УДК 519.816

Саланда І.П., аспірант; **Барабаш О.В.,** д.т.н.; **Мусієнко А.П.,** к.ф.-м.н.

МЕТОДИ ПОШУКУ ОПТИМАЛЬНИХ МАРШРУТІВ ГРАФА СТРУКТУРИ РОЗГАЛУЖЕНОЇ ІНФОРМАЦІЙНОЇ МЕРЕЖІ ЗА ЗАДАНИМ КРИТЕРІЄМ ОПТИМАЛЬНОСТІ ПРИ РІЗНИХ ОБМЕЖЕННЯХ

Salanda I.P., Barabash O.V., Musienko A.P. Methods of search of optimum routes of the graph structure of branched network information for a given optimality criterion under various constraints. During the service information network should consider its future modifications, alterations, emergence of new nodes or lines, possible failure of servers that the functional stability of the network. This is necessary as soon as possible to adapt the router. This is due to research in this paper.

At work the approximate algorithms for finding optimal routes graphs independent information networks that improve performance branching information networks.

Approximate algorithm for finding optimal routes helps reduce the dimension of the problem of finding optimal routes and reduce the complexity of solving the problem of finding the shortest routes, subject to certain requirements structure information network. These algorithms appropriate to apply a relatively large corporate computer networks with the need for independent optimal ways for the time linearly depends on the number of vertices.

In addition, the work can be used in the construction of complex technical systems, distributed information systems and video surveillance systems, the integrated systems of technical protection of information.

Keywords: optimal route, extensive information network, search algorithm.

Саланда І.П., Барабаш О.В., Мусієнко А.П. Методи пошуку оптимальних маршрутів графа структури розгалуженої інформаційної мережі за заданим критерієм оптимальності при різних обмеженнях. В даній роботі розроблено наближені алгоритми пошуку оптимальних незалежних маршрутів графа інформаційної мережі, що дозволяють підвищити швидкодію розгалужених інформаційних мереж.

Наближений алгоритм пошуку оптимальних маршрутів дозволяє зменшити розмірність задачі знаходження оптимальних маршрутів і скоротити трудомісткість рішення задачі пошуку найкоротших шляхів з урахуванням певних вимог до структури інформаційної мережі. Дані алгоритми доцільно застосовувати в порівняно великих корпоративних обчислювальних мережах при необхідності отримання оптимальних незалежних шляхів за час, який лінійно залежить від кількості вершин графа.

Ключові слова: оптимальний маршрут, розгалужена інформаційна мережа, алгоритм пошуку.

Саланда И.П., Барабаш О.В., Мусиенко А.П. Методы поиска оптимальных маршрутов графа структуры разветвленной информационной сети по заданному критерию оптимальности при различных ограничениях. В данной работе разработаны приближенные алгоритмы поиска оптимальных независимых маршрутов графа информационной сети, которые позволяют повысить быстродействие разветвленных информационных сетей.

Приближенный алгоритм поиска оптимальных маршрутов позволяет уменьшить размерность задачи нахождения оптимальных маршрутов и сократить трудоемкость решения задачи поиска кратчайших путей с учетом определенных требований к структуре информационной сети. Данные алгоритмы целесообразно применять в сравнительно больших корпоративных вычислительных сетях при необходимости получения оптимальных независимых путей за время, которое линейно зависит от количества вершин графа.

Ключевые слова: оптимальный маршрут, разветвленная информационная сеть, алгоритм поиска.

Вступ

Високі вимоги, що ставляться до ефективності, надійності і швидкості передачі даних в розгалужених інформаційних мережах (РІМ) сприяють появі нових методів і алгоритмів маршрутизації та надійності мережевого обладнання. Для своєчасної та безпечної доставки інформації в мережі важливу роль відіграє оптимальна маршрутизація, яка стійка до різних відмов, ефективна в умовах підвищеної завантаженості мережі та запобігає несанкціонованому доступу до інформації.

