УДК 519.853: 658.52

В. В. Козалєтов, В. М. Бацамут

МЕТОД МАРШРУТИЗАЦІЇ ПОШУКОВИХ ГРУП НА СТАЦІОНАРНІЙ МЕРЕЖІ ДЛЯ ЕФЕКТИВНОГО ПРОЧІСУВАННЯ НАСЕЛЕНИХ ПУНКТІВ І ПОШУКУ ОЗБРОЄНИХ ЗЛОЧИНЦІВ

Запропоновано метод, який усуває проблему неефективного обходу пошуковими групами певного населеного пункту, що є важливим у ході виконання пошукових, рятувальних, моніторингових завдань у кризових районах різного характеру походження.

У методі застосовано ідею динамічного програмування, яка полягає у послідовному поділі початкової структури транспортної мережі населеного пункту на низку підструктур. У межах кожної підструктури відшукується реберно-простий найдовший/найкоротший шлях між парою визначених вершин. Сукупність визначених екстремальних шляхів складає Маршрутний план для пошукових груп у ході проведення пошукових заходів, що, в певному розумінні, упорядковує прочісування населеного пункту.

Метод розроблено і викладено в термінах теорії графів і теорії множин та подано у формалізованому вигляді.

***Ключові слова:** прочісування, пошукові групи, мультиагентна система, маршрутизація, мережевий об'єкт, зважений неорієнтований (орієнтований) граф, екстремальні шляхи, критерій оптимізації, метод.*

Постановка проблеми. Одним із завдань, які визначені у ст. 1 Закону України «Про Національну гвардію України» [1], є « … припинення терористичної діяльності, діяльності незаконних воєнізованих або збройних формувань (груп), терористичних організацій, організованих груп та злочинних організацій». Виконання цього завдання обумовлене функцією Національної гвардії України (НГУ), що викладена у ст. 2 цього ж Закону, а саме «… участь у спеціальних операціях із знешкодження озброєних злочинців, припиненні діяльності не передбачених законом воєнізованих або збройних формувань (груп), організованих груп та злочинних організацій на території України, а також у заходах, пов'язаних із припиненням терористичної діяльності». Потрібно зауважити, що виконання цієї функції буде дуже складним завданням для НГУ у післявоєнний період [2].

Одним із способів дій підрозділів (формувань) НГУ, у ході виконання цих завдань, є і залишатиметься пошук. Зважаючи на відносно щільне розташування населених пунктів територією України, значна частина пошукових операцій проводитиметься саме в них. Разом з тим мережева організація населених пунктів зумовлює те, що бойовий порядок підрозділу, призначеного для проведення пошуку, вимушено поділятиметься на кілька пошукових груп, які будуть змушені окремо одна від одної просуватися певними маршрутами з метою охоплення якомога більшої кількості об'єктів перевірки [3]. Тож для умов населених пунктів ефективність пошукових дій значною мірою обумовлюватиметься обраними маршрутами прочісування (руху) окремих пошукових груп. Крім того, ці маршрути руху повинні відповідати певним вимогам, викладеним у статті [3].

Отже, вироблення (обґрунтування) маршрутів руху для пошукових груп у межах певного населеного пункту є складним питанням, яке вирішувалося і вирішується евристичними методами на основі особистого досвіду, знань (інтуїції) командира (начальника), який приймає управлінське рішення. На мережевих структурах з великою щільністю об'єктів і зв'язків між ними зазначена задача евристичними методами розв'язуватиметься завжди неправильно (із похибками). У деяких випадках ці похибки можуть бути доволі значними. Тому наукове завдання розроблення методу маршрутизації пошукових груп на стаціонарній мережі для ефективного прочісування населених пунктів і пошуку озброєних злочинців у ході стабілізаційної операції на деокупованій території наразі є актуальним. Саме це питання розглядатиметься у статті.

Аналіз останніх досліджень і публікацій. Теоретичним базисом функціонування різних мультиагентних систем (МАС) є теорія алгоритмів. Розвиток теорії алгоритмів відбувається за двома напрямами: по-перше, розширення кола практичних задач, для розв'язання яких використовують наявні алгоритми; по-друге, розроблення і вдосконалення алгоритмів для розв'язання нових задач, що виникають у ході створення і функціонування МАС. У питаннях маршрутизації окремих агентів МАС провідну роль відіграє добре розвинена теорія графів.

Якщо підходити не строго, то задача маршрутизації пошукових груп у ході прочісування населеного пункту силами військового формування полягає у знаходженні в структурі модельного зваженого графа $G=(P,E)$ усіх (або *k*, де $k>1$) найдовших реберно-простих шляхів між обраними підмножинами вершин графа, що лежать на протилежних його окраїнах і загальна сума ваг яких (шляхів) максимізує цільову функцію F [3]. При цьому додатковою і обов'язковою вимогою до шуканого *Маршрутного плану* є відсутність спільних ребер у різних шляхів.

Наближеною до цієї і доволі відомою задачею теорії графів з чисельними практичними застосунками є задача про гамільтонові шляхи, тобто про те, є чи ні в графі простий шлях, у якому кожна вершина графа зустрічається рівно один раз. У випадку, коли граф не містить гамільтонового шляху, в деяких застосунках має сенс пошук шляху максимальної довжини в графі. Пошук такого шляху відомий як задача про найдовший шлях. Як і пошук гамільтонового шляху, пошук найдовшого шляху також є складною NP-повною задачею.

Так, у праці [4] автори доводять, що навіть якщо граф має гамільтонів шлях, задача пошуку шляху довжиною

для деякого $\varepsilon<1$ є NP-повною, де *n* – кількість вершин початкового графа. Автори стверджують, що немає поліноміального алгоритму апроксимації з постійним чинником для задачі про найдовший шлях, якщо тільки не $P=NP$ [4]. Схожі результати досліджень також наведено в публікаціях [5, 6, 7].

На відміну від задачі про гамільтонів шлях, для задачі про найдовший шлях відомо кілька алгоритмів з поліноміальною складністю, які працюють із деревами та деякими іншими класами графів. Лінійний алгоритм пошуку найдовшого шляху на структурі дерева був запропонований Дейкстрою у 1960 р., формальний опис якого можна знайти у дослідженні [8]. Пізніше, за результатами вдосконалення алгоритму Дейкстри для дерев, автори праці [9] розв'язали задачу пошуку найдовшого шляху для зважених дерев та блокових графів з лінійним часом обчислення, а для кактусів з поліноміальним часом обчислення – $O(n^2)$, де *n* – кількість вершин початкового графа. Нещодавно були запропоновані поліноміальні алгоритми, що розв'язують задачу пошуку найдовшого шляху на дводольних графах з обчислювальною складністю $O(n)$ [10], на птоломеєвих графах з обчислювальною складністю $O(n^5)$ [11]. У праці [12] автори подали свій поліноміальний алгоритм для інтервальних графів, який заснований на ідеї динамічного програмування і має обчислювальну складність $O(n^4)$. У статті [13] автори запропонували поліноміальний алгоритм, у якому також використали підхід динамічного програмування, але застосували лексикографічний пошук у глибину (так званий LDFS обхід графа) для графів порівнянності. Обчислювальна складність такого алгоритму також обмежується $O(n^4)$. Проведений аналіз літературних джерел свідчить, що задача в формальній постановці, наведена вище, ніким не ставилася і не розв'язувалася.

