5) відповідно.

$$C = \sum cost_i \quad (4)$$

$$T = \frac{1}{N_{orders}} \sum sumtime_i \quad (5)$$

Задача, що розглядається, є узагальненням задачі комівояжера (Travelling salesman problem, TSP) на випадок декількох маршрутів. Також оптимізаційна функція тепер

складається не тільки із сумарної відстані. Деякі алгоритми розв'язання такого типу задач показані в праці (Cormen et al., 2009).

Відзначимо, що при $N_{drivers} = N_{warehouses} = 1, D_0 = W_0$ та $\alpha = 1,\ \beta = 0$ задача стає еквівалентною до TSP. Таким чином, у загальному вигляді представлена задача належить до класу NP-складних, що розглянуті в праці (Knuth, 1974).

У зв'язку з тим, що задача належить до класу NP-складних, на момент написання даної роботи не знайдено алгоритмів, що знаходять точне рішення. Однак ця задача є прикладом задачі цілочислового лінійного програмування (ЦЛП). (Наконечний & Савіна, 2003) Для задач ЦЛП існує декілька наближених алгоритмів, найефективнішими серед яких є: генетичний алгоритм; пошук із заборонами; алгоритм імітації відпалу; мурашиний алгоритм (ACO). Точне рішення даного типу задач можна знайти за допомогою алгоритмів з експоненціальним часом, а саме: пошук грубою силою; метод гілок і меж; динамічне програмування.

Генетичний алгоритм – евристичний, натхненний процесом природного відбору алгоритм, що належить до більшого класу еволюційних алгоритмів. Генетичні алгоритми зазвичай використовуються для створення високоякісних рішень для оптимізації та пошуку завдань, спираючись на біологічні оператори, такі як мутація, схрещення та вибір.

Генетичний алгоритм є однією з евристик, яка часто використовувалася для пошуку оптимальних рішень багатьох комбінаційних проблем. У генетичному алгоритмі популяція рішень кандидатів (званих індивідів, істот або фенотипів) до оптимізаційної проблеми еволюціонує до кращих рішень. Кожне рішення-кандидат має набір властивостей (його хромосоми або генотип), які можуть мутувати. Генетичний алгоритм широко використовується як потужний інструмент для вирішення різних задач оптимізації за різних умов, що показано в працях (Eiben et al., 1994), (Mitchell, 1995), (Mitchell, 1996), (Koza, 1994) та ін.

Завдання, що постало перед нами, можна ефективно виконати за допомогою генетичного алгоритму. Опишемо, який саме вид генетичного алгоритму будемо використовувати.

1) Представлення хромосом. Хромосоми для рішення задачі будемо представляти у вигляді депо, у яке заходить транспортний засіб, та списку пунктів призначення в порядку їх об'їзду. У геномі зберігається $N_{drivers}$ записів виду {номер депо, список замовлень у порядку об'їзду}, тобто за записом для кожного транспортного засобу. Якщо засіб не буде використано, номер депо зробимо таким, що дорівнює – 1. Наприклад, розподіл маршрутів на рис.1 (депо позначені трикутниками, транспортні засоби – прямокутниками, пункти призначення – кружечками) відповідає наступному геному:

$$\{\{1,\{3\}\},\{1,\{2,5,1\}\},\{2,\{10,4,9,8\}\},\{3,\{7,6\}\}\}.$$

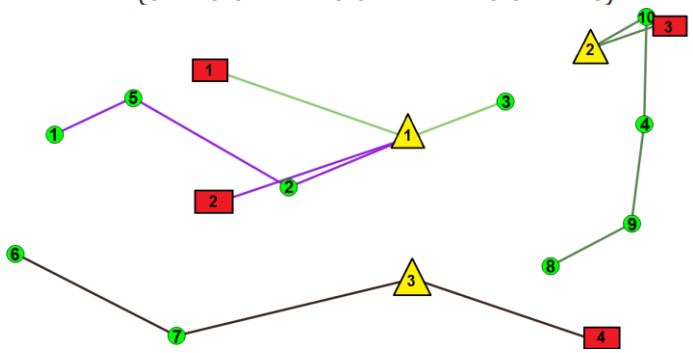

Рис.1. Маршрути доставки

Джерело: авторська розробка

2) Генерація початкової популяції. Початкова популяція з N хромосом буде згенеровано жадібним алгоритмом:

1. *N* раз виконуємо пункти 2-4:
2. Обираємо випадкове не розподілене замовлення:

2.1. Якщо локально вигідніше додати це замовлення в кінець якогось маршруту, то додаємо, після чого обчислюємо нове оптимальне депо для цього маршруту.

