МАТЕМАТИЧНІ ТА ІНФОРМАЦІЙНІ МОДЕЛІ В ЕКОНОМІЦІ

УДК 004.8:519.85:656.7

Л.Ф. ГУЛЯНИЦЬКИЙ, А.І. ПАВЛЕНКО

ДИНАМІЧНА ЗАДАЧА ПОШУКУ НАЙКОРОТШОГО ШЛЯХУ З ДОДАТКОВИМИ УМОВАМИ ПРИ ПОБУДОВІ МАРШРУТУ АВІАПЕРЕЛЬОТІВ

Анотація. *Розглянута задача пошуку оптимального маршруту мандрівника між заданими пунктами з додатковими умовами на мережі авіасполучень певного регіону. Пропонується і досліджується підхід до розв'язання задачі пошуку шляху між заданими вершинами на відомому графі, що подає схему можливих авіаперельотів, з урахуванням вартості перельоту у залежності від часу. При цьому шлях може формуватися з урахуванням обмежень за часом, вартістю, бажаними або забороненими проміжними пунктами. Для пошуку шляху мінімальної вартості розроблено і досліджено спеціальний алгоритм оптимізації мурашиними колоніями з динамічним поколінням мурах. Природний паралелізм його обчислювальної схеми дозволяє отримувати і уточнювати отриманий розв'язок із урахуванням змін в умовах перельотів. Подається математична модель задачі, а також опис загальних особливостей запропонованого алгоритму. Для оцінки практичної ефективності алгоритму проведено обчислювальні експерименти, а також порівняння з класичною схемою оптимізації мурашиними колоніями.*
Ключові слова: *планування маршруту авіаперельотів, динамічна задача пошуку найкоротшого шляху, оптимізація мурашиними колоніями.*

Вступ

Розвиток та популяризація інформаційних і комунікаційних технологій дозволяє збирати інформацію і планувати індивідуальні екскурсійні чи бізнесові маршрути самостійно. За таких умов транспортним перевізникам важливо мати розвинену систему планування маршрутів із урахуванням особистих вподобань користувачів і різних умов подорожі: часу подорожі, видів транспорту для пересування, географії подорожі, бажаних місць для відвідування тощо.

Існуючі сервіси пошуку подорожей (SkyScanner, Kayak, Aviasales, Momondo тощо) дозволяють обирати початковий і кінцевий пункти подорожі для заданого часового періоду і більше орієнтовані на пошук найкоротшого шляху (найдешевшого або найшвидшого) між двома пунктами.

У ряді випадків користувачу бажано поєднати відвідування кількох міст і оптимізувати свій маршрут за часом, вартістю і особистими вподобаннями. Пропонується розглянути задачу планування маршруту подорожі, де відомі початковий, кінцевий пункти і побажання мандрівника. Користувач може визначити бажані проміжні пункти для відвідування, заборонені пункти, часовий проміжок, тривалість, максимальна вартість подорожі тощо. Така задача є різновидом *динамічної задачі пошуку найкоротшого шляху*.

Задача пошуку найкоротшого шляху є класичною задачею комбінаторної оптимізації, яка полягає у мінімізації суми ваг ребер, що складають маршрут між двома заданими вершинами на зваженому орієнтованому графі. Широке застосування задачі у різних галузях людської діяльності сприяє її активному вивченню і дослідженню нових алгоритмів та підходів. У пошукових системах в режимі реального часу нові маршрути повинні бути визначені протягом адекватного часу після запиту користувача [1]. Традиційні методи пошуку оптимального найкоротшого шляху часто не можуть бути використані для застосування в реальному часі через свою обчислювальну складність.

Деякі з перших досліджень були опубліковані в 1958 році, в одному із яких Кук і Хелсі [2] запропонували алгоритм, заснований на динамічному програмуванні з дискретизацією часу. Альтернативними шляхами розв'язання різних варіацій задачі займались Дрейфус [3], Халперн, Орден і Ром та інші. Складність задачі та її широке застосування в багатьох областях людської діяльності стимулює вивчення різних підходів і методів розв'язання. Велика увага приділяється наближеним методам пошуку, навіяним природою [4]. Вони включають в себе алгоритми оптимізації мурашиних колоній (ОМК), запропонованих Доріго [5, 6], які успішно застосовуються для розв'язування різних класів задач комбінаторної оптимізації, включаючи задачі комівояжера, маршрутизації, задачі про призначення, класифікації тощо.