В процесі обслуговування інформаційної мережі слід враховувати її модифікації в майбутньому, перепланування, появу нових вузлів або ліній зв'язку, можливі відмови серверів, тобто функціональну стійкість мережі. При цьому виникає необхідність за найкоротші терміни адаптувати маршрутизатор. Цим обумовлюється розробка найбільш ефективних алгоритмів пошуку оптимальних маршрутів від вузла до вузла, які враховують можливі зміни РІМ.

Постановка завдання в загальному вигляді. Математична модель комп'ютерної мережі може бути представлена у вигляді графа, вершини якого моделюють вузли-джерел та вузли-приймачів інформації, а гілки графа відповідають можливим каналам передачі даних. Гілкам графа присвоюють певні значення. Коли потрібно мінімізувати час затримки передачі даних необхідно вирішувати задачу про найкоротший шлях між вузлом-джерелом і вузлом-приймачем інформації.

В задачі про найкоротший маршрут (shortest-paths problem) задається зважений орієнтований граф *G=(X,U)* із функцією ваги $l: U \to R$. Вага маршруту $\{\mu_j\}$ дорівнює сумарній вазі ребер, що в нього входять.

Вага найкоротшого маршруту (shortest-path weight) із вершини s у вершину t визначається співвідношенням: $\delta(s,t) = \min\left\{ l(\mu_r) : s \xrightarrow{\mu_r} t \right\}$, якщо існує маршрут від s до t, і $\delta(s,t)=\infty$ – в противному випадку.

Тоді за означенням найкоротший маршрут (shortest path) із вершини s у вершину t – це будь-який маршрут, вага якого задовольняє співвідношенню $l(\mu_r)=\delta(s, t)$.

Вагу кожного ребра можна інтерпретувати не як відстань, а як іншу метрику. Часто вони використовуються для подання тимчасових інтервалів, вартості, штрафів, збитків або будь-якої іншої величини, яка лінійно накопичується в міру просування вздовж ребер графа і яку потрібно звести до мінімуму.

Алгоритм дозволяє вирішити багато інших задач, в тому числі ті, що перераховані нижче.

1) Задачу про найкоротший шлях в заданий пункт призначення (single-destination shortest-paths problem).

2) Задачу про найкоротший шлях між заданою парою вершин (single-pair shortest-paths problem).

3) Задачу про найкоротший шлях між усіма парами вершин (all-pairs shortest-paths problem).

Області застосування. Задача побудови оптимального маршруту зустрічається в різних областях. Дані задачі можна використовувати при побудові складних технічних систем [1], розподілених інформаційних систем та в системах відеоконтролю [2-6], в комплексних системах технічного захисту інформації [7,8].

Аналіз досліджень та публікацій. У задачі побудови оптимального маршруту історія довга і цікава історія. Грехем і Хелл в своїй статті «Історія задачі побудови мінімального остового дерева» [9] починають відлік історії проблеми з роботи Чекановського (Czekanowski) 1909 року. Перший і, напевно, найпростіший алгоритм пошуку мінімального остового дерева належить Борувці (Boruvka), який в 1926 році, набагато раніше, ніж з'явилися перші комп'ютери, і навіть раніше, ніж була створена конструктивна теорія графів, представив своє рішення даної задачі. Трохи пізніше, в 1938 році Шоку (Choquet), а потім Флорек (Florek), Лукасевич (Lukaziewicz), Перкал (Perkal), Штейнгауз (Stienhaus) і Зубжицький (Zubrzycki) в 1951 році і Солліне (Sollin) на початку шістдесятих знову перевідкривається алгоритм.

Серед найвідоміших методів пошуку найкоротшого маршруту можна виділити наступні алгоритми: Беллмана-Форда, Дейкстри, Флойда Уоршала.

