

**Р. В. Білоус, С. Д. Погорілий**

Київський національний університет ім. Тараса Шевченка  
Проспект Академіка Глушкова, 2, корпус 5, 03022 