У даному дослідженні пропонується шукати оптимальні маршрути подорожей, які здійснюються літаками. Розв'язком задачі вважатимемо послідовність авіарейсів, тобто послідовність аеропортів і час відправлення разом із тривалістю зупинок у проміжних пунктах. Варто зазначити, що, наприклад, туристичні маршрути зазвичай будуються з тривалими зупинками у проміжних пунктах. Разом з тим, користувачі прагнуть зменшувати матеріальні витрати на перельоти і іноді можуть нехтувати послідовністю проміжних пунктів, видом авіаліній, зручними стиковками і пересадками в аеропортах тощо. Окрім того, маршрут може бути покращений у вартості при включенні додаткових, не визначених користувачем проміжних пунктів. Інколи користувачеві необхідно виключити з маршруту деякі проміжні пункти, що можуть додаватись для покращення вартості (через візові обмеження, небажання відвідувати деякі міста тощо). Оскільки low-cost компанії часто пропонують авіарейси з великим коливанням вартості за короткий період часу, доцільним є не обмежувати подорож заданими початковою і кінцевою датою, а визначити період і діапазон тривалості подорожі.

Багато прикладних задач із різних сфер людської діяльності (комунікації, транспортних мереж тощо), в яких необхідно оперувати кількома зв'язками між вузлами, адекватно моделюються за допомогою мультиграфів, тобто графів, у яких між вершинами може існувати декілька дуг. У зв'язку зі збільшенням зацікавленості до динамічного управління транспортними системами постає проблема пошуку найкоротших шляхів на графах великої розмірності (наприклад, дорожньої мережі), де ваги дуг динамічно змінюються з плином часу [7]. Оскільки між аеропортами може існувати кілька рейсів, схема авіаперельотів адекватно подається мультиграфом.

1. Детерміновані дискретні динамічні задачі пошуку найкоротшого шляху

Розглядаються основні відмінності між статичною і динамічною задачами пошуку найкоротшого шляху.

Статична задача пошуку найкоротшого шляху між заданими вершинами. Дано граф $G=(V,A)$, де $V=\{v_1,...,v_n\}$ – множина n вершин; A – множина дуг, s – початкова, а d – кінцева вершина. Кожній дузі (i,j) відповідає вартість c_{ij} , яка може вимірюватись у матеріальних витратах, тривалості, відстані або будь-якій іншій мірі ресурсних витрат. Метою задачі пошуку є знаходження оптимального шляху (найкоротшого, найдешевшого, найшвидшого в залежності від вимірюваних ресурсів) між однією початковою вершиною і однією цільовою (задача one-to-one), між однією початковою і усіма іншими вершинами (задача one-to-all), між усіма вершинами і однією цільовою (all-to-one). Такі задачі частіше розв'язують методами динамічного програмування [8, 9] або алгоритмом Дейкстри [10] для графів з невід'ємними вартостями дуг.

Динамічна задача пошуку найкоротшого шляху у нашому випадку виглядає так. Дано граф $G=(V,A)$ з $\|V\|$ вершинами і $\|A\|$ дугами, початкова s і кінцева вершини d. Кожна дуга (i,j) характеризується часовою затримкою d_{ij}, тобто необхідним часом для переміщення з пункту i в пункт j, $(i,j)\in A$. Така затримка залежить від часу відправлення з пункту i. Функція $d_{ij}(t)$ повертає тривалість переміщення з пункту i в час t в пункт j. Отже, прибуття в пункт j відбувається в час $t+d_{ij}(t)$. В різних варіантах задачі функція затримки може бути неперервною або дискретною, стохастичною або детермінованою. Задача пошуку найкоротшого шляху зі стохастичними затримками може моделювати задачу пошуку найшвидшого шляху у місті з інтенсивним рухом і заторами, тобто коли для деяких проміжків маршруту неможливо точно оцінити час проїзду.

