

або доведення її до розрахункового мінімуму, що дозволяє надійно забезпечити кінцевий резуль- | тат.

Список використаної літератури:

1. Гусев А.А. Проблемы автоматизации сборки изделий в серийном производстве и прогрессивные пути их решения / Гусев А.А. // Автоматизация и современные технологии. - 1993. - №5. - С.2-8.
2. Захаров Н.В. Технологичность структуры изделия в условиях автоматизации сборочного производства / Захаров Н.В., Тимофеев Ю.В. // Автоматизированные станочные системы и роботизированные производства. - Тула: ТПИ, 1992.- С.76 - 83.

Захаров М.Н., Захарова О.И. "Исследование точности сборки многопозиционных технологических систем металлообработки резанием при их реинжиниринге".

Работа посвящена исследованию точности сборки многопозиционных технологических систем металлообработки резанием при их реинжиниринге. На основе выполненных исследований обеспечена возможность прогнозирования конечного результата при проектировании и изготовлении многопозиционных агрегатных станков, что позволило повысить точность обработки деталей на этом оборудовании.

Ключевые слова: многопозиционный, сборка, технология, системы, приспособление.

Zaharov M., Zaharova O. Study accuracy of multiposition metal cutting technology systems at reengineering

The work is devoted to research multiposition accuracy of metal cutting technology systems in their reengineering. Multi-process systems metal cutting, usually designed to the one option implementation, the method consists of fitting individual parts and components, which greatly affects the possibility of re-engineering of the equipment. After assembly design process conducted multiple installation and removal units, fitting, finishing, layout and so on. All this leads to increasing complexity of technological systems assembly multiposition metal machining during their reengineering, as well as theoretical and practical unpredictability of the final result. The process of designing and manufacturing of the process equipment not controlled "pass-through" system of quality assessment criteria and approval of the results at each stage. So, now the question arose of development of science-based methodology design and production process systems multiposition metal machining during their reengineering that will achieve the necessary precision equipment at the least cost, through harmonization results in all phases of design and production, and thus provide a predictable end features at the lowest cost. In this case we can speak of process control engineering and assembly at reengineering unique complex technological systems that can significantly improve their competitiveness, which is relevant.

Keywords: multiposition, assembly, technology, systems, equipment.

Стаття надійшла в редакцію: 06.05.2015р.
Рецензент: д.т.н., професор Тарельник В.Б.

УДК: 681.518+10.629

ПРАКТИЧНЕ ВИКОРИСТАННЯ МЕТОДУ ГЕНЕТИЧНИХ АЛГОРИТМІВ ДЛЯ РОЗВ'ЯЗАННЯ ЗАДАЧІ КОМІВОЯЖЕРА В ГЕОІНФОРМАЦІЙНИХ СИСТЕМАХ

А. М. Юнда, к.ф.-м.н, доцент
І. В. Жигулін, завідувач лабораторіями кафедри моделювання складних систем
С. О. Петров, к.т.н., ст. викладач
Р. О. Руденко, студент
Сумський державний університет

У статті проведено дослідження та адаптацію генетичного алгоритму як евристичного методу для розв'язання задачі комівояжера. Алгоритм застосовано у вигляді web-сервісу геоінформаційної системи з використанням картографічного інтерфейсу Яндекс.Карт.

Ключові слова. геоінформаційна система, евристичні алгоритми, задача комівояжера, генетичний алгоритм, web-сервіс, API, Яндекс.Карт.

Постановка проблеми. Розвиток картографічних, геолокаційних та геоінформаційних систем, а також поширення мобільних пристроїв, розкриває можливість для впровадження великої кількості сервісів, що все частіше використовуються як звичайними користувачами так і бізнес-структурами або підприємствами у різноманітних галузях [1]. Такі сервіси дозволяють окрім базових функцій: представлення різноманітних карт місцевостей, знаходження гео-об'єктів на карті

(будинків, вулиць, організацій) та ін., мають можливість визначати місцезнаходження користувача, а також прокладати оптимальні маршрути між пунктами. В сучасних умовах постають задачі прокладання оптимальних маршрутів з дотриманням додаткових умов: врахування швидкісного режиму, перешкод, дорожньої ситуації, можливість додавання транзитних точок. Задачі ускладнюються використанням таких технологій в інтерактивних системах, в умовах обмеження обчислювальних можливостей та пам'яті мобільних пристроїв.