Метою статті є розроблення Методу маршрутизації пошукових груп на стаціонарній мережі для ефективного прочісування населених пунктів і пошуку озброєних злочинців у ході стабілізаційної операції на деокупованій території.

Виклад основного матеріалу. Проблема NP-повноти у ході пошуку саме найдовших шляхів (шляхів найбільшої сумарної ваги) у структурі мережевого об'єкта полягає у можливій наявності циклів, що призводить до невиправданого штучного збільшення його довжини (до «зациклювання»), а отже, до неможливості адекватної ідентифікації реального шляху. Для усунення такої незручності під час розв'язання задачі на максимум (пошуку найдовших шляхів) зазвичай початковий неорієнтований граф $G=(P,E)$ зображують у вигляді орієнтованого графа, що дає змогу позбутися циклів. Під час такої процедури ребрам графа надають направленість уздовж загальної направленості обходу графа (рис. 1). Унаслідок такого перетворення ребра графа стають дугами – отримують направленість, а сам граф стає орієнтованим і позначається $\vec{G}=(P,E)$.

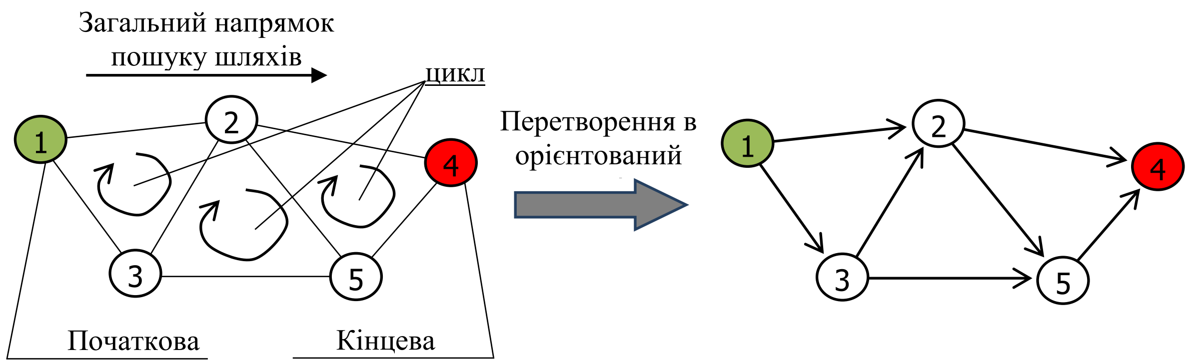
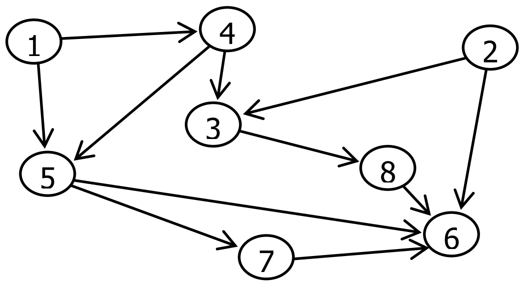


Рисунок 1 – Перетворення неорієнтованого графа в орієнтований

Оскільки прочісування населеного пункту проводитиметься від однієї його околиці до протилежної (що можна уявити у вигляді умовної «хвилі», яка проходить населеним пунктом), то така загальна направленість під час проведення пошукових заходів також має місце. З рисунка 1 видно, що в структурі орієнтованого графа вже немає циклів. Такий граф називається ациклічним.

Для пошуку критичних шляхів в ациклічному графі потрібно встановити черговість обходу його вершин. Для цього скористаємось алгоритмом топологічного сортування Кана [14]. Алгоритм знаходить у графі вершини без вхідних дуг та видаляє усі дуги, що виходять з цієї вершини, і саму вершину, фіксуючи її номер у списку S. Процес ітераційно повторюється доти, доки всі вершини не будуть переглянуті та не потраплять у список і не будуть топологічно упорядковані. Алгоритм також базується на твердженні, що в ациклічному графі завжди є принаймні одна вершина без вхідних дуг. Отже, топологічне сортування в ациклічному графі завжди можна розпочати і завершити.

Разом з тим алгоритм Кана на одному і тому ж ациклічному графі може давати кілька варіантів упорядкування вершин (рис. 2), що робить пошук екстремальних шляхів неоднозначним. З метою однозначного упорядкування вершин пропонується вдосконалити алгоритм Кана таким чином. На кожній ітерації алгоритму через p_k^- позначатимемо вершини графа, які мають тільки вихідні дуги, де k – кількість таких вершин на певній ітерації алгоритму. Під час аналізу p_k^- вершин додатково рекомендується ураховувати вагові коефіцієнти дуг і ітераційно передусім обирати ту вершину, в якої певна дуга матиме найбільшу вагу.



$$S_1 : p_1 \to p_4 \to p_2 \to p_5 \to p_3 \to p_8 \to p_7 \to p_6$$
$$S_2 : p_2 \to p_1 \to p_4 \to p_3 \to p_5 \to p_7 \to p_8 \to p_6$$
$$S_3 : p_1 \to p_2 \to p_4 \to p_5 \to p_7 \to p_3 \to p_8 \to p_6$$

Рисунок 2 – Різні варіанти топологічних сортувань вершин графа
(набір можливих варіантів неостаточний)

Така модифікація алгоритму забезпечить не тільки топологічне сортування вершин, а також їхнє упорядкування за вагою інцидентних дуг. Вершини із більшою вагою інцидентних дуг потраплятимуть у список першими, і навпаки.

У формалізованому вигляді критерій вибору вершин у процесі їхнього сортування запишеться так:

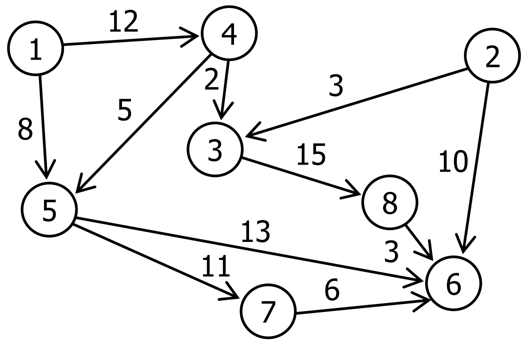
$$p_i^- \to S : \exists\, w(p_i^-, p_j) \to \max, \tag{1}$$

де p_i^- – поточна вершина, що має тільки вихідні дуги;

$w(p_i^-, p_j)$ – вага дуги, що виходить з вершини p_i^-;

S – список для зберігання вершин графа.

З урахуванням ваг дуг сортування вершин буде однозначне, що забезпечує однозначний обхід вершин графа для формування їхніх поточних вагових оцінок у ході побудови шуканих екстремальних шляхів (рис. 3).



$$S_1 : p_1 \to p_2 \to p_4 \to p_3 \to p_5 \to p_7 \to p_8 \to p_6$$

Рисунок 3 – Топологічне сортування вершин графа за модифікованим алгоритмом Кана (сортування однозначне)

Отже, на цьому етапі були вирішені два проблемні питання, пов’язані:

а) із негативним впливом циклів у структурі графа під час пошуку шляхів найбільшої довжини;

б) із порядком обходу вершин ациклічного графа для формування поточних вагових оцінок вершин графа під час пошуку шляхів найбільшої довжини.