Додатково кожна дуга $(i,j)\in A$ характеризується вартістю c_{ij}, тобто відмінним від часу ресурсом – відстанню або ціною проїзду. Для деяких варіантів задачі також вводиться вартість простою у проміжних пунктах $w_i(t)$, яка характеризує вартість зупинки за одиницю часу у пункті i з часу t по $t+1$. Наприклад, очікування в пункті i з часу t_1 по t_q дорівнює $\sum_{z=1}^{q} w_i(t_z)$. Метою динамічної задачі пошуку найкоротшого шляху є знаходження мінімальних шляхів за часом (найшвидших шляхів), мінімальних шляхів за вартістю (найдешевших) або багатокритеріальних найкоротших шляхів у графі G.

Просторово-часова мережа (time-space network). Для будь-якого дискретного динамічного графу $G=(V,A)$ можна побудувати просторово-часову мережу (або розширену за часом мережу), де кожна вершина дублюється для кожного дискретизованого часового кроку. Таким чином, потужність множини вершин графу збільшується до $n\theta$, де n – потужність множини вершин $V=\{v_1,...,v_n\}$, θ – кількість часових кроків для розв'язуваної задачі. Оскільки отримана просторово-часова мережа є статичною, з'являється можливість для поставленої динамічної задачі застосовувати загальні алгоритми пошуку найкоротших шляхів [11].

Властивості FIFO і узгодженості вартості. Динамічний граф є графом FIFO, якщо кожна дуга графу задовольняє властивості FIFO. Дузі (i,j) притаманна властивість FIFO (first in first out), якщо $t+d_{ij}(t)\le(t+k)+d_{ij}(t+k)$, $\forall t,k\ge 1$.

Властивість узгодження вартості аналогічна FIFO для іншого виду ресурсів, тобто дуга (i,j) вважається узгодженою за вартістю, якщо при виході з пункту i вартість шляху менша, ніж при виході з пункту j.

Чабіні [11] виділив 6 типів динамічних задач на основі шести критеріїв:

1. Задачі пошуку найменших за часом або вартістю шляхів. Для першого типу задач вартістю проходу по дузі є час, в той час як для другого типу – більш узагальнена вартість, яка часто змінюється з часом.

2. Час може бути неперервним або дискретизованим. Дискретизована динамічна мережа може бути подана як статична розширена просторово-часова мережа.

3. Мережі FIFO і non-FIFO. Дрейфус припустив, що загальним алгоритмом Дейкстри можна розв'язувати динамічну задачу, поставлену Куком і Хелсі, так само ефективно, як і загальні статичні задачі. У 1993 році Кауфман і Сміт довели, що це можливо, лише якщо мережа володіє властивостями FIFO.

4. Дозволений або заборонений простій у проміжних пунктах. Якщо розглядати пошук найшвидшого шляху для автомобільних доріг, то простій у проміжних пунктах не є обов'язковим. Якщо ж розглядати подорожі громадським транспортом, то простої на зупинках слід враховувати. Шульц [12] запропонував спосіб розширювати граф з врахуванням простоїв у проміжних пунктах.

5. Кількість початкових і цільових пунктів (one-to-all, all-to-one, one-to-one).

6. Цілочислове або дійсне подання вартостей переходу по дузі. Вартість або час можна подавати у відносних одиницях виміру і використовувати їх для спрощення розв'язання задачі на мультиграфі [13].

Останнім десятиліттям стали розвиватись дослідження класу задач пошуку оптимальних туристичних маршрутів (shortest path tour problem, multimodal trip planner та ін.). Такі задачі включають дослідження пошуку шляхів для комбінації різних видів транспорту [14], задачі пошуку найкоротшого шляху через задані вершини [13, 15], розв'язання задачі комівояжера для динамічного маршруту між заданими пам'ятками [16].

2. Математична модель задачі

Дано зважений орієнтований мультиграф $G=(V,A)$, в якому між двома вершинами (відповідають аеропортам у відповідних містах) може існувати декілька дуг, що відображають наявність авіасполучення між цими містами, де $V=\{v_1,...,v_n\}$ – множина n вершин, які являють собою n пунктів (міст або аеропортів); A – множина дуг, які відображають наявні авіасполучення.

Між деякими пунктами можуть існувати кілька рейсів від різних авіакомпаній, які відрізняються за вартістю і часом.