У багатьох професійних та життєвих сферах виникає потреба скласти оптимальний маршрут, який повинен проходити через множину точок у довільному порядку. Метою роботи є створення web-сервісу, який в інтерактивному режимі дозволяє додавати на карті проміжні пункти призначення та отримувати карту оптимального маршруту. Критерієм оптимальності вважається мінімізація сумарної довжини або часу об'їзду всього маршруту. [2] Ця задача комбінаторної оптимізації називається задачею комівояжера.

Аналіз останніх досліджень і публікацій. Задача комівояжера належить до класу NP-повних задач [3], тому її не можна розв'язати за поліноміальний час. Метод повного перебору, який має складність $O(n!)$, не дозволяє виконати необхідні умови навіть для кількості точок більше 10. Всі інші точні методи також мають експоненціальний час роботи. До інших методів відносяться наближені алгоритми, що мають такі евристички: алгоритм імітації відпалу [4], мурашиний алгоритм [5], генетичні алгоритми [6]. Для розв'язання задачі комівояжера вирішено застосувати метод генетичних алгоритмів, який добре досліджений у публікаціях [7].

Постановка завдання. В умовах поставленої задачі маємо вирішити асиметричну задачу комівояжера. У загальному випадку її можна визначити як знаходження гамільтонового циклу орієнтованого графа. Маємо множину вершин $V = \{v_i\}, i=1..n$ та множину ребер $E = \{(v_i, v_j)\}, i=1..n, j=1..n$, де кожному ребру відпо-

відає значення відстані $c_{ij} \geq 0$ та $c_{ii} = 0$. Потрібно знайти таку перестановку вершин $p = (p_1, p_2, \dots, p_n)$ при $p_1 = 1$, щоб $\sum_{i=1}^{n-1} c_{p_i p_{i+1}} + c_{p_n p_1} \rightarrow \min$.

В кінцевому вигляді web-сервіс має мати клієнт-серверну архітектуру, де на серверній частині, за допомогою інтерфейсу CGI, будуть виконуватись обчислювальні операції та передаватись клієнту результати у вигляді web-сторінки з JavaScript-компонентами. Таким чином, сервіс має можливість виконуватись однаково швидко навіть на мобільних пристроях.

Виклад основного матеріалу. Для реалізації картографічного інтерфейсу клієнтської частини сервісу пропонується використовувати JavaScriptAPI Яндекс.Карт, що містить набір JavaScript-компонентів для легкого представлення інтерактивних карт на будь-якому сайті. Модуль «Маршрутизатор», що є частиною API Яндекс.Карт, дозволяє отримувати найкоротший маршрут між двома точками на карті. Таким чином, клієнтська частина складається з web-сторінки, що містить JavaScript-компоненти для отримання від користувача проміжних точок та зв'язку з API. Далі формується матриця суміжності отриманого графу шляхом перерахунку найкоротших маршрутів між всіма парами точок. Запити для перерахунку маршрутів виконуються асинхронно в інтерактивному режимі, що дозволяє суттєво зменшити час очікування. Коли користувач закінчує формування завдання, готова матриця суміжності передається до серверної частини за допомогою AJAX-запиту.

Серверна частина має вигляд CGI-скрипту, що написаний на мові C++, та виконується web-сервером. Цей скрипт розв'язує задачу комівояжера та повертає клієнтській частині потрібний порядок обходу вхідного графу. Клієнтська частина відображає знайдений маршрут і користувач отримує результат у потрібній для нього формі (візуально на карті, у вигляді текстової карти маршруту, збереження треку для подальшого використання, збереження у вигляді графічного файлу або вивід на друк).

Загалом схему роботи сервісу представлено на рисунку 1. Найбільш ресурсовитратна частина обчислень, що потребує оптимізації, це безпосередньо розрахунок оптимального маршруту.

Вхідною умовою для розв'язання задачі комівояжера є матриця суміжності орієнтованого повного графу з n вершинами, заповнена відстанями найкоротших маршрутів між кожною парою вершин. Відповіддю на задачу є перестановка з n вершин, що відповідає порядку обходу вершин знайденого маршруту. Вершини графу будемо називати містами, а будь-яку перестановку, що формує замкнутий маршрут – туром.