Алгоритм Беллмана-Форда носить ім'я двох американських вчених: Річарда Беллмана (Richard Bellman) і Лестера Форда (Lester Ford). Форд фактично винайшов цей алгоритм в 1956 році при вивченні іншої математичної задачі, підзадача якої звелася до пошуку найкоротшого шляху на графі, і Форд зробив начерк остаточного вирішення завдання цього алгоритму. Беллман в 1958 році опублікував статтю, присвячену конкретно завданню знаходження найкоротшого шляху, і в цій статті він чітко сформулював алгоритм у тому вигляді, в якому він відомий нам зараз. Алгоритм маршрутизації RIP (алгоритм Беллмана-Форда) був вперше розроблений в 1969 році, як основний для мережі ARPANET.

Другий найбільш відомий підхід носить назву алгоритм Дейкстри – це алгоритм на графах, винайдений нідерландським вченим Едсгером Дейкстром в 1959 році. Його використовують протоколи маршрутизації OSPF і IS-IS.

Інший відомий алгоритм Флойда-Воршала – це приклад динамічного програмування. Він був опублікований у звичній сьогодні формі Робертом Флойдом у 1962 році. Проте, це практично той же алгоритм, що був опублікований Бернардом Роєм у 1959 році та Стефаном Маршалом у 1962 році для знаходження транзитивного замикання в графі, і є досить тісно пов'язаним з алгоритмом Кліні (опублікованим у 1956 році) для перетворення детермінованих скінченних автоматів у регулярні вирази. Сучасне формулювання алгоритму, як трьох вкладених циклів було вперше подано Пітером Інгерманом у 1962 році.

Відомий також алгоритм Левіта, опублікований у 1971 році, який є аналогом алгоритму Дейкстри, та алгоритм Джонсона (1977 році) що базується на алгоритмах Беллмана-Форда і Дейкстри.

Починаючи з 80-х років було запропоновано багато різних підходів для оптимізації і поліпшення швидкості роботи відомих алгоритмів.

Проблемами удосконалення методів і алгоритмів маршрутизації в інформаційних мережах займалися такі вчені, як Д. Бертсекас, Д. Гарсіа-Діас, П. Гупта, А.Б. Гольдштейн, Б.С. Гольдштейн, Д. Кантор, О.Я. Кравець, Д.В. Куракін, І.П. Норенков, А. Філіпс, С. Флойд. Завдання знаходження найкоротших шляхів в транспортній системі розглядали в своїх працях вчених Л. Беллман, Г. Габов, С. Гудман, Е. Дейкстра, В.А. Євстигнєєв, В.Н. Касьянов, Р. Седжвік, Р. Тар'я, С. Флойд, Л. Форд, Д. Фулкерсон. Детальний опис методів пошуку найкоротших шляхів можна знайти в роботах Т. Кормена, Ч. Лейзерсон, Р. Ривеста та інших.

Базові алгоритми пошуку найкоротшого маршруту. Алгоритм Беллмана-Форда дозволяє вирішити задачу про найкоротший шлях з фіксованого витоку в загальному випадку, коли вага будь-якого ребра може бути від'ємною. Цей алгоритм відрізняється своєю простотою. До його переваг також відноситься те, що він визначає, чи міститься в графі цикл з від'ємною вагою, досяжний з витоку. Час виконання – O ($|X|\times|U|$). Якщо граф заданий матрицею суміжності, то алгоритм буде виконуватись за $O(|U|^3)$ часу.

На відмінну від попереднього, алгоритм Дейкстри характеризується меншим часом виконання, але вимагає, щоб вага кожного ребра була невід'ємною. Даний метод знаходить найкоротший шлях від однієї вершини графа до всіх інших вершин. Найпростіша реалізація алгоритму Дейкстри потребує $O(|X|^2)$ дій, а найгірша швидкодія – $O(|X| \times |U| log|X|)$.