2.2. Якщо локально вигідніше розпочати новий маршрут із цього замовлення та є вільні транспортні засоби, розпочинаємо новий маршрут.

3. Якщо ще залишились не розподілені замовлення, переходимо до пункту 2.

4. Додаємо отриманий план доставок до популяції.

3) Обчислення фітнес-функції. Генетичний алгоритм мінімізуватиме значення функції, наведеної в першому розділі. Розглянемо, як саме будемо обчислювати значення фітнес-функції. Сумарні кошти, витрачені на доставку – це сума коштів за всіма транспортними засобами (формула 6).

$$C = \sum_{i=0}^{N_{drivers}-1} cost_i \tag{6}$$

Нехай *i*-ий транспортний засіб рухається за маршрутом, зазначеним у формулі (1). Тоді кошти, витрачені ним, пропорційні сумарній пройденій відстані (формула 7).

$$cost_i = \left(\begin{array}{c} dist(D_i, W_{warehouse_i}) + dist(W_{warehouse_i}, O_{ord_{i,0}}) + \\ + \sum_{k=0}^{cou_i-1} dist(O_{ord_{i,k}}, O_{ord_{i,k+1}}) \end{array} \right) * rate_i \tag{7}$$

де $dist(A,B)$ значить відстань між точкам A і B, $rate_i$ – кошти, які витрачає *i*-ий транспортний засіб на проїзд одиничної відстані.

Середній час доставки заказів можна обчислити за формулою (8).

$$T = \frac{1}{N_{orders}} \sum_{i=0}^{N_{orders}-1} time_i = \frac{1}{N_{orders}} \sum_{i=0}^{N_{drivers}-1} sumtime_i \tag{8}$$

де $time_i$ – час доставки і-го замовлення, $sumtime_i$ – сума часів доставки всіх замовлень, що везе *i*-ий транспортний засіб.

Сумарний час доставки всіх замовлень, що належать до *i*-го транспортного засобу, може бути обчислений за формулою (9).

$$\begin{aligned} sumtime_i &= \frac{1}{speed_i} \sum_{i=0}^{cou_i-1} \left(\begin{array}{c} dist(D_i, W_{warehouse_i}) + \\ + \sum_{k=0}^{i-1} dist(O_{ord_{i,k}}, O_{ord_{i,k+1}}) \end{array} \right) = \\ &= \frac{1}{speed_i} (cou_i * dist(D_i, W_{warehouse_i}) \\ &\quad + \sum_{k=1}^{cou_i-1} (cou_i - k) * dist(O_{ord_{i,k-1}}, O_{ord_{i,k}})) \end{aligned} \tag{9}$$

де $speed_i$ – швидкість і-го транспортного засобу, cou_i – кількість замовлень, що везе і-ий транспортний засіб.

Фінальне значення фітнес-функції, відповідно до математичної моделі, обчислюється за формулою (10).

$$fitness = \alpha * C + \beta * T \tag{10}$$

4) Відбір. У нашому алгоритмі буде використаний так званий турнірний відбір. Якщо популяція складається з N осіб. Тоді будемо *t* раз робити наступну операцію:

1. Обрати випадкові *t* осіб із поточної популяції.
2. Вибрати особу з найбільшим значенням фітнес-функції.
3. Додати вибрану особу до набору для схрещення.
4. Повернути залишені $t-1$ особу до популяції.

5) Схрещення. Обираються дві хромосоми з набору для схрещення. Потім відбуваються наступні операції:

1. Обрати випадковий маршрут з першої хромосоми $route_1$.
2. Обрати випадковий суфікси цього маршруту $suff_1$ таким чином, що $\frac{1}{3}|route_1| \leq |suff_1| \leq \frac{2}{3}|route_1|$.
3. Видалити з другої хромосоми всі замовлення, що належать до $suff_1$.

4. Вставити $suff_1$ у другу хромосому в найкращу позицію.
5. Для отриманої хромосоми в змінених маршрутах обчислити нові оптимальні депо.
6. Додати нову хромосому до популяції.

Блок-схему роботи алгоритму можна побачити на рис. 2.