Першим етапом генетичного алгоритму є формування початкової популяції розміром pop_sz випадковими особинами, що представляють тур. Ці особини повинні бути закодовані певним чином. В нашому випадку будемо для кожного міста зберігати посилання на наступне місто в циклі.

Генерація початкової популяції здійснюється випадково, але для підвищення конкурентноздатності, наступне місто в турі з деякою ймовірністю обирається серед найближчих до нього.

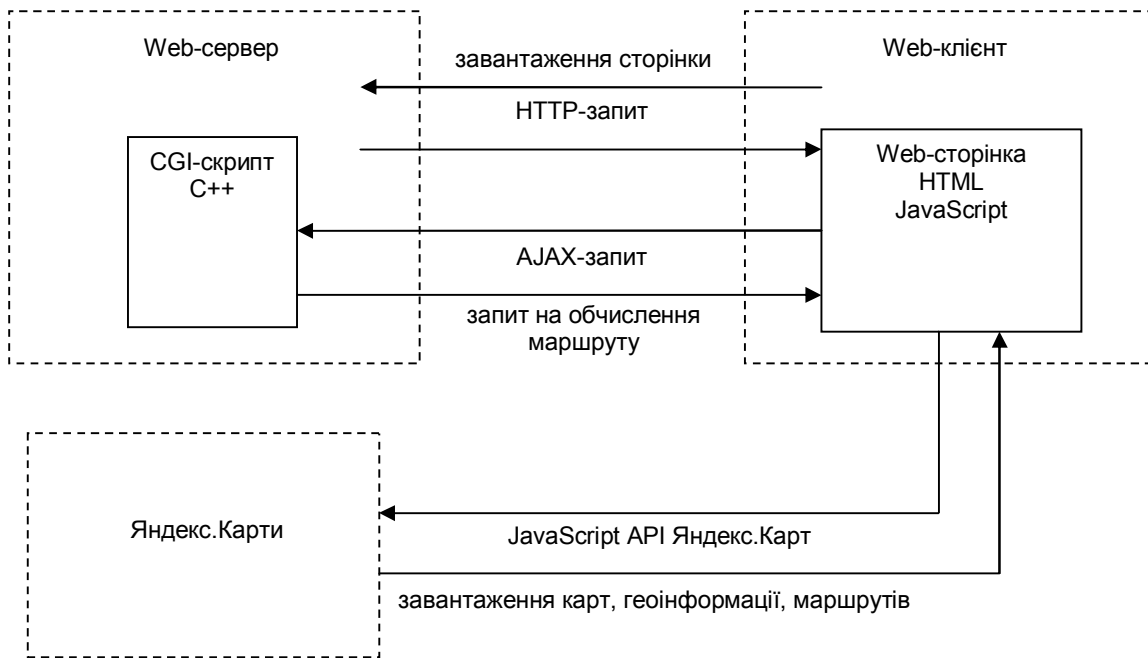


Рисунок 1 – Схема роботи web-сервісу.

Відбір особин для оператора схрещування використовується за принципом колеса рулетки, що базується на рангу особини. Ранг особини розраховується за значенням фітнес-функції, що є критерієм визначення кращої особини та дорівнює довжині туру. Введемо позицію особини (pos) у відсортованому за фітнес-функцією списку популяції. Особина з найменшим значенням фітнес-функції є найпристосованішою та має $pos = 1$, а з найбільшим значенням $pos = pop_sz$. Відбір особини обирається випадково з ймовірністю пропорційно ваговому рангу, який обраховується так:

$$rank(pos) = 2 - SP + 2 \cdot (SP - 1) \cdot \frac{pos - 1}{pop_sz - 1}, \quad \text{де } SP -$$

рівень селекції ($1 \leq SP \leq 2$), при цьому ваговий ранг буде дорівнювати від $2 - SP$ до SP . [5]

Оператор схрещування застосовується для відібраної зі всієї популяції на попередньому етапі пари особин. Якщо обидві особини однакові, тобто мають однакове значення фітнес-функції, до другого туру-батька застосовується мутація. Це робиться для того, щоб популяція не накопичувалась однаковими особинами.