В комп'ютерних науках, алгоритм Флойда-Воршелла використовується для вирішення задачі про найкоротший шлях між усіма парами вершин у зваженому графі з додатними або від'ємними вагами ребер (але без від'ємнозначних циклів). При звичайній реалізації алгоритм видасть сумарні ваги найкоротших шляхів між всіма парами вершин, хоча він не поверне інформацію про самі шляхи. Алгоритм Флойда-Воршалла є хорошим для обчислення шляху між усіма парами вершин в щільних графах, в яких більшість або всі пари вершин, з'єднані ребрами. Для розріджених графів з невід'ємними вагами ребер, найкраще використовувати алгоритм Дейкстри від кожної можливої вихідної вершини, оскільки складність Дейкстри $O(|X|^2)$ є кращою, ніж $O(|X|^3)$ – складність алгоритму Флойда-Воршала.

Вище коротко наведені лише базові алгоритми. Ознайомитися з найбільш відомими алгоритмами можна за статтями зі списку використаної літератури. Не дивлячись на велику кількість публікацій, проблема пошуку оптимального маршруту є досить різнобічна, та залишається актуальною й продовжує досліджуватися сьогодні.

Метою даної статі є удосконалення існуючих методів пошуку найкоротшого маршруту та розробка наближеного алгоритму пошуку оптимальних незалежних маршрутів графа інформаційної мережі, який відрізняється від існуючих швидкодією.

Основна частина

При синтезу мережі із заданими властивостями виникає наступна важлива задача.

Нехай задано зважений граф $G(X,U)$, в якому виділена пара вершин $s,t \in X$. Кожному ребру $u \in U$ поставлено у відповідність число $l(u)$.

В графі $G(X,U)$ між вершинами s і t необхідно знайти k ланцюгів $\{\mu_j\}$, таких, щоб наступний функціонал набув найменшого значення:

$$f = \sum_{j=1}^{k} \sum_{u_i \in \mu_j} l(u_i) \tag{1}$$

при умовах, що $\{\mu_j\}_k$ – незалежні за вершинами (ребрами), число ребер в μ_j не перевищує діаметру d графа G.

Наближені алгоритми пошуку оптимальних незалежних ланцюгів. Розв'яжемо дану задачу з припущенням, що $k = 2$ і без обмеження на діаметр. Рішення цієї задачі необхідно починати з пошуку найкоротшого маршруту в графі $G=(X,U)$.

Алгоритм I.

Крок 1. Побудова найкоротших маршрутів. Між вершинами s і t з врахуванням позначених вершин і ребер графа G шукаємо найкоротший маршрут, тобто позначені елементи графа не входять в знайдений маршрут. Спочатку всі елементи графа G не позначені. Між вершинами s і t знаходиться найкоротший маршрут $\eta=(s, x_1, x_2,\ldots, x_p, t)$. Вершини і ребра маршруту η позначаються в G. Отримаємо граф G_0.

Крок 2. Пошук січних ланцюгів:

2.0. В графі G_0 знаходимо найкоротші ланцюги (процедура кроку 1) між вершиною s і вершинами η. Якщо існує маршрут між вершинами s і t в G_0, то $\varphi_1 := \rho_{G_0}(s,t) + \rho_G(s,t)$, $\varphi_2 := \infty$. При $\rho_{G_0}(s,t) = \rho_G(s,t)$, знайдені маршрути оптимальні. Виберемо найближчу до t вершину $x_k \in \eta$, для якої досягається мінімум

$$\rho_{G_0}(s,x_k) - \rho_G(s,x_k). \tag{2}$$

Вершини і ребра ланцюга $\mu_1 = \overline{(s,x_k)}$ позначаються. Отримаємо позначений граф G_1.

2.1. В G_1 між вершинами $s, x_1^1,\ldots,x_{k-1}^1$ і $x_{k+1}^1,\ldots,x_p^1, t$ знаходимо найкоротші ланцюги, тобто знайдемо матрицю відстаней в графі $G_1 : \left\{\rho_{G_1}(x_f^1, x_r^1)\right\}$, де $f \le k-1, r \ge k+1$.

Якщо $\forall x_f^1, x_z^1 : \rho_{G_1}(x_f^1, x_z^1) = \infty$, то в G_0 знайдеться вершина x_1, для якої справедлива умова (2), і знову переходимо до 2.1. Така вершина обов'язково знайдеться, якщо вершини s і t 2-локально зв'язні в G.