Позначимо (v_i,v_j) множину дуг із вершини v_i до вершини v_j (множину рейсів між пунктами), $v_i,v_j \in A$, їх кількість $N_{ij}=\|(v_i,v_j)\|$, а a_{ij}^k – конкретну дугу $a_{ij}^k \in (v_i,v_j)$, $k \in \{1,...,N_{ij}\}$ – рейс. Не виключаються випадки, коли $(v_i,v_j)=\varnothing$ для деяких вершин і напрямів – це відповідає випадкам, коли між відповідними пунктами не існує прямого сполучення.

Проходження по кожній дузі a_{ij}^k несе витрати у вигляді часових затримок $\lambda(a_{kl})$ і вартостей $c(a_{i_k,i_{k+1}},t)$. $c(a_{i_k,i_{k+1}},t)$ – невід'ємна залежна від часу функція, яка являє собою загальну вартість переходу $a_{i_k,i_{k+1}}$. В реальності туристам часто необхідно очікувати наступний рейс у проміжних пунктах або планувати тривалу зупинку у місті. Час очікування у початковій вершині v_{i_k} дуги a_{i_k} позначимо як $g(a_{i_k})$.

Необхідно знайти оптимальний за вартістю шлях з початкової вершини $s \in V$ до цільової $d \in V$ у заданий проміжок часу $T=[t_{0_{\min}},t_{0_{\max}}] \subseteq \mathrm{T}$ (припускається, що хоча б один такий шлях існує). Нехай $t_0 \in T$ – час відправлення з вершини s.

Шляхом $x(s,d,t_0)$ із пункту s у пункт d називається упорядкована послідовність дуг $(a_{i_1i_2},a_{i_2i_3},a_{i_3i_4},\dots,a_{i_{w-1}i_w})$, для якої:

1) $i_1=s,\ i_w=d$,
2) $a_{ij} \in (v_i,v_j),\ v_i,v_j \in V,\ i,j \in \{i_1,i_2,\dots,i_w\}$.

Якщо прохід по конкретній дузі $a_{kl} \in A$, що входить до маршруту і відповідає перельоту із вершини k у вершину l, починається в час t_{k-1}, то прибуття в пункт l відбудеться в $t_k=t_{k-1}+\lambda(a_{kl})$, де $\lambda(a_{kl})$ – час такого перельоту.

Час подорожі – різниця часу прибуття і часу початку подорожі (час вважаємо фіксованим).

$$t(x)=\sum_{k=1}^{w-1}\left[t(a_{i_k,i_{k+1}})+g(a_{i_k})\right]-t_0, \qquad (1)$$

де $t(x)$ – час проходження повного шляху;

$t(a_{i_k,i_{k+1}})$ – час транзиту вздовж дуги $a_{i_k,i_{k+1}}$;

$g(a_{i_k})$ – час очікування у початковій вершині v_{i_k} дуги a_{i_k}, причому $g(a_{i_w})=0$.

Вартість шляху x визначається як $c(x,t)=\sum_{k=1}^{w-1} c(a_{i_k,i_{k+1}},t)$, де w – кількість дуг у шляху x.

Метою є знаходження оптимального за вартістю шляху $x^*(s,d,t_0)$ (або доступних шляхів із урахуванням додаткових умов). Додатково можуть накладатись умови: задаватись обов'язкові для відвідування вершини $V_{mandatory}$ (2), заборонені вершини $V_{\text{prohibited}}$ (3), максимальна вартість повного шляху $c_{\max}$ (4), максимальна кількість вершин у маршруті n_{max} (5), тривалість маршруту у часі (6). Варто зазначити, що змінювати міста (вершини) кожного дня необов'язково:

$$V_{mandatory} \subseteq x^*, \tag{2}$$

$$\forall v \in V_{\text{prohibited}} : v \notin x^*, \tag{3}$$

$$c(x^*,t) \le c_{\max}, \tag{4}$$

$$\left\| x^* \right\| \le n_{\max}, \tag{5}$$

$$t_{\min} \le t(x^*) \le t_{\max}, \tag{6}$$

де $t_{\min}, t_{\max}$ – межі допустимого часового проміжку.