Серед різних операторів схрещування, як найбільш ефективний, був обраний метод послідовного конструктивного кросоверу [6]. В результаті схрещування двох особин з'являється нащадок, який поєднує найкращі якості обох батьків. Цей алгоритм можна представити так:

1) Встановлюємо початкове місто туру-нащадка: $i = 1, p_i = 1$.

2) Обираємо найближче за обходом місто у першого туру-батька, яке ще не

використовувалось та позначимо це місто через α . Аналогічне місто з другого туру-батька обозначимо через β .

3) Якщо $c_{p_i, \alpha} \leq c_{p_i, \beta}$, наступне місто встановлюється α : $p_{i+1} = \alpha$. Інакше, наступне місто встановлюється β : $p_{i+1} = \beta$.

4) Якщо $i \neq n - 1$, то $i = i + 1$ та переходимо на шаг 2, інакше закінчуємо роботу алгоритму.

Для підтримання мінливості з невеликою ймовірністю застосовується мутація до отриманого після схрещування нащадка. Для цього випадково обирається місто, яке видаляється з ланцюга, та його сусіди з'єднуються напряму. Далі випадково обирається інше місто, перед яким вставляється видалене місто.

Отриманий нащадок додається до популяції та процес відбору, схрещування та мутації повторюється $pop_sz / 2$ разів, додаючи до популяції стільки ж нащадків. Для скорочення популяції відбираються найкращі за фітнес-функцією pop_sz особин, що будуть приймати участь у наступній генерації алгоритму.

Залежно від відведеного часу на розрахунок та кількості вхідних вершин визначається кількість генерацій алгоритму. На кожній такій генерації популяція поступово покращується новими особинами, що замінюють найменш пристосованих. Після закінчення процесу обирається найкраща особина-тур в кінцевій популяції та повертається результат.

Метод генетичних алгоритмів має декілька параметрів, від значення яких залежить час ро-

боти та відхилення отриманого результату від оптимального розв'язку, це: розмір популяції, вірогідність застосування мутації до нащадків, вірогідність вибору найближчих міст для початкової популяція та ін.

Для знаходження оптимальних значень па-

раметрів, проведено тестування роботи алгоритму на зразках бібліотеки симетричних та асиметричних задач комівояжера TSPLIB. Результати відхилення від оптимального розв'язку в залежності від ймовірності мутації та ймовірності вибору найближчого міста наведені в таблиці 1.

Таблиця 1 – Залежність відхилення розв'язку від параметрів.

Ймовірність вибору найближчих міст, %	Ймовірність мутації, %		
	3	5	10
0	6,31	6,40	6,45
10	6,38	6,43	6,45
20	6,31	6,31	6,45
30	6,22	6,40	6,34
40	6,36	6,40	6,43
50	6,24	6,31	6,40
60	6,36	6,33	6,41
70	6,26	6,26	6,24
80	6,29	6,37	6,33
90	6,20	6,20	6,32

Найменше відхилення від оптимального розв'язку було зафіксовано при значенні ймовірності мутації 5% та ймовірності вибору найближчих міст 90%. Оптимальні значення можуть відрізнятися в залежності від специфіки вхідних даних. Час роботи алгоритму суттєво не залежить

від цих параметрів.

Вибір розміру популяції залежить від бажаного відхилення або часу роботи (рисунок 2). Для обмеження часу роботи до 400 мс ефективніше використовувати розмір популяції 500, від 400 мс до 1400 мс – 1000, більше 1400 мс – 2000.