Виберемо пару вершин x_f^1, x_r^1, де x_r^1 – найближча до t при умові досягнення мінімуму різниці:

$$\min_{f,r}\left(\rho_{G_1}(x_f^1, x_r^1) - \rho_G(x_f^1, x_r^1)\right). \tag{3}$$

Якщо $x_r^1 = t$, то процес закінчується і ланцюг $\mu_2 = (x_t^1, t)$ січний. Переходимо до процедури кроку 3. Якщо $\rho_{G_1}(x_f^1, t) = \infty$, то будуються незалежні маршрути (процедура кроку 3) і $\varphi_2 := \varphi$. При $\varphi_1 > \varphi_2$, $\varphi_1 := \varphi_2$ і ці маршрути запам'ятовуються. Нехай для вершин x_f^1, x_r^1 виконується умова (3), тоді вершини і ребра ланцюга $\mu_2 = (x_f^1, x_r^1)$ позначаються. Отримаємо граф G_2.

2.k. Нехай дано позначений граф G_k та вершини x_d^{k-2} і x_r^{k-1} $(d < r)$, які є кінцями (найближчими до t) січних ланцюгів, побудованих за допомогою двох попередніх ітерацій. Між множинами вершин $\{x_d^{k-2}, x_{d+1}^{k-2}, ..., x_{r-1}^{k-2}\}$ і $\{x_{r+1}^{k-1}, ..., x_p^{k-1}, t\}$ знаходимо найкоротші ланцюги і будуємо матрицю відстаней $\{\rho_{G_k}(x_f^k, x_r^k)\}$. Далі виконуємо ті ж дії, що і в кроці 2.2, за виключенням того, що у випадку необхідності повернення здійснюється не до 2.1, а до 2.k-1 і кінець січного ланцюга вибирається правіше ніж x_r^{k-1} .

Крок 3. Побудова незалежних маршрутів. Виходимо із вершини s за позначеними ребрами до ланцюга μ_1, потім рухаємося по μ_1 до вершини з η і знову – по η до t і так далі. Потім починаємо рухатися від вершини s по η до січного ланцюга μ_2, за яким доходимо до вершини x_r^1 із η. Від вершини x_r^1 рухаємося по η в напрямку t до ланцюга μ_3 і т.д.

Побудовані маршрути η_1 і η_2 будуть оптимальні, якщо для будь-якого k і будь-яких вершин x_f, x_r (які є кінцями μ_k): $\rho_{G_k}(x_f^k, x_r^k) = \rho_G(x_f, x_r)$. В противному випадку їх оптимальність не гарантується. Нехай довжина ланцюга η_1 більша за η_2, тобто $\rho(\eta_1) \geq \rho(\eta_2)$. Тоді $\varphi := \rho(\eta_1) r_1 + \rho(\eta_2) r_2$.

Тим самим задачу вирішено.

На практиці в даній задачі можна знехтувати вагою ребер. Тоді задача матиме наступний вигляд: в неорієнтованому графі $G=(X,U)$ знайти k незалежних за вершинами (ребрами) обмежених (s, t)-ланцюгів.

Якщо $k = 2$ і $d \leq 6$, то ця задача для довільних $l(u)$ вирішується за допомогою алгоритму II.

Алгоритм II.

Крок 1. Якщо $d \geq n-1$, то задача вирішується звичайним методом. Якщо $d < \rho(s,t)$, то допустимих шляхів не існує. Якщо $d=\rho(s,t)$, то на крок 3, інше на крок 2.

Крок 2. Так як вершини s і t знаходяться на відстані d в графі G, то довжина (s,t)-лагцюга рівна d. Відповідно, граф $G=(X,U)$ можна перетворити в орієнтований граф $G'=(X',U')$, такий, що розв'язанню задачі на графі G взаємно однозначно відповідає розв'язок на орграфі G'.