Мета динамічної задачі пошуку найкоротшого шляху – мінімізація вартості шляху (7) серед припустимих:

$$c(x^*,\mathrm{t}) = \min_{\mathrm{x(s,d,t_0)}} \{\mathrm{c(x,t)}\}. \tag{7}$$

3. Опис методу ОМК

Ідея ОМК [4] навіяна способом розв'язання задач оптимізації за допомогою низькорівневої комунікаційної поведінки співпрацюючих мурах, які шукають шлях між їх мурашником і джерелом їжі. Як і в реальному житті, мурахи починають випадково блукати в пошуках їжі і врешті повертаються до мурашника. Під час прогулянки вони залишають феромонні сліди, які роблять вже пройдений ними шлях більш привабливим для інших мурах, так як їх шанси на успіх у пошуку продовольства зростає. Наступні мурахи обирають маршрути, беручи до уваги кількість залишеного феромону і видиму довжину шляху (відстань між сусідніми пунктами). Однак з часом феромонні сліди випаровуються, таким чином, зменшуючи свою силу привабливості. Очевидно, короткі і популярні шляхи мають більшу концентрацію феромону, ніж довгі.

Моделювання процесу випаровування в ОМК сприяє уникненню передчасної збіжності до локального оптимуму.

Алгоритм ітераційно створює покоління мурах, які стохастично обирають проміжні пункти і будують маршрут поетапно. Пункт, в який початково поміщається мураха, визначається накладеними умовами задачі. На початковому етапі алгоритму створюються і ініціалізуються матриці відстаней між пунктами, феромонів, задається початкова «краща» вартість маршруту та інші параметри алгоритму. На наступному етапі створюється покоління мурах, які одночасно вирушають зі стартової точки. Кожна мураха з ймовірністю p_{ij}^{u} (8) визначає наступний проміжний пункт:

$$p_{ij}^{u}=\frac{\tau_{ij}^{\alpha}\eta_{ij}^{\beta}}{\sum\limits_{j\in allowed\ j}\tau_{ij}^{\alpha}\eta_{ij}^{\beta}}, \tag{8}$$

де τ_{ij}^{α} – кількість феромону, залишеного на шляху з пункту i в j, $\alpha \geq 0$ – параметр, який визначає вплив наявного рівня феромону τ_{ij}; η_{ij}^{β} – бажаність переходу ij в залежності від апріорних знань про відстань (видимість мурахи), $\eta_{ij}=1/d_{ij}$, β – параметр, який контролює вплив η_{ij}. Після завершення побудови маршруту усіма мурахами поточного покоління, відбувається оновлення феромону:

$$\tau_{ij}:=(1-\rho)\tau_{ij}+\sum_{u}\Delta\tau_{ij}^{u}, \tag{9}$$

де ρ – коефіцієнт випаровування феромону, $\Delta\tau_{ij}^{u}$ – рівень феромону (10), який залишила мураха u:

$$\Delta\tau_{ij}^{u}=\begin{cases} c_{predefined}/c(x_u), & якщо\ (i,j)\in x_u \\ & в\ іншому\ разі \end{cases} \tag{10}$$

Тут $c_{predefined}$ – коефіцієнт, який зазвичай відповідає порядку оптимальної вартості маршруту; $c(x_u)$ - вартість знайденого мурахою u маршруту. Якщо покоління виявило розв'язок, кращий за поточний оптимальний, тоді його необхідно оновити. Для різних типів задач існують евристики, тобто апріорні знання, які доцільно використовувати для покращення розв'язання, тому на наступному кроці можуть бути виконані деякі специфічні опціональні процедури – «daemon actions». Після побудови повного шляху мураха звільняє усі задіяні ресурси і губиться.

Якщо умова завершення не задоволена, то створюється нова популяція мурах і починається нова ітерація алгоритму.

4. ОМК для пошуку шляху в динамічному мультиграфі з додатковими умовами

Пропонується алгоритм на основі ОМК для пошуку оптимального шляху у динамічному мультиграфі з додатковими умовами, який працює з дискретизованим часом. В умовах поставленої задачі важливим критерієм є час побудови маршруту. Алгоритм оперує заданою кількістю мурах, а не поколіннями. Класична схема ОМК створює таку ж кількість мурах, як і запропонована.

Під час ініціалізації визначається значення $c_{predefined}$ додатковою процедурою пошуку припустимого розв'язку мурахою. Якщо за визначену параметром кількість ітерацій на етапі ініціалізації не було знайдено жодного припустимого маршруту з початкового пункту до цільового (без додаткових умов), то алгоритм припиняє роботу і переходить до наступного запиту.