Вирішення цих питань дало можливість приступити до розроблення Методу маршрутизації пошукових груп на стаціонарній мережі для ефективного прочісування населених пунктів і пошуку озброєних злочинців.

Нехай транспортна структура деякого населеного пункту буде подана зваженим графом G' (див. рис. 4).

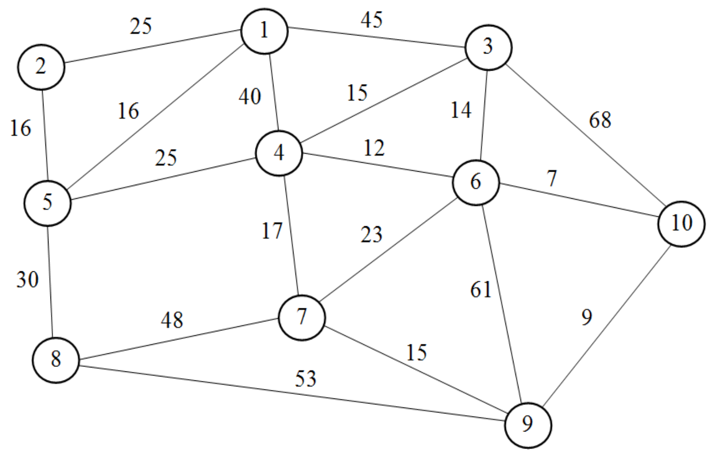


Рисунок 4 – Початковий зважений граф G', що моделює транспортну мережу населеного пункту (ваги на ребрах – кількість об’єктів перевірки на відповідній вулиці населеного пункту)

Потрібно знайти всі реберно-прості найдовші шляхи від вершин p_2 та p_5 до вершин p_9 та p_{10}, тобто вважається, що прочісування під час пошуку озброєних злочинців здійснюватиметься із заходу на схід населеного пункту. Тож до підмножини A увійдуть вершини p_2 та p_5, отже ($A = \{p_2, p_5\}$), а підмножину B наповнять вершини p_9 та p_{10}, ($B = \{p_9, p_{10}\}$).

Етап I. Надамо ребрам графа загальну направленість від вершин p_2 та p_5 у бік протилежної окраїни графа (вершини p_9 та p_{10}). Результатом цих дій стане орієнтований зважений граф $\vec{G}'$, зображений на рис. 5. Вершини, що увійшли у підмножини *A* і *B*, позначені відповідними кольорами (зеленим – вершини $p_i \in A$, червоним – вершини $p_i \in B$).

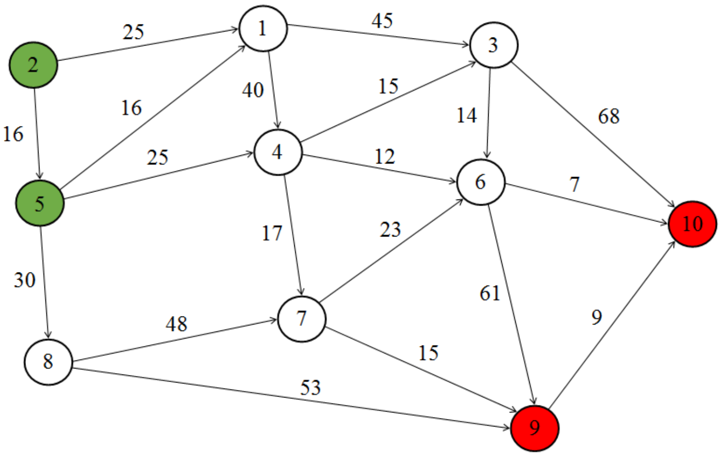


Рисунок 5 – Орієнтований зважений граф $\vec{G}'$

Етап II. Знайдемо найдовший шлях між будь-якими парами вершин, що входять до підмножин A і B відповідно. Для цього за критерієм (1) проводиться топологічне сортування вершин графа $\vec{G}'$. Його результатом є така однозначна послідовність вершин: $p_2 \to p_5 \to p_8 \to p_1 \to p_4 \to p_3 \to p_7 \to p_6 \to p_9 \to p_{10}$.

Обходячи вершини графа за цією послідовністю, кожна вершина p_j отримуватиме індивідуальну оцінку довжини шляху до неї від деякої $p_i \in A$. При цьому вагові оцінки пропонується розраховувати за виразом

$$d(p_j) := \max\{d(p_j), d(p_i^-) + w(p_i^-, p_j)\}, \tag{2}$$

де $d(p_j)$ – поточна оцінка довжини шляху до вершини, яка є суміжною із вершиною p_i^-;

$d(p_i^-)$ – поточна оцінка довжини шляху до вершини, яка обирається зі списку топологічного сортування S;

$w(p_i^-, p_j)$ – вага дуги між суміжними вершинами.

Тут потрібно зазначити, що внаслідок виконання виразу (2) на кожній ітерації вершини графа отримуватимуть максимальні оцінки довжин шляхів, які ведуть до цих вершин. Ідентифікацію самих шляхів слід проводити зворотним рахунком за отриманими оцінками.

Отже, алгоритм обходу вершин графа з формуванням поточних оцінок найдовшого шляху для кожної вершини $p_i \notin A$ на орієнтованому зваженому графі складатиметься з таких кроків.

Крок 1. Усім вершинам $p_i \in A$ присвоїти поточну оцінку $d(p_i) = 0$.

Крок 2. Решті вершин, включно і вершинам $p_i \in B$, присвоїти оцінку $d(p_i) = -\infty$.

Крок 3. Для вершини, номер якої знаходиться першим у топологічно упорядкованому за вдосконаленим алгоритмом Кана списку S, провести перерахунок за виразом (2) поточних оцінок її суміжних вершин. Видалити номер цієї вершини зі списку S.

Крок 4. Якщо в списку S не залишилося номерів вершин, вважати, що всі вершини переглянуті, всі вони отримали оцінки максимальних за довжиною шляхів, закінчити роботу алгоритму, приступати до ідентифікації шляхів. У протилежному випадку перейти до кроку 3.

Отримані в процесі роботи розробленого алгоритму поточні оцінки довжин шляхів, а також сам ідентифікований найдовший шлях у структурі початкового графа, зображені на рис. 6.