Видалимо з G вершини $\{x_i\}$, для яких $\rho(s, x_i) \geq d$, $t \notin \{x_i\}$.

Для кожної пари суміжних вершин x_i і x_j, для яких $\rho(s,x_i)=\rho(s,x_j)$, видалимо ребро, що їх з'єднує. Ребра, що залишились орієнтуємо від s до t, тобто ребро $u=(y,z)$ орієнтоване від s до t тоді і лише тоді, коли $\rho(s,y)<\rho(s,z)$. Далі в орграфі G' за допомогою відомих методів знаходимо незалежні маршрути між вершинами s і t.

Очевидно, що довжина кожного такого маршруту не перевищує d.

Крок 3. Нехай $d>\rho(s,t)$. Тоді для зменшення перебору з графа G можна видалити вершини $\{x_i\}$, для яких $\rho(s,x_i)>\left[\dfrac{d+\rho(s,t)}{2}\right]$. В отриманому графі $G''=(X'',U'')$, використовуючи модифікований метод, будемо послідовно збільшувати кількість незалежних за вершинами (s,t)-ланцюгів, які задовольняють обмеженню на довжину кожного ланцюга, так як для побудови незалежних за ребрами (s,t)-ланцюгів використовується аналогічний алгоритм. В останньому випадку виключенням є спосіб побудови зворотного січного маршруту. В даному випадку, потрапивши в побудований маршрут по січному ланцюгу, необов'язково повертатись назад і, крім того, січні ланцюги можуть перетинатися вершинами.

Крок 4. Між вершинами s і t графа $G''=(X'',U'')$ знайти найкоротший ланцюг μ_1. Очевидно, $\rho(\mu_1)=\rho(s,t)\leq d$. В підграфі $G_1''=(X_1'',U_1'')$, в якому відсутні не кінцеві вершини μ_1, знайти найкоротший (s,t)-ланцюг в підграфі G_2'', G_3'', ….

Крок 5. Якщо кількість побудованих ланцюгів дорівнює заданому числу або мінімальному перетину по вершинам, то переходимо на крок 10, інакше на крок 6.

Крок 6. В кожному ланцюзі $\mu_i=(s=x_1^i,\dots,x_2^i=t)$ будь-якій вершині x_j^i порівняємо два числа: $\gamma(x_j^i)=\rho_{\mu_i}(s,x_j^i)$ і $\beta(x_j^i)=\rho_{\mu_i}(x_j^i,t)$ тобто числове значення довжин відповідних частин ланцюга μ_i.

Крок 7. Починаючи з вершини s, побудуємо зворотній січний маршрут, дотримуючись правила: при виході з чергової вершини x_j^i побудованої $\mu_i(s,t)$-ланцюга і вході у вершину x_l^k іншого (але можливо і того ж) (s,t)-ланцюга μ_k для січного ланцюга (x_j^i,x_l^k) має виконуватись нерівність: $\gamma(x_j^i)+\bar{\rho}(x_j^i,x_l^k)+\beta(x_l^k)\leq d$, де $\bar{\rho}(x_j^i,x_l^k)-$ довжина відповідної частини січного ланцюга. Якщо ця нерівність виконується, то вершинам x_j^i,x_l^k підставляються наступні пари чисел:

$$\bar{\gamma}(x_j^i):=\gamma(x_j^i),\ \ \bar{\beta}(x_j^i):=\bar{\rho}(x_j^i,x_l^k)+\beta(x_l^k),$$
$$\bar{\gamma}(x_l^k):=\bar{\rho}(x_j^i,x_l^k)+\gamma(x_j^i),\ \ \bar{\beta}(x_l^k):=\beta(x_l^k).$$