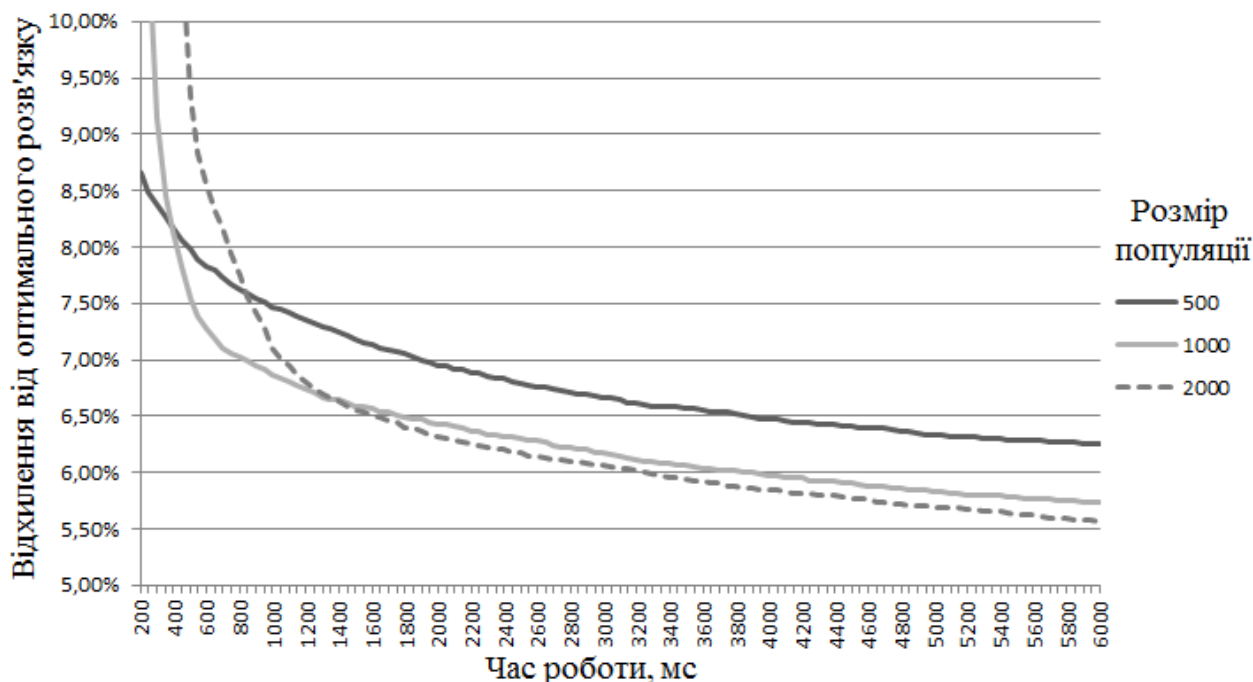


Рисунок 2 – Результати роботи алгоритму в залежності від розміру популяції.

Результати тестування для кожного окремого зразка TSPLIB асиметричної та симетричної задачі комівояжера наведені в таблицях 2 та 3. Тестування проводилось на комп'ютері з процесором IntelPentiumDual-Core 2.00 ГГц та оперативної пам'яті 2,00 ГБ під управлінням ОС Windows 7 шляхом запуску та перевірки кожного тесту 20

разів та фіксування середніх значень отриманих розв'язків, часу першого знаходження отриманих розв'язків та часу повної роботи програми при таких параметрах алгоритму: ймовірності мутації 5%, ймовірності вибору найближчих міст 90%, розміру популяції 2000, кількості генерацій алгоритму 800.

Таблиця 2 – Результати тестування для асиметричних зразків TSPLIB.

Тест TSPLIB	Кількість міст	Оптим. розв'язок	Знайд. розв'язок	Відхилення	Час знаходж. оптим. розв.	Час роботи алгоритму
br17	17	39	39	0,00%	81мс	1214 мс
ftv33	34	1286	1370,5	6,57%	312 мс	1372 мс
ftv35	36	1473	1498,35	1,72%	400 мс	1391 мс
ftv38	39	1530	1579,65	3,25%	687 мс	1433 мс
p43	43	5620	5629,9	0,18%	471 мс	1619 мс
ry48p	48	14422	15315,35	6,19%	1269 мс	1485 мс
ft53	53	6905	7702,65	11,55%	1167 мс	1626 мс
ftv55	56	1608	1719,8	6,95%	543 мс	1670 мс
ftv64	65	1839	1966,7	6,94%	1055 мс	1828 мс
ftv70	71	1950	2110,5	8,23%	847 мс	1948 мс
kro124p	100	36230	40645,05	12,19%	2038 мс	2558 мс
ftv170	171	2755	3215,65	16,72%	3347 мс	4008 мс

Таблиця 3 – Результати тестування для симетричних зразків TSPLIB.