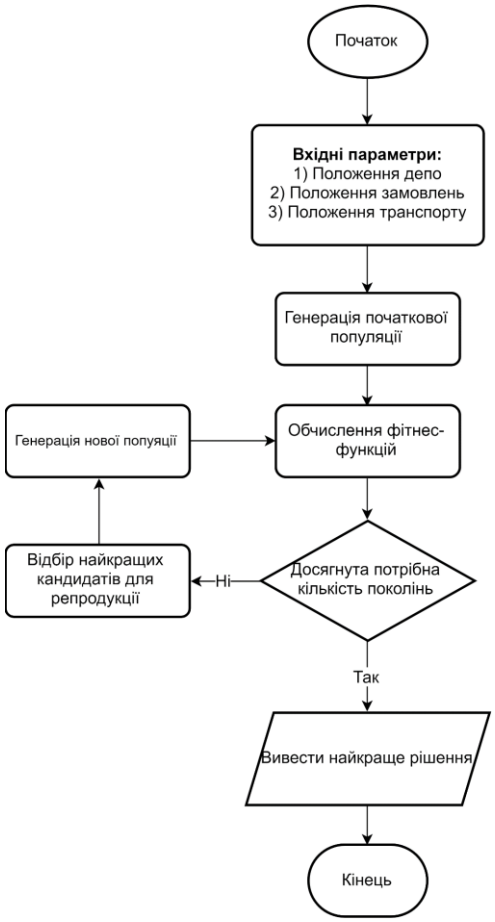

Рис.2. Блок-схема роботи генетичного алгоритму

Джерело: авторська розробка

Алгоритми та система демонстрації реалізовані за допомогою мови програмування C++ та фреймворку *Qt*. Можливість застосування цих засобів представлена в праці (Steeb, 2001), де показано їх потужність для такого типу завдань. А мова програмування C++ є однією з найшвидших завдяки потужним компіляторам.

Для проведення експерименту щодо роботи алгоритмів потрібно згенерувати замовлення, транспортні засоби та депо, а також увести необхідні параметри алгоритмів у відповідні поля. У залежності від значень введених параметрів, можливість використання певних алгоритмів може бути заблоковано через обчислювальну складність. При цьому кнопки запуску цих алгоритмів будуть неактивними. Для здійснення запуску алгоритму треба натиснути на відповідну кнопку. Під час його роботи деякі елементи інтерфейсу будуть неактивними.

Після того, як алгоритм завершить свою роботу, на екрані відобразиться знайдений ним план доставок та обчислені значення сумарної відстані (C) і середнього часу очікування замовлення (T) (рис. 3). Замовлення позначені кружечками, депо – трикутниками, а транспортні засоби – прямокутники.

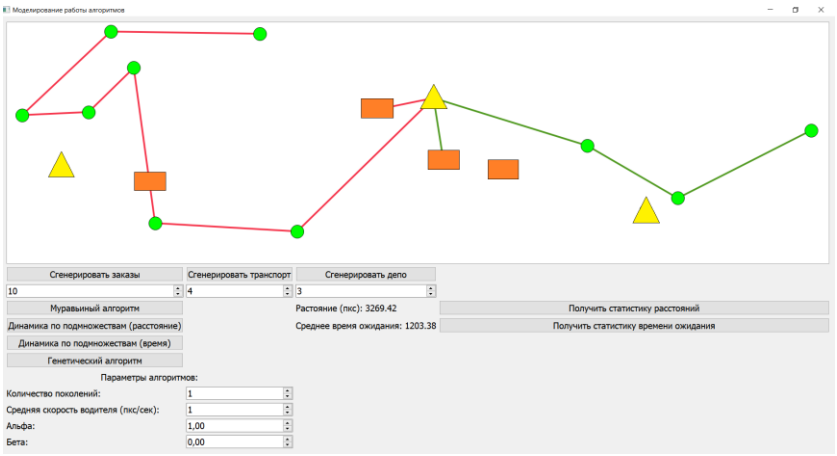

Рис.3. Відображений план доставок

Джерело: авторська розробка в програмі

Діаграми, на яких можна переглянути відстані, що проїдуть транспортні засоби та час очікування для кожного замовлення показано на рис. 4 та рис. 5 відповідно.