Крім того, для всіх вершин $\{y_j\}$, які належать ланцюгу $(\overline{x_l^k,t})$ (цей ланцюг необов'язково лежить в μ_k, тобто він може складатися з відрізків $\{\mu_l\}$ і раніше побудованих січних ланцюгів), змінимо значення $\gamma(y_j)$ за допомогою оператора:

$$\bar{\gamma}(y_j):=\gamma(y_j+\bar{\gamma}(x_l^k)-\gamma(x_l^k)).$$

Аналогічно, для всіх вершин $\{z_f\}$, що належать ланцюгу $(\overline{s,x_j^i})$, змінюються значення для $\beta(z_f)$ за правилом:

$$\overline{\beta}(z_f) = \beta(z_f) + (\overline{\beta}(x_f^i) - \beta(x_j^i)).$$

Таким чином, усім необхідним вершинам на $\{\mu_i\}$ поставлено у відповідність нові пари чисел γ і β.

Якщо надалі виявиться, що дана частина січного ланцюга заборонена, тобто подальша побудова ланцюга неможлива, то значення відповідних пар чисел відновлюються. Як тільки січний ланцюг досягає вершини t, процес побудови зворотного січного маршруту закінчиться на кроці 8. В противному випадку переходимо на крок 9.

Крок 8. Далі здійснюється збільшення числа незалежних (s,t)-ланцюгів на одиницю (див. крок 3 алгоритму I); переходимо на крок 10.

Крок 9. Задане число незалежних і обмежених (s,t)-ланцюгів знайти не вдається.

Крок 10. Задане число незалежних і обмежених ланцюгів знайдено.

Для графів, що мають відносно невеликий діаметр можна запропонувати спрощену процедуру побудови незалежних маршрутів мінімальної довжини.

Алгоритм III.

Крок 1. В графі G знаходиться мінімальний маршрут $\theta=(x_i, y_1,\dots, y_p, x_j)$ між вершинами x_i і x_j. Ребра і вершини цього маршруту позначаються. Отримаємо граф G': $\varphi_1 = \infty,\ \varphi_2 = 0$.

Крок k. Виберемо k-ту пару вершин y_f, y_r із $\{y_i\}$ і в G' знайдемо найкоротший ланцюг $\mu_k = \overline{(y_f, y_r)}$. В графі G відмітимо вершини і ребра ланцюгів $(x_i, y_1,\dots, y_f)$, $(y_r, y_{r+1},\dots, x_j)$, μ_k. Отримаємо граф G_k. В G_k знайдемо найкоротший ланцюг η_k між вершинами x_i і x_j:

$$\varphi_2 := \rho_G(x_i, y_f) + \rho_G(y_f, y_r) + \rho_G(y_r, x_j) + \rho_{G_k}(x_i, x_j).$$

Якщо $\varphi_2 < \varphi_1$, то $\varphi_1 := \varphi_2$; замінити пару вершин y_f, y_r.

Після закінчення роботи алгоритму в φ_1 міститиметься значення сумарної довжини знайдених маршрутів; визначаються вони однозначно парою вершин y_f, y_r.

Складність даного алгоритму $C_{10}^2 \cdot n$, де n – кількість вершин графа, що відрізняється від традиційних методів, у яких найменша швидкодія $O(n^2)$.

Висновок

В даній роботі розроблено наближені алгоритми пошуку оптимальних незалежних маршрутів графа інформаційної мережі, що дозволяє підвищити швидкодію розгалужених інформаційних мереж.

Наближений алгоритм пошуку оптимальних маршрутів дозволяє зменшити розмірність задачі знаходження оптимальних маршрутів і скоротити трудомісткість рішення задачі пошуку найкоротших шляхів з урахуванням певних вимог до структури інформаційної мережі. Даний алгоритм доцільно застосовувати в порівняно великих корпоративних обчислювальних мережах при необхідності отримання оптимальних незалежних шляхів за час, який лінійно залежить від кількості вершин графа.

Література

1. Barabash O. Methods of self-diagnosis of telecommunication networks based on flexible structures of test connections / O. Barabash, N. Lukova-Chuiko, A. Musiyenko // Zborník príspevkov z medzinárodného vedeckého seminára „Riadenie bezpečnosti zložitých systémov". 23 – 27 februára 2015. – Liptovský Mikuláš: Akadémia ozbrojených síl generála Milana Rastislava Štefánika, 2015. – Str. 226 – 231.