Тест TSPLIB	Кількість міст	Оптим. розв'язок	Знайд. розв'язок	Відхилення	Час знаходж. оптим. розв.	Час роботи алгоритму
gr17	17	2085	2085	0,00%	40 мс	1101 мс
fri26	26	937	937	0,00%	22 мс	1245 мс
dantzig42	42	699	723,95	3,57%	1055 мс	1437 мс
berlin52	52	7542	7834,892	3,88%	1261 мс	1568 мс
eil76	76	538	569,2262	5,80%	1762 мс	2048 мс
kroA100	100	21282	23596,67	10,88%	2144 мс	2535 мс
bier127	127	118282	123231	4,18%	2688 мс	3037 мс
kroB150	150	26130	28806,72	10,24%	2997 мс	3568 мс
brg180	180	1950	1982	1,64%	628 мс	4248 мс
a280	280	2579	2863,889	11,05%	5099 мс	6654 мс

Висновки. Геоінформаційні системи зі зрозумілим інтерфейсом користувача та великою кількістю можливостей набувають все більшу популярність, особливо на ринку мобільних пристроїв.

Задача, що досліджена в роботі, належить до класу NP-повних задач, тому для її розв'язання доцільно використовувати евристику.

Вибраний евристичний метод розв'язку задачі є доцільним, оскільки наївний підхід до її розв'язку при наявності багатьох проміжних пун-

ктів становить відому проблему комбінаторної оптимізації, що потребує значних обчислювальних ресурсів.

Швидкість обчислень та доступність розробленого сервісу на багатьох системах досягається завдяки клієнт-серверній архітектурі.

В перспективі можна використовувати метод генетичних алгоритмів безпосередньо на мобільних пристроях, обладнаних GPS-навігатором.

Список використаної літератури:

1. Попов Б.Н., Алейник А.В. Геоинформационные системы как интернет-сервисы: обзор и перспективы. Информационные технологии и системы: управление, экономика, транспорт, право: Сб. тр. Международной научно-практической конференции «Инфогео 2013», часть 2 / Вып. 2 (11)
2. Р.А. Руденко, С.А. Петров, «IMA::2012»: Матеріали та програма НТК (Суми, 16-21 квітня 2012 р.), с.39 (2012)
3. Lawler, EL, Lenstra, JK, Rinnooy Kan, AHG, Shmoys, DB: The Travelling Salesman Problem, Chichester: Wiley (1985)
4. P. van Laarhoven and E. H. L. Aarts, Simulated Annealing: Theory and Applications, Kluwer Academic, 1987
5. M. Dorigo and L. M. Gambardella, Ant Colony System: A Cooperative Learning Approach to the Travelling Salesman Problem, IEEE Transaction on Evolutionary Computation, Vol. I, pp. 53-66, 1997
6. D.E. Goldberg, Genetic Algorithm in Search, Optimization and Machine Learning. Addison-Wesley, 1989.
7. Noraini Mohd Razali, John Geraghty, Genetic Algorithm Performance with Different Selection Strategies in Solving TSP: Proceedings of the World Congress on Engineering 2011 Vol II, 2011.
8. Ahmed, ZH: Genetic algorithm for the traveling salesman problem using sequential constructive crossover. International Journal of Biometrics & Bioinformatics. 3(6), 96–105 (2010).

Юнда А.Н., Жигулин И.В., Петров С.А., Руденко Р.А. Практическое использование метода генетических алгоритмов для решения задачи коммивояжера в геоинформационных

системах

В статье проведено исследование и адаптацию генетического алгоритма как эвристического метода для решения задачи коммивояжера. Алгоритм применено в виде web-сервиса геоинформационной системы реализованного на базе интерфейса Яндекс.Карт.

Ключевые слова. Геоинформационная система, эвристические алгоритмы, задача коммивояжера, генетический алгоритм, web-сервис, API, Яндекс.Карты.

Unda A.N., Zhygulin I.V., Petrov S.A., Rudenko R.A., The genetic algorithms application to solve the traveling salesman problem in geographic information systems

The paper investigated and improved the genetic algorithm like an heuristic method for solving the traveling salesman problem. Algorithm implemented as a web-service in framework of geographic information system based on Yandex.Maps interface.

Keywords. geographic information system, heuristic algorithms, traveling salesman problem, genetic algorithm, web-services, API, Yandex.Maps.

Стаття надійшла в редакцію: 05.05.2015р.

Рецензент: к.т.н., доцент Шандиба О.Б.