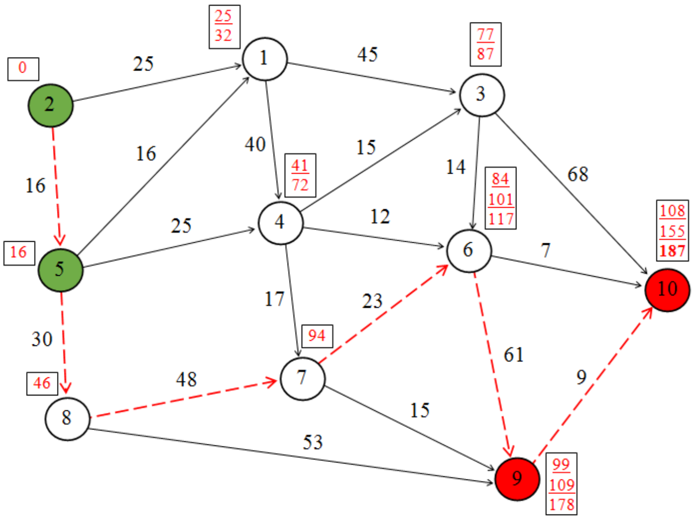


Рисунок 6 – Початковий зважений граф $\vec{G}'$ та найдовший шлях у його складі (позначений пунктиром)

За результатами роботи алгоритму отримано шлях (маршрут) до вершин p_9 та p_{10}, що належать підмножині B, а саме: $M_1 = \{p_2, p_5, p_8, p_7, p_6, p_9\}$ та $M_2 = \{p_2, p_5, p_8, p_7, p_6, p_9, p_{10}\}$. Абсолютна вага маршруту $w(M_1)$=178, а маршруту $w(M_2)$=187. Тому з цієї пари обирається найдовший маршрут M_2, йому присвоюється умовний номер M_1 (за черговістю нумерації), і цей маршрут фіксується у *Маршрутному плані*.

Етап III. Усі дуги, що складають маршрут M_1, зі структури $\vec{G}'$ видаляються. Вилучаються також усі висячі вершини, що виникають унаслідок видалення цих дуг. У результаті отримується структура $\vec{G}''$ (див. рис. 7), на якій відшукується наступний найдовший шлях. Для цього проводиться топологічне сортування вершин графа $\vec{G}''$. Його результатом є така послідовність вершин: $p_2 \to p_5 \to p_8 \to p_1 \to p_4 \to p_3 \to p_7 \to p_6 \to p_{10} \to p_9$. Здійснюється обхід вершин графа з формуванням для кожної з них власної в гової оцінки.

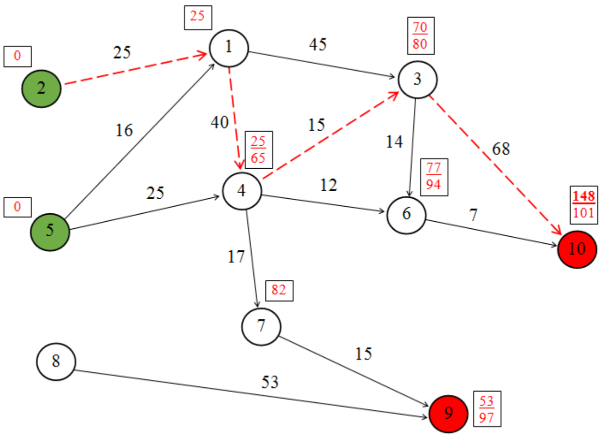


Рисунок 7 – Зважений граф $\vec{G}''$ та найдовший шлях у його структурі (позначений пунктиром)

Вершини p_9 та p_{10}, що належать підмножині B, отримали свої оцінки. Максимальна довжина маршруту до вершини p_9 становить 97 одиниць, до вершини p_{10} – 148 одиниць. Отже, обирається найдовший маршрут, яким є $M_2 = \{p_2, p_1, p_4, p_3, p_{10}\}$. Його абсолютна вага становить $w(M_2)$=148. Маршрут фіксується у *Маршрутному плані*.

Етапи IV-V. Дії за цими етапами наведені у скороченому вигляді (рис. 8).

$S: p_5 \to p_8 \to p_1 \to p_4 \to p_7 \to p_3 \to p_6 \to p_{10} \to p_9$ $\qquad$ $S: p_5 \to p_8 \to p_4 \to p_7 \to p_6 \to p_9$

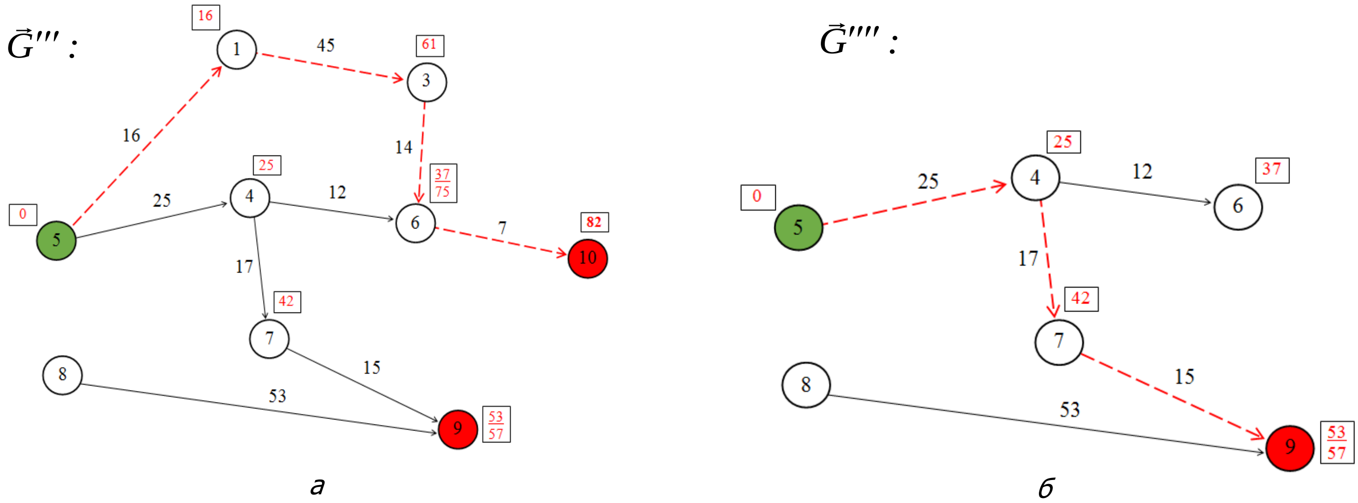


Рисунок 8 – Реберно-прості найдовші шляхи (позначені пунктиром):
а – у складі структури $\vec{G}'''$; *б* – у складі структури $\vec{G}''''$

За результатами виконання цих етапів отримуються такі маршрути: $M_3 = \{p_5, p_1, p_3, p_6, p_{10}\}$ з вагою маршруту $w(M_3)$ = 82; $M_4 = \{p_5, p_4, p_7, p_9\}$ з вагою маршруту $w(M_4)$ = 57.

Видаливши зі структури $\vec{G}''''$ маршрут M_4, можна переконатися, що шляхів між вершинами з множин A і B немає (між ними втрачене транзитивне замкнення). Це є ознакою, що були знайдені всі реберно-прості найдовші шляхи між визначеними множинами вершин у структурі початкового мережевого об'єкта $\vec{G}'$. Такі шляхи називатимемо шляхами нульового рівня, вони безпосередньо з'єднують вершини з підмножин *A* і *B*.

Отже, у результаті послідовної декомпозиції початкового графа $\vec{G}'$, а саме $\vec{G}' \to \vec{G}'' \to \vec{G}''' \to \vec{G}''''$, у його структурі знайдено чотири реберно-прості найдовші шляхи (такі, що не мають взаємних дуг), а саме: $M_1 = \{p_2, p_5, p_8, p_7, p_6, p_9, p_{10}\}$; $M_2 = \{p_2, p_1, p_4, p_3, p_{10}\}$; $M_3 = \{p_5, p_1, p_3, p_6, p_{10}\}$; $M_4 = \{p_5, p_4, p_7, p_9\}$.