На першому етапі усі дуги, що містять $V_{mandatory}$ (або цільову вершину), наділяються додатковим значенням феромону $\tau_{\max}$, всі інші – τ_0. Під час вибору проміжних пунктів мурахи оперують дугами, а не вершинами, як у класичному ОМК. Таким чином, алгоритм ефективно справляється з проблемою існування кількох дуг між вузлами.

Як тільки мураха завершує побудову маршруту або її частковий маршрут не задовольняє умовам (2)-(6), мураха губиться і звільняє ресурси. Тому алгоритм не затримує невикористовувані ресурси, як при синхронізації покоління мурах класичної схеми ОМК. Варто зазначити, що на відміну від багатьох відомих алгоритмів ОМК, у запропонованому підході мурахи співпрацюють через постійне оновлення феромону після кожного проходу мурахи (онлайнове покрокове відкладання феромону), а не після завершення діяльності всього покоління. Кількість мурах належить до параметрів алгоритму. Порівняння ефективності класичного і запропонованого підходів подані далі.

Як зазначалося, оновлення феромону здійснюється після кожного вдалого проходу мурахи і лише для тих дуг, які належать знайденому маршруту:

$$\tau_{ij} := \tau_{ij} + \Delta\tau_{ij}^{u}, \tag{11}$$

$$\tau_{\min} \le \tau_{ij} \le \tau_{\max}, \tag{12}$$

$$(i, j) \in x_u, \tag{13}$$

де τ_{min}, $\tau_{\max}$ – параметри алгоритму. Випаровування феромону виконується в рамках «дій демона» після кожної b-ї мурахи для всієї матриці феромонів (14). Така процедура виконується для зменшення операцій над базою даних

$$\tau_{ij} := (1-\rho)\tau_{ij}, \ \forall (i, j) \in A. \tag{14}$$

5. Результати обчислювальних експериментів

Для дослідження практичної ефективності алгоритмів був проведений обчислювальний експеримент, у якому використовувалися реальні дані, що були зібрані за допомогою парсерів і API сайтів для пошуку маршрутів для подорожування, таких як SkyScanner і Google (QPX Express). Отримані дані включають 15497 рейсів за один тиждень. Відповідно до умов експерименту, необхідно було знайти оптимальний шлях між аеропортом Бориспіль та 113 аеропортами в Європі.

Для наочності, наступні діаграми містять інформацію лише про 50 випадкових цільових міст.

Рис. 1 подає результати застосування динамічного алгоритму з одним поколінням і класичного алгоритму ОМК за часом (в сек.).

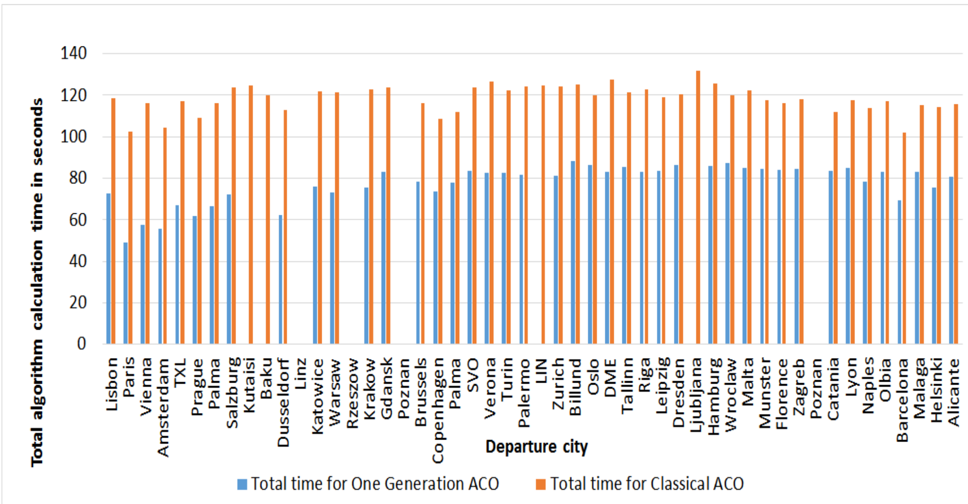

Рисунок 1 – Порівняння алгоритму з одним поколінням та класичної ОМК за часом роботи

Класичний алгоритм виявився у середньому на 31% повільніший від запропонованого. У той же час, запропонований алгоритм не зміг знайти прийнятні маршрути для 25 із 113 цільових міст, в той час як класичний – для 16.