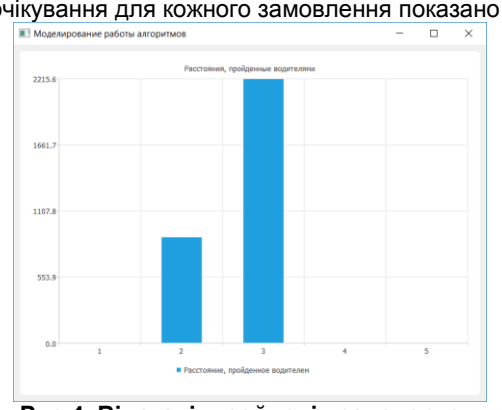

Рис.4. Відстані, пройдені транспортом

Рис.5. Час очікування замовлень

Джерело: авторські розрахунки в програмі

Для порівняння алгоритмів будемо використовувати декілька різних наборів вхідних даних. Властивості наборів представлені в табл. 1.

Таблиця 1

Специфікації наборів вхідних даних

Набір даних	Кількість замовлень	Кількість транспортних засобів	Кількість депо	α	β
1а	5	2	2	1	0
1б	5	2	2	0	1
2а	10	5	4	1	0
2б	10	5	4	0	1
3а	16	5	5	1	0
3б	16	5	5	0	1
4а	50	10	7	1	0
4б	50	10	7	0	1
5а	100	20	15	1	0
5б	100	20	15	0	1

Джерело: авторська розробка

Для кожного набору було згенеровано по 5 тестових розташувань замовлень, водіїв та депо. Обчислені результати роботи та час виконання алгоритмів, що наведені в табл. 2. Для реалізації мурашиного алгоритму виставлено 10000 поколінь, для генетичного – 100. Розмір популяції генетичного алгоритму дорівнює 110. Також обчислимо відхил (Δ) результату генетичного алгоритму від роботи мурашиного (визначимо як оптимальний) за формулою:

$$\Delta = \left(\frac{res_{genetic}}{optimal} - 1\right) * 100\%.$$

Алгоритми виконувались на 4-ядерному процесорі Intel® Core® i7-6700HQ @ 3.2 GHz, 16 GB RAM.

Таблиця 2

Результати роботи та час виконання алгоритмів

		Тест 1	Тест 2	Тест 3	Тест 4	Тест 5	Середнє
Набір даних 1а							
Мурашиний	Результат	3630.4	2589.9	2985.2	4189.4	2282.8	3135.5
	Час роботи	0.089	0.089	0.089	0.083	0.088	0.088
Генетичний	Результат	2379.6	2080.4	1850.7	2432.6	2239.0	2196.5
	Час роботи	0.648	0.629	0.644	0.633	0.628	0.636
Набір даних 1б							
Мурашиний	Результат	1270.0	865.8	935.5	1787.6	819.0	1135.6
	Час роботи	0.089	0.088	0.088	0.083	0.088	0.087
Генетичний	Результат	1090.5	781.0	733.9	1215.5	787.3	921.6
	Час роботи	0.628	0.626	0.631	0.633	0.634	0.630
Набір даних 2а							
Мурашиний	Результат	6364.6	8455.4	5571.6	7538.1	5677.5	6721.4
	Час роботи	0.418	0.418	0.410	0.422	0.415	0.417
Генетичний	Результат	3236.4	3177.5	3071.6	4260.5	2578.3	3264.9
	Час роботи	0.258	0.285	0.268	0.251	0.267	0.266
Набір даних 2б							
Мурашиний	Результат	1258.3	1473.5	1154.4	1656.0	1193.7	1347.2
	Час роботи	0.418	0.417	0.427	0.425	0.419	0.421
Генетичний	Результат	540.1	670.7	567.1	762.7	506.6	609.4
	Час роботи	0.265	0.265	0.269	0.265	0.262	0.265
Набір даних 3а							
Мурашиний	Результат	8546.6	6508.5	8616.8	8347.2	9025.2	8208.9
	Час роботи	0.972	0.951	0.943	0.942	0.947	0.951
Генетичний	Результат	4682.5	3933.6	4371.3	3512.9	3576.9	4015.4
	Час роботи	5.012	4.835	4.801	5.140	4.804	4.918
Набір даних 3б							
Мурашиний	Результат	1447.9	1059.9	1274.4	1488.0	1686.6	1391.4
	Час роботи	0.99	0.93	0.96	0.95	0.97	0.96
Генетичний	Результат	806.6	657.8	723.9	604.0	557.0	669.9
	Час роботи	4.764	4.727	4.735	4.785	4.687	4.740
Набір даних 4а							
Мурашиний	Результат	18809.2	19689	16566.8	15731	19860.2	18131.2
	Час роботи	12.79	13.03	12.93	12.92	13.01	12.94
Генетичний	Результат	7498.3	8233.4	7014.4	8310.3	7992.6	7809.8
	Час роботи	37.86	41.30	31.81	33.05	36.61	36.13
Набір даних 4б							
Мурашиний	Результат	1693.2	1941.1	1214.5	1326.4	1673.1	1569.7
	Час роботи	12.77	12.84	13.40	13.30	12.91	12.98
Генетичний	Результат	556.0	666.9	552.1	620.4	628.9	604.9
	Час роботи	29.68	30.24	29.61	30.04	29.84	29.88
Набір даних 5а							
Мурашиний	Результат	41645.3	30084.5	36077.2	32736	39849	36078.4
	Час роботи	92.63	88.73	91.72	92.80	90.18	91.21
Генетичний	Результат	9794.7	10180.3	10782.1	9887.9	10475.	10224
	Час роботи	152.64	162.61	159.66	149.00	175.57	159.90
Набір даних 5б							
Мурашиний	Результат	1961.9	1266.9	1513.6	1390.3	1587.8	1544.1
	Час роботи	93.24	88.16	91.55	98.07	91.03	92.41
Генетичний	Результат	360.7	393.2	446.4	396.4	466.2	412.6
	Час роботи	133.63	131.28	133.35	131.96	133.32	132.71