Уважно вивчивши рисунки 6, 7, 8, можна дійти висновку про доцільність реінжинірингу знайдення шляхів з метою зменшення розкиду значень довжин шляхів за їх вагою. В умовах реальної службово-бойової обстановки такий реінжиніринг шляхів покликаний привести до певного балансу за навантаженням (кількістю об'єктів перевірки на маршруті руху) між різними пошуковими групами та до зменшення загального часу проведення пошукових заходів.

Суть реінжинірингу полягатиме у такому. Якщо між будь-якою парою вершин з підмножин A і/або B є дуга, яка їх з'єднує, і ця дуга увійшла до деякого екстремального шляху, що має максимальну (або більшу) оцінку довжини відносно інших шляхів, то таку дугу доцільно занести до складу шляху, що має мінімальну (або меншу) довжину.

Згідно з рисунком 6 такими дугами є (p_2, p_5) та (p_9, p_{10}). Вони входять до складу шляху $M_1 = \{p_2, p_5, p_8, p_7, p_6, p_9, p_{10}\}$, загальна довжина якого є найбільшою серед інших визначених шляхів і становить $w(M_1)$=187. Разом з тим довжина шляху $M_4 = \{p_5, p_4, p_7, p_9\}$ становить $w(M_4)$= 57. Отже, різниця за оцінками між цими шляхами буде $\Delta_{1-4} = 130$ одиниць. Тому дуги (p_2, p_5) і (p_9, p_{10}) доцільно вилучити зі складу маршруту M_1 і додати їх до складу маршруту M_4.

Після такого реінжинірингу відповідні маршрути та їхні довжини зміняться, а саме $M_1 = \{p_5, p_8, p_7, p_6, p_9\}$, $w(M_1) = 162$; $M_4 = \{p_2, p_5, p_4, p_7, p_9, p_{10}\}$, $w(M_4) = 82$, і різниця за довжиною між цими шляхами становитиме $\Delta_{1-4} = 80$ одиниць. Отже, вдалося дещо збалансувати найдовший та найкоротший шляхи на 50 одиниць. При цьому обсяг загального покриття шляхами початкової структури $\vec{G}'$ не змінився і залишився на рівні 474 одиниць. Сукупно визначені шляхи після проведеного реінжинірингу зображено на рис. 9.

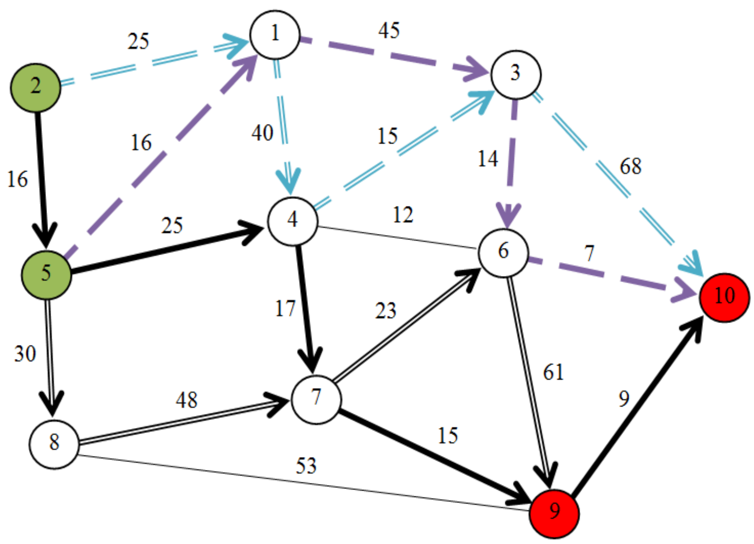


Рисунок 9 – Маршрутний план одночасного обходу початкового зваженого графа G' між обраними групами вершин за критерієм max кількості об'єктів перевірки (після реінжинірингу шляхів)

Шляхи, подані на рис. 9, є оптимізованими (максимізованими) за довжиною і є основою *Маршрутного плану* в частині порядку прочісування населеного пункту пошуковими групами.

Схема, наведена на рисунку, дає командиру (начальнику) таку інформацію.

1. Для проведення пошукових заходів у межах населеного пункту, що моделюється структурою $\vec{G}'$, потрібно не менше 4 пошукових груп.

2. Для кожної пошукової групи чітко визначена її початкова точка (рубіж) уведення у пошук (входу в населений пункт) і кінцева точка (рубіж) виведення із пошуку.

3. Кожній пошуковій групі визначено конкретний маршрут пошуку (руху) в межах певного населеного пункту. Внаслідок цього кожній пошуковій групі встановлена (відома) певна кількість об'єктів перевірки («обсяг роботи»). Маршрути пошуку різних пошукових груп не перетинаються дугами (вулицями), що дає змогу такій МАС виконувати завдання ефективно.

4. У сукупності маршрути пошуку оптимізовані як за напрямком руху, так і за кількістю об'єктів огляду (перевірки), що лежать уздовж кожного з них (у цьому випадку опосередковано оптимізований час, потрібний на проведення пошуку у населеному пункті в цілому).

Підсумкові оцінки виробленого *Маршрутного плану* за сукупністю певних показників наведено у табл. 1.

Таблиця 1 – Оцінки виробленого *Маршрутного плану* за сукупністю показників

№ пор.	Параметр (показник)	Значення показника
1	Кількість пошукових груп, що проводитимуть пошукові заходи у населеному пункті, $K_{пг}$,(шт.)	4
2	Маршрути пошуку, M_i	$M_1 = \{p_5, p_8, p_7, p_6, p_9\}$, $M_2 = \{p_2, p_1, p_4, p_3, p_{10}\}$, $M_3 = \{p_5, p_1, p_3, p_6, p_{10}\}$, $M_4 = \{p_2, p_5, p_4, p_7, p_9, p_{10}\}$
3	Кількість об'єктів огляду на маршруті, O_i, (шт.)	$O_1 = 162$; $O_2 = 148$; $O_3 = 82$; $O_4 = 82$
4	Кількість об'єктів, які будуть переглянуті у разі реалізації певного *Маршрутного плану*, O_{PL}, (шт.)	474
5	Оцінка часу проведення пошуку озброєних злочинців у населеному пункті, $T_{пш}$, (год)	27
6	Імовірність виявлення злочинців у разі реалізації певного *Маршрутного плану*, $P_{вз}$	0,88
7	Здатність угруповання провести пошукові заходи протягом директивного часу, $P_{пш}$	0,58

Примітка. Початкові дані: початковий зважений граф $\vec{G}'$; підмножини вершин $A = \{p_2, p_5\}$ і $B = \{p_9, p_{10}\}$; критерій оптимізації *Маршрутного плану* – максимум кількості перевірених об'єктів протягом директивного часу; директивний час – 25 год (1500 хв); середній час огляду одного об'єкта на маршруті пошуку – 10 хв; час висування сил угруповання НГУ до району виконання завдання – 1 год; час блокування району – 0,7 год; час висування пошукових груп на вихідне положення – 0,3 год.

Уведемо наступне поняття, а саме – ранг вершини p_i, що не входить до складу підмножин *A* та *B*. Під рангом вершини p_i розумітимемо її реберну (дугову) віддаленість від вершин, що належать підмножині *A* або *B* (рис. 10).