Рис. 2 демонструє результати експерименту для 50 випадкових цільових міст та ілюструє порівняння алгоритму з одним поколінням та класичного алгоритму ОМК за найкращою ціною маршруту (в EUR).

Як бачимо, запропонований алгоритм забезпечує на 16% кращі результати.

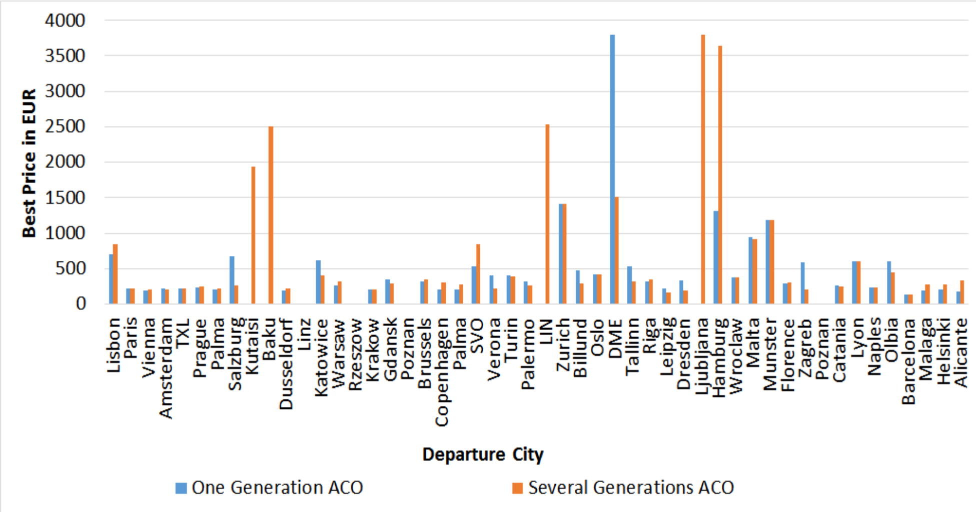

Рисунок 2 – Порівняння алгоритму з одним поколінням та класичної ОМК за найкращою ціною маршруту

Рис. 3 подає залежність результатів роботи алгоритму від кількості мурах.

Рисунок 3 – Залежність між загальною кількістю мурах алгоритму і вартістю знайденого шляху

Табл. 1 відображає параметри алгоритмів ОМК, що використовувались в обчислювальному експерименті.

Таблиця 1 – Параметри алгоритмів

Назва	Значення
Кількість запусків для кожного запропонованого підходу	100
Загальна кількість мурах, що була створена алгоритмом	500
Коефіцієнт впливу залишеного феромону α	0.5
Коефіцієнт впливу η_{ij} – β	0.5
Максимальна кількість перехідних вузлів n_{max}	10
Розмір покоління класичної ОМК	20
Номер мурахи, після якої починається випаровування в підході з одним поколінням b	20
Початкова кількість феромонів τ_0	0.1
Мінімальна кількість феромонів $\tau_{\min}$ за дугу	0.1
Максимальна кількість феромонів $\tau_{\max}$ за дугу	0.7
Коефіцієнт випаровування феромону ρ	0.1
Максимальна кількість ітерацій для виявлення $c_{predefined}$	3000
Початковий аеропорт	KBP

В цілому слід зазначити, що попередні результати вказують на можливість запропонованого алгоритму забезпечити прийнятну якість результатів і значно зменшений час роботи. Проте за таким алгоритмом не завжди можна знайти оптимальні шляхи, які надає можливість отримати класичний алгоритм ОМК.