Джерело: авторські розрахунки

З результатів проведеного експерименту в програмі можна зробити такі висновки:

Генетичний алгоритм знаходить дуже близькі до оптимальних рішення у випадку $\beta = 1$. Відхил від оптимального становить в середньому до 2%.

Генетичний алгоритм знаходить близькі до оптимальних рішення у випадку $\alpha = 1$. Відхил від оптимального становить в середньому до 28.5%.

Генетичний алгоритм на вхідних даних, занадто великих для використання точних алгоритмів, знаходить у випадках $\alpha = 1$ рішення в 2-4 рази краще за мурашиний.

Генетичний алгоритм на вхідних даних, занадто великих для використання точних алгоритмів, знаходить у випадках $\beta = 1$ рішення в 3-4 рази краще за мурашиний.

Таким чином, генетичний алгоритм вигідніше використати при кількості замовлень більше 16. При цьому будуть знайдені рішення, близькі до оптимальних.

Висновки. Для збільшення продуктивності роботи підприємства, що займається доставкою, необхідно відповідально та за допомогою розрахунків проводити планування доставок з застосуванням інформаційних технологій. Плани доставок мають формуватися відповідно до стратегії компанії та мати певний баланс між коштами, витраченими на доставку, та середнім часом очікування. Задача розподілення маршрутів доставки належить до класу NP-складних.

Для розподілу маршрутів доставки був запропонований метод на основі генетичного алгоритму, де основою стала задача комівояжера за умови багатьох маршрутів. Розроблена програма за допомогою мови програмування C++ та фреймворку Qt для демонстрації роботи різних алгоритмів та методів визначення оптимального плану доставки. Це дозволяє експериментально оцінювати можливості, ефективність роботи та порівнювати їх. За допомогою цієї програми на основі експериментальних даних та при певних заданих умовах встановлена перевага запропонованого генетичного алгоритму над відомим мурашиним. Генетичний алгоритм продемонстрував результат роботи, що відкриває можливості для його практичного застосування в розподілі маршрутів доставок.

Література

1. Dantzig G. B., Ramser J. H. The truck dispatching problem. *Management Science*. 1959. Vol. 6, No. 1. P. 80–91. DOI: https://doi.org/10.1287/mnsc.6.1.80.

2. Stodola P., Mazal J. Tactical models based on a multi-depot vehicle routing problem using the ant colony optimization algorithm. *International journal of mathematical models and methods in applied sciences*. 2015. Vol. 9. P. 330–337.

3. Chitty D., Wanner E., Parmar R., Lewis P. Applying partial-ACO to large-scale vehicle fleet optimisation. arXiv. 2019. URL: https://arxiv.org/abs/1904.07636.