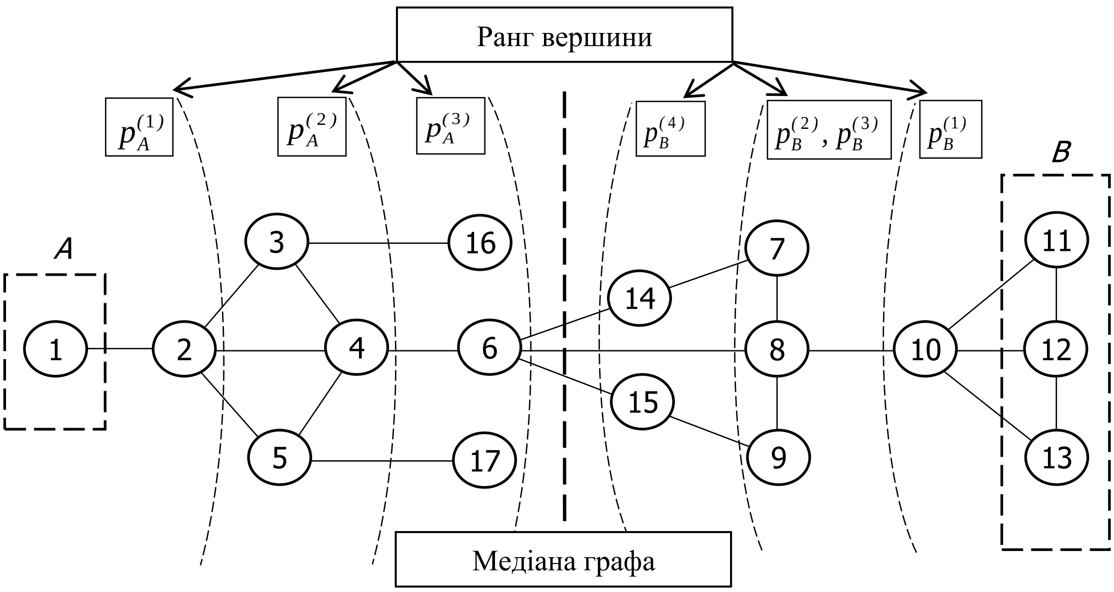


Рисунок 10 – Ранг вершин, що не належать підмножинам A та B (варіант графа)

Так, вершина p_2 має ранг, рівний одному, оскільки знаходиться від вершини $p_1 \in A$ на відстані одного ребра. Вершини p_3, p_4, p_5 мають ранг, рівний двом, оскільки знаходяться на відстані двох ребер від $p_1 \in A$. Вершини p_{14}, p_{15} мають ранг, рівний чотирьом, оскільки знаходяться на відстані чотирьох ребер від вершин $\{p_{11}, p_{12}, p_{13}\} \in B$ і т. д.

Якщо під час вироблення *Маршрутного плану* між вершинами $p_i \in A$ і вершинами $p_i \in B$ втратиться транзитивне замкнення, то в такому разі є доцільним перевірити вершини із рангами 1, 2,…, k, що знаходяться на певному віддалені від вершин $p_i \in A$ і $p_i \in B$, на наявність транзитивного замкнення між ними. У разі його наявності слід шукати екстремальні шляхи між заданими вершинами і додавати ці шляхи до *Маршрутного плану*. У такому разі початкові і кінцеві вершини таких шляхів будуть відповідно додатковими точками введення у пошук і виведення з пошуку пошукових груп. Самі такі шляхи будуть називатися відповідно шляхами 1, 2,…, k рівнів. Граничну величину рангу (k) додаткових вершин визначатиме відповідний командир (начальник), який безпосередньо на місці прийматиме рішення на проведення пошукових заходів.

З урахуванням викладених вище особливостей пошуку реберно-простих найдовших шляхів у структурі мережевого об’єкта на рис. 11 у формалізованому вигляді зображено структурну схему розробленого Методу маршрутизації пошукових груп на стаціонарній мережі для ефективного прочісування населених пунктів і пошуку озброєних злочинців у ході стабілізаційної операції на деокупованій території.

Оскільки прочісування може проводитися, за певних умов, за критерієм мінімального часу на здійснення таких заходів, розроблений Метод маршрутизації пошукових груп здатний знаходити також реберно-прості найкоротші шляхи. З цією метою у Методі використано відповідний інструмент – алгоритм Дейкстри [15] (див. рис. 11).