Висновки

Робота містить опис і формалізацію спеціальної динамічної задачі пошуку найкоротшого шляху з додатковими умовами. Подано огляд динамічних задач пошуку найкоротших шляхів і аналіз специфіки побудови оптимальних маршрутів авіаперельотів. Через необхідність мінімізації часу роботи у режимі реального часу було розвинено підхід на основі методу ОМК – розроблено оригінальний алгоритм розв'язання динамічної задачі пошуку найкоротшого шляху на мультиграфі підвищеної розмірності, який опрацьовує одне покоління, а процедура вибору наступного проміжного пункту оперує поняттями дуг. Запропонований підхід може застосовуватися до розв’язання задач із додатковими умовами – максимальною кількістю проміжних міст, максимальною вартістю, забороненими і бажаними містами. Алгоритм дозволяє гнучко доповнювати модель додатковими умовами і постійно контролювати допустимість отримуваних розв’язків. Окрім того, використовується додаткове оновлення матриці феромонів по тих дугах, які містять бажані міста.

Отримані попередні результати показують, що запропонований алгоритм забезпечує прийнятну якість результатів і значно менший час роботи – це є дуже важливим при функціонуванні у режимі реального часу. Проте він не завжди знаходить припустимі шляхи, які отримує класичний алгоритм ОМК.

Додаткових досліджень потребує оптимізація параметрів алгоритму, а також порівняння з іншими алгоритмами пошуку. Схема оновлення феромону може бути покращена для роботи з кількома запитами на основі використання вже отриманої інформації.

СПИСОК ЛІТЕРАТУРИ

1. L. Fu. Heuristic shortest path algorithms for transportation applications: State of the art / L. Fu, D. Sun, L.R. Rilett. – Computers & Operations Research, 2006. – Vol. 33, No. 11. – P. 3324–3343.
2. K.L. Cooke. The shortest route through a network with time-dependent intermodal transit / K.L. Cooke, E. Halsey. – J. Math. Anal. Appl, 1966. – P. 493–498.
3. S.E. Dreyfus. An appraisal of some shortest-path algorithms / S.E. Dreyfus. – Operations Research, 1969. – P. 395–412.
4. Pintea C.M. Advances in Bio-inspired Computing for Combinatorial Optimization Problems / Pintea C.M. – Berlin, Heidelberg: Springer-Verlag, 2014. – 188 p.
5. M. Dorigo. Ant Colony Optimization, A Bradford Book / M. Dorigo, T. Stützle. – MIT Press.Cambridge, Massachusetts, London, 2004. – 305 p.
6. M. Dorigo. Ant colony optimization: overview and recent advances / M. Dorigo, T. Stützle. – Handbook of metaheuristics, Springer US, 2010. – P. 227–263.
7. B. Ding. Finding Time-Dependent Shortest Paths over Large Graphs / B. Ding, J. Xu Yu, Lu Qin. – Proceedings of the 11th international conference on Extending database technology: Advances in database technology, Nantes, France, 2008. – P. 205–216.
8. R. Bellman. On a routing problem / R. Bellman. – Quarterly of Applied Mathematics, 1958. – P. 87–90.
9. L.R. Ford. Network Flow Theory / L.R. Ford. – The RAND Corporation, Santa Monica, California, 1956. – 923 p.
10. E.W. Dijkstra. A note on two problems in connexion with graphs / E.W. Dijkstra. – Numerische Mathematik 1, 1959. – P. 269–271.
11. I. Chabini. Discrete dynamic shortest path problems in transportation applications / I. Chabini. – Transp. Res. Rec, 1998. – no. 1645. – P. 170–175.
12. Schulz F. Timetable information and shortest paths, 2005.
13. I. Bérubé. Time-dependent shortest paths through fixed sequence of nodes: Application to a travel planning problem / I. Bérubé, J. Potvin and J. Vaucher. – Comput. Oper. Res, 2006. – vol. 33, no. 6. – P. 1838–1856.
14. T. Xu. Research on the Connecting Path Search Algorithm for Air-Rail Integration / T. Xu, X. Ding, J. Li .– J. of Software, 2013. – 8, no. 8. – P. 1889–1896.
15. P. Festa. The Shortest Path Tour Problem: Problem Definition, Modeling and Optimization / P. Festa. – Proc. International Network Optimization Conference, Pisa, Italy, 2009. – 923 p.
16. D. Gavalas, V. Kasapakis, C. Konstantopoulos, K. Mastakas, and G. Pantziou. A survey on algorithmic approaches for solving tourist trip design problems / Journal of Heuristics, 2014. – v.20 n.3. – P. 291–328.

Стаття надійшла до редакції 14.04.2015