4. Diestel R. Graph theory. NY: Springer-Verlag, 2005. 422 p.

5. Cormen T., Leiserson C., Rivest R., Stein C. *Introduction to algorithms* (2nd ed.). MIT Press, 2009. 1202 p.

6. Knuth D. Postscript about NP-hard problems. *ACM SIGACT News.* 1974. Vol. 6, No. 2. P. 15–16. DOI: https://doi.org/10.1145/1008304.1008305.

7. Наконечний С. І., Савіна С. С. Математичне програмування: навч. посіб. Київ: КНЕУ, 2003. 452 с.

8. Eiben A. E., Raué Paul-Erik, Ruttkay Z. Genetic algorithms with multi-parent recombination. PPSN III: Proceedings of the International Conference on Evolutionary Computation. The Third Conference on Parallel Problem Solving from Nature Jerusalem, Israel, October 9–14, 1994. P. 78–87.

9. Mitchell M. Genetic algorithms: an overview. *Complexity*. 1995. Vol. 1, No. 1. P. 31–39. DOI: https://doi.org/10.1002/cplx.6130010108.

10. Mitchell M. An Introduction to Genetic Algorithms. Cambridge, MA: MIT Press, 1996.

11. Koza J. R. Introduction to Genetic Programming. In K. E. Kinnear (Ed.), Advances in Genetic Programming. Cambridge*: MIT Press*, 1994. P. 21–41.

12. Steeb W.-H. The Nonlinear Workbook: Chaos, Fractals, Cellular Automata, Neural Networks, Genetic Algorithms, Fuzzy Logic: with C++, Java, SymbolicC++ and Reduce Programs. Singapore: World Scientific, 2001. 596 p.

References

1. Dantzig, G. B., Ramser, J. H. (1959). The truck dispatching problem. *Management Science*, 6 (1), 80-91. doi: https://doi.org/10.1287/mnsc.6.1.80.
2. Stodola, P., Mazal, J. (2015). Tactical models based on a multi-depot vehicle routing problem using the ant colony optimization algorithm. *International journal of mathematical models and methods in applied sciences*, 9, 330-337.
3. Chitty, D., Wanner, E., Parmar, R., & Lewis, P. (2019). Applying Partial-ACO to Large-scale Vehicle Fleet Optimisation. arXiv. Retrieved from https://arxiv.org/abs/1904.07636.
4. Diestel, R. (2005). *Graph Theory*. NY: Springer-Verlag.
5. Cormen, T., Leiserson, C., Rivest, R., & Stein, C. (2009). *Introduction to Algorithms* (2nd ed.). MIT Press.
6. Knuth, D. (1974). Postscript about NP-hard problems. *ACM SIGACT News*, 6(2), 15-16. doi: https://doi.org/10.1145/1008304.1008305.
7. Nakonechny, S.I., Savina, S.S. (2003). *Mathematical programming:* textbook. Kyiv: KNEU. (in Ukrainian)
8. Eiben, A. E., Raué, P., Ruttkay, Z. (1994). Genetic algorithms with multi-parent recombination. PPSN III: Proceedings of the International Conference on Evolutionary Computation. The Third Conference on Parallel Problem Solving from Nature Jerusalem, October 9–14, 2010, Israel, 78-87.
9. Mitchell, M. (1995). Genetic algorithms: an overview. *Complexity*, 1(1), 31-39. doi: https://doi.org/10.1002/cplx.6130010108.
10. Mitchell, M. (1996). *An introduction to genetic algorithms*. Cambridge, MA: MIT Press.
11. Koza, J. R. (1994). Introduction to genetic programming. In K. E. Kinnear (Ed.), Advances in Genetic Programming. Cambridge: *MIT Press*, 21-41.
12. Steeb, W.-H. (2001). *The Nonlinear Workbook: Chaos, Fractals, Cellular Automata, Neural Networks, Genetic Algorithms, Fuzzy Logic: with C++, Java, SymbolicC++ and Reduce Programs.* Singapore: World Scientific.