2. Машков В.А. Организация самоконтроля модульных систем на основе оптимальных структур проверочных свіязей / В.А. Машков, О.В. Барабаш // Электронное моделирование. – К.: НАН Украины, 1995. – Том 17, № 3. – С. 68 – 75.

3. Барабаш О.В. Математична модель інтелектуалізації процесів управління рухомим об’єктом на основі нечітких семантичних мереж / О.В. Барабаш, Д.М. Обідін, А.П. Мусієнко // Наукові записки Українського науково-дослідного інституту зв’язку. – К.: УНДІЗ, 2014. – № 3(31). – С. 5 – 9.

4. Барабаш О.В. Алгоритм самодіагностування технічного стану вузлів комутації інформаційних систем / О.В. Барабаш, Д.М. Обідін, А.П. Мусієнко // Сучасний захист інформації. – К.: № 2 – 2014. – С. 114 – 121.

5. Барабаш О.В. Аналіз побудови мережі відеоконтролю пунктів митного спостереження на основі функціонально стійкої системи / О.В. Барабаш, С.В. Бодров, А.П. Мусієнко // Зв'язок. – К.: ДУТ, 2014. – № 2. – С. 8 – 11.

6. Барабаш О.В. Модель бази знань інтелектуальної системи управління високошвидкісного рухомого об’єкта на основі її верифікації / О.В. Барабаш, Д.М. Обідін, А.П. Мусієнко // Системи обробки інформації. – Харків: ХУПС, 2014. – № 5 (121). – С. 3 – 6.

7. Lukova-Chuiko N. Choice of Reasonable Variant of Signal and Code Constructions for Multirays Radio Channels / N. Lukova-Chuiko, S. Toliupa, O. Oksiuk // Second International Scientific-Practical Conference Problems of Infocommunications. Science and Technology 2015. – IEEE PIC S&T 2015. – PP. 269 – 271.

8. Лукова-Чуйко Н.В. Математична модель взаємовідносин загроз та комплексних систем захисту інформації / Н.В. Лукова-Чуйко // Вісник інженерної академії України. – К., 2015. – № 3. – С. 131 – 135.

9. Graham R.L. On the history of the minimum spanning tree problem / R.L. Graham, P. Hell // Annals of the History of Computing, 1985. – №7 (1). – P. 43 – 57.

Автори статті

Саланда Іванна Петрівна – аспірант, Східноєвропейський національний університет імені Лесі Українки, Луцьк, Україна. Тел. +38 097 555 56 27. E-mail: ivanna_priymas@mail.ru

Барабаш Олег Володимирович – доктор технічних наук, професор, завідувач кафедри вищої математики, Державний університет телекомунікацій, Київ, Україна. Тел. +38 097 911 08 54. E-mail: bar64@ukr.net

Мусієнко Андрій Петрович – кандидат фізико-математичних наук, доцент кафедри вищої математики, Державний університет телекомунікацій, Київ, Україна. Тел. +38 093 752 00 19. E-mail: mysienkoandrey@gmail.com

Authors of the article

Salanda Ivanna Petrivna – graduate student. Lesya Ukrainka Eastern European National University, Lutsk, Ukraine. Tel. +38 097 555 56 27. E-mail: ivanna_priymas@mail.ru

Barabash Oleh Volodymyrovych – sciences doctor (technic), professor, Head of Department of Mathematics, State University of Telecommunications, Kyiv, Ukraine. Tel.: +38 097 911 08 54. E-mail: bar64@ukr.net

Musiyenko Andriy Petrovych – candidate of sciences (physics-mathematics), Associate Professor of Department of Mathematics, State University of Telecommunications, Kyiv, Ukraine. Tel. +38 093 752 00 19. E-mail: mysienkoandrey@gmail.com

Дата надходження в редакцію: 19.03.2016 р. Рецензент: д.т.н., проф. Ю.В. Кравченко