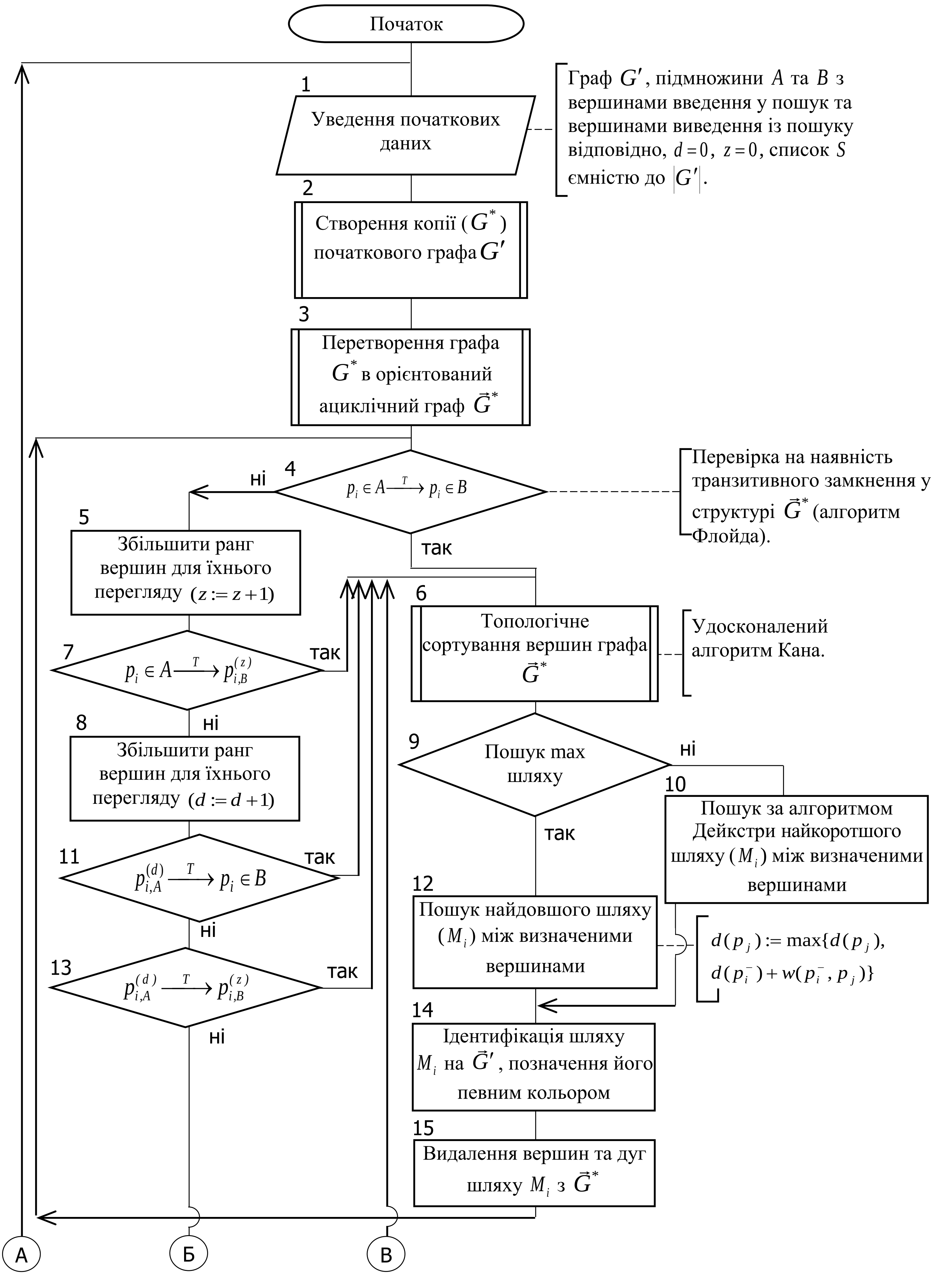


Рисунок 11 – Структурна схема Методу маршрутизації пошукових груп на стаціонарній мережі для ефективного прочісування населених пунктів і пошуку озброєних злочинців у ході стабілізаційної операції на деокупованій території

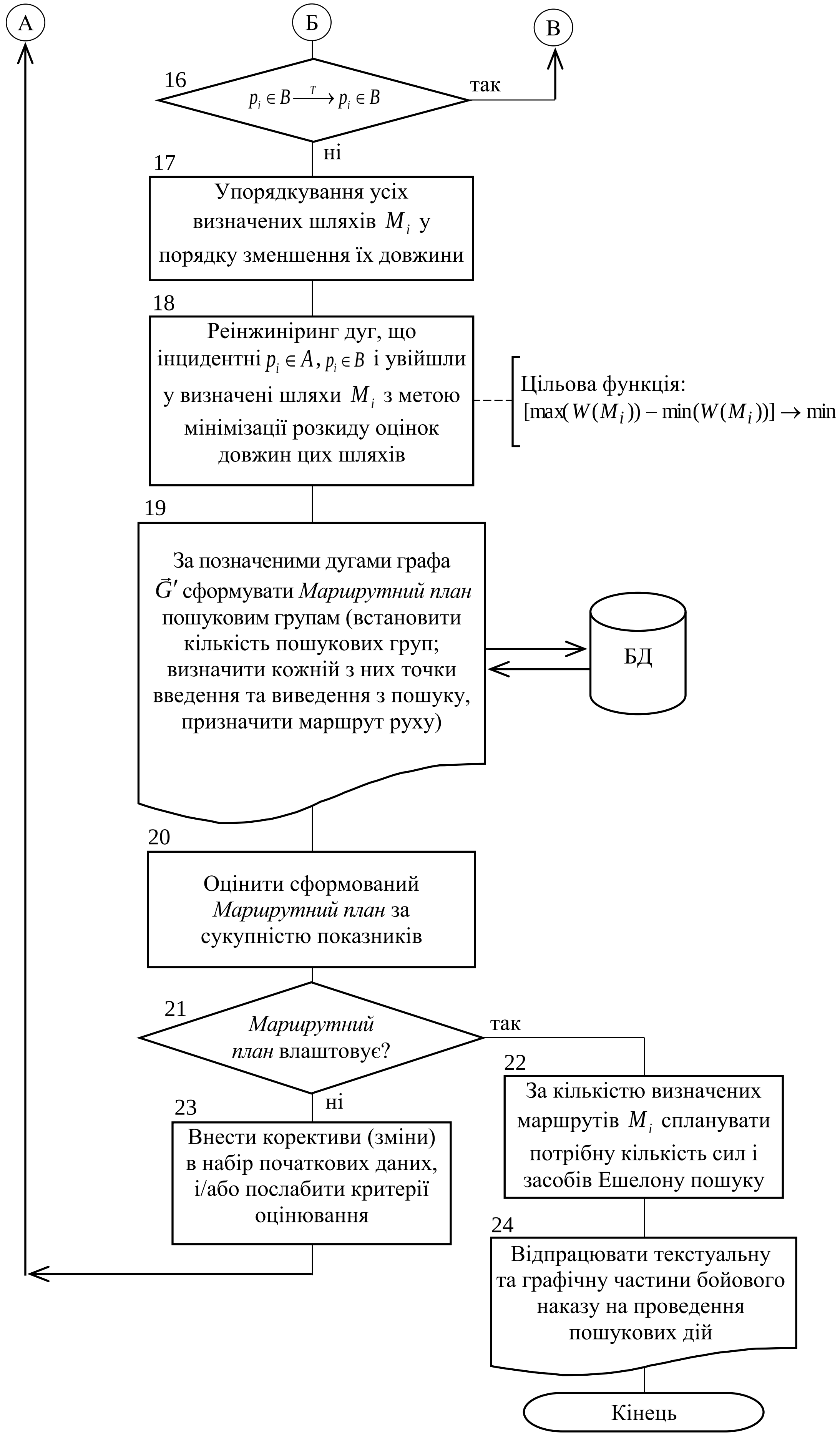


Рисунок 11, аркуш 2

За допомогою Методу орган військового управління має змогу виробити раціональні *Маршрутні плани* пошуку озброєних злочинців у населеному пункті пошуковими групами НГУ. Отримувані варіанти рішень оптимізовані за критерієм max кількості перевірених об'єктів, або за критерієм min часу на проведення пошукових заходів. Рішення про застосування того чи іншого критерію прийматиметься командиром (начальником) згідно з обстановкою, що склалася у районі виконання завдання і/або у зоні відповідальності, наявних сил і засобів, отриманого завдання від вищого органу управління.

Обчислювальна складність алгоритму, що реалізує розроблений Метод розв'язання задачі пошуку в структурі мережевого об'єкта всіх найдовших реберно-простих шляхів між визначеними парами вершин, має поліноміальну залежність і в асимптотиці оцінюється як $O(P \cdot \log P)$, де P – розмірність модельного графа. Така обчислювальна складність алгоритму обумовлена тим, що на кожній його ітерації кількість вершин графа зменшується унаслідок видалення знайдених екстремальних шляхів разом із відповідними їм вершинами. Отже, алгоритм здатний виробляти варіанти управлінських рішень у реальному масштабі часу, що є важливою властивістю для військової справи.

Висновки

Розроблений Метод доцільно використовувати в органах військового управління Національної гвардії України під час планування пошукових заходів у межах проведення стабілізаційних операцій на деокупованих територіях. Варіанти рішень, що виробляються Методом, будуть основою рішення командира на пошукові дії.

Розробленню Методу послугувало комплексне поєднання низки відомих інструментів. Так, показано, що розв'язання задачі пошуку всіх реберно-простих найдовших шляхів слід проводити поєднанням методу динамічного програмування та методу пошуку найдовших шляхів. Пошук усіх реберно-простих найкоротших шляхів здійснюється шляхом поєднання методу динамічного програмування разом із відомим алгоритмом Дейкстри.