Olena Maslennikova
Associate Professor
V.N. Karazin Kharkiv National University
4 Svobody Sq., 61022, Kharkiv Ukraine
E-mail: 21041950EM@gmail.com, ORCID: https://orcid.org/0000-0002-7870-5743
Daniil Ostashev
student
V.N. Karazin Kharkiv National University
4 Svobody Sq., 61022, Kharkiv Ukraine
E-mail: ostashev@tutanota.com

GENETIC ALGORITHM OF DISTRIBUTION OF DELIVERY ROUTES

Information technologies have become an integral part of the life of modern society. That is why they are actively used to solve the complicated tasks related to optimisation of activities in economy, management, finance, social and other spheres. Logistics are not an exception, and information technologies are used to automate delivery processes, which contributes to the efficient and cost-effective operation of enterprises. This is possible provided that the supplier's resources are managed rationally and provided with an efficient way of distributing delivery routes. This explains the relevance of this work. The purpose of the article is to develop a genetic algorithm to determine the optimal delivery plan, as well as to compare its work with some well-known delivery optimization methods. The object of study is the delivery routes. The subject of the study is the distribution route distribution algorithm. The article proposes a genetic algorithm for solving the traveling salesman problem with several routes. The statement of the problem of determining the optimal delivery plan is presented, the possibility of using a genetic algorithm for the task is considered, the type of the genetic algorithm is described and a block diagram of its implementation is presented. A software implementation of the genetic algorithm and other methods of distributing delivery routes has been prepared, which is carried out using the C++ programming language and the Qt framework. For the algorithms to work, you need to generate an order, vehicles and depots, as well as enter the necessary parameters. The results of the operation of the algorithm and

other selected methods using the program were obtained, which made it possible to evaluate their effectiveness on the basis of experimental data for given parameters and to compare their work. The advantage of the proposed genetic algorithm over the well-known ant algorithm in randomly generated tests has been experimentally established. This indicates the applied value of the results obtained and the possibility of applying the algorithm in real conditions of distribution of supply routes.

Keywords: distribution of delivery routes, genetic algorithm, ant algorithm.
JEL Classification: R40, C02, C61, C63.

Е.В. Масленникова
доцент
Харьковский национальный университет имени В.Н. Каразина
пл. Свободы, 4, 61022, Харьков, Украина
E-mail: 21041950EM@gmail.com, ORCID: https://orcid.org/0000-0002-7870-5743
Д.А. Осташев
студент
Харьковский национальный университет имени В.Н. Каразина
пл. Свободы, 4, 61022, Харьков, Украина
E-mail: ostashev@tutanota.com

ГЕНЕТИЧЕСКИЙ АЛГОРИТМ РАСПРЕДЕЛЕНИЯ МАРШРУТОВ ДОСТАВКИ

Информационные технологии стали неотъемлемой частью жизни современного общества. Именно поэтому их активно применяют для решения сложных задач, связанных с оптимизацией деятельности в экономике, менеджменте, финансах, социальной и других сферах. Логистика не стала исключением, а информационные технологии используют для автоматизации процесса доставки, что способствует эффективной и рентабельной работе предприятий. Это возможно при условии рационального использования ресурсов поставщика и обеспечения его эффективным способом распределения маршрутов доставки. Именно этим объясняется актуальность данной работы. Целью статьи является разработка генетического алгоритма для определения оптимального плана доставки, а также сравнение его работы с некоторыми известными методами оптимизации доставки. Объектом исследования являются маршруты доставок. Предмет исследования – алгоритм распределения маршрутов доставки. В статье предложен генетический алгоритм для решения задачи коммивояжера с несколькими маршрутами. Представлено постановку задачи определения оптимального плана доставки, рассмотрено возможность применения генетического алгоритма для поставленной задачи, описано вид генетического алгоритма и представлено блок-схему его реализации. Подготовлено программную реализацию генетического алгоритма и других методов распределения маршрутов доставки, что осуществлено с помощью языка программирования *C++* и фреймворка *Qt*. Для работы алгоритмов нужно сгенерировать заказ, транспортные средства и депо, а также ввести необходимые параметры. Получено результаты работы алгоритма и других выбранных методов с помощью программы, что позволило оценить их эффективность на основе экспериментальных данных при заданных параметрах и сравнивать их работу. Экспериментально установлено преимущество предложенного генетического алгоритма над известным муравьиным на случайно сгенерированных тестах. Это свидетельствует о прикладной ценности полученных результатов и возможности применения алгоритма в реальных условиях распределения маршрутов поставок.

Ключевые слова: распределение маршрутов доставки, генетической алгоритм, муравьиный алгоритм.
JEL Classification: R40, C02, C61, C63.