Пошук найдовших шляхів на довільному графі є NP-повною задачею, яка наразі не має методу оптимального розв'язання у реальному масштабі часу. За своїми властивостями розроблений Метод є квазіоптимальним, оскільки його оптимальність у статті не доводилася. Алгоритм, що втілює запропонований Метод, характеризується поліноміальною складністю, завдяки чому здатен оперативно формувати варіанти управлінських рішень. Така здатність є визначальною для застосування в умовах військового управління.

Оскільки Метод розроблений у термінах теорії графів, то у своєму складі він використовує комбінаторні алгоритми та призначений для роботи із мережевими об'єктами достатньо великої розмірності й щільності зв'язків, його використання на практиці без засобів автоматизації є неможливим. Для ефективного використання Метод потребує розроблення відповідного програмного забезпечення у межах певної геоінформаційної системи.

Подальшим напрямом дослідження у межах розглянутої у статті проблематики слід вважати розроблення методу маршрутизації пошукових груп на нестаціонарній мережі.

Перелік джерел посилання

1. Про Національну гвардію України : Закон України від 13.03.2014 р. № 876-VII. *Відомості Верховної Ради України*. 2014. № 17. Ст. 594.
2. Морквін Д. А., Бацамут В. М., Гох І. М. Прогнозовані виклики, що стоятимуть перед державою у післявоєнний період: завдання НГУ із забезпечення державної безпеки та виконання правоохоронних функцій. *Безпека держави*. 2024. № 1. С. 90–98. DOI: https://doi.org/10.33405/2786-8613/2024/1/3/309959
3. Козалєтов В. В., Бацамут В. М. Проблемні питання у діяльності органів управління Сил безпеки під час планування пошукових дій у населених пунктах на деокупованій території України: шляхи вирішення. *Безпека держави*. 2025. Т. 1. № 5. С. 47–55. DOI: https://doi.org/10.33405/2786-8613/2025/1/5/336712
4. Karger D., Motwani R., Ramkumar G.D.S. On approximating the longest path in a graph, *Algorithmica*. 1997. 18. pp. 82–98. DOI: https://doi.org/10.1007/BF02523689.
5. Feder T., Motwani R. Finding large cycles in Hamiltonian graphs, Proc. of the 16th annual ACM-SIAM Symp. on Discrete Algorithms (SODA), ACM. 2005. P. 166–175. DOI: https://doi.org/10.1016/j.dam.2009.12.006.

6. Gabow H., Nie S. Finding long paths, cycles and circuits, *Proc. of the 19th annual International Symp. on Algorithms and Computation (ISAAC)*, LNCS 5369. 2008. P. 752–763. DOI: https://doi.org/10.1007/978-3-540-92182-0_66.

7. Zhang Z., Li H. Algorithms for long paths in graphs, *Theoret. Comput. Sci.* 2007. 377. P. 25–34. DOI: https://doi.org/10.1016/j.tcs.2007.02.012.

8. Bulterman R., van der Sommen F., Zwaan G., Verhoeff T. et al. On computing a longest path in a tree, *Inform. Proc. Lett.* 2002. 81. P. 93–96. DOI: https://doi.org/10.1016/S0020-0190(01)00198-3.

9. Uehara R., Uno Y. Efficient algorithms for the longest path problem. *Proc. of the 15th annual International Symp. on Algorithms and Computation (ISAAC)*, LNCS 3341. 2004. P. 871–883. DOI: https://doi.org/10.1007/978-3-540-30551-4_74.

10. Uehara R., Valiente G. Linear structure of bipartite permutation graphs and the longest path problem. *Inform. Proc. Lett.* 2007. 103. P. 71–77. DOI: https://doi.org/10.1016/j.ipl.2007.02.010.

11. Takahara Y., Teramoto S., Uehara R. Longest path problems on ptolemaic graphs, *IEICE Trans. Inf. and Syst.* 91-D. 2008. P. 170–177. DOI: https://doi.org/10.1093/ietisy/e91-d.2.170.

12. Ioannidou K., Mertzios G., Nikolopoulos S. The Longest Path Problem has a Polynomial Solution on Interval Graphs, *Algorithmica*. 2011. Vol. 61. P. 320–341. DOI: https://doi.org/10.1007/s00453-010-9411-3.

13. Mertzios G. B., Corneil D. G. A Simple Polynomial Algorithm for the Longest Path Problem on Cocomparability Graphs, *SIAM Journal on Discrete Mathematics*. 2012. Vol. 26. Iss. 3. P. 940–963. DOI: https://doi.org/10.1137/100793529.

14. Arthur B. Kahn. Topological sorting of large networks. *Communications of the ACM*. 1962. 5 (11). P. 558–562. DOI: https://doi.org/10.1145/368996.369025.

15. Dijkstra E. W. A note on two problems in connexion with graphs. *Numerische Mathematik*. 1959. Vol. 1. Iss. 1. P. 269–271. https://doi.org/10.1007/BF01386390.

Стаття надійшла до редакції 19.12.2025 р.
Прийнято до друку після рецензування 16.01.2026 р.
Дата публікації 29.05.2026 р.

UDC 519.853: 658.52

V. Kozalietov, V. Batsamut

METHOD OF ROUTING SEARCH GROUPS ON A FIXED NETWORK FOR EFFECTIVE SWEEPING OF POPULATED AREAS AND SEARCHING FOR ARMED CRIMINALS

The relevance of this study is driven by the need for the further development of collective behavior models for systems with a multi-agent organizational structure, as well as for equipping such systems with intelligence capable of ensuring the synchronization of joint efforts of multiple agents in achieving system-level objectives.

The proposed method addresses the problem of inefficient traversal of populated areas by search teams (agents within a multi-agent system), which is critically important for the execution of search, rescue, and monitoring tasks in crisis-affected areas of various origins.

The proposed method is based on the concept of dynamic programming, which consists in the successive decomposition of the initial structure of a settlement's transportation network into a set of substructures. Within each substructure, an edge-simple longest/shortest path between a specified pair of vertices is determined.

The decomposition process continues until the transitive connectivity between the defined entry and exit points of the search within the settlement structure is no longer preserved. The set of identified extremal paths forms a routing plan for search teams during search operations, which, in a certain sense, organizes and systematizes the sweeping of the populated area.

By its nature, the problem under consideration is NP-complete, and the proposed method belongs to the class of quasi-optimal approaches. A polynomial estimate of the computational complexity of the combinatorial algorithm implementing this method has been substantiated.

The method has been developed and presented in terms of graph theory and set theory and is formalized within the article. The procedure for generating a routing plan for search teams using the proposed method is demonstrated through a simplified practical example.

The method significantly reduces the labor intensity associated with developing a rational variant of search operations and is intended for use by military command and control bodies of the National Guard of Ukraine in planning search activities within a designated populated area in de-occupied territories.

Keywords: *area sweeping, search teams, multi-agent system, routing; network object, weighted undirected (directed) graph, extremal paths, optimization criterion, method*

Козалєтов Віталій Валерійович – ад'юнкт, Національна академія Національної гвардії України
https://orcid.org/0009-0007-9540-8277

Бацамут Володимир Миколайович – доктор військових наук, професор, заступник начальника інституту з наукової роботи – начальник науково-дослідної лабораторії службово-бойового застосування НГУ навчально-наукового інституту забезпечення державної безпеки, Національна академія Національної гвардії України
https://orcid.org/0000-0003-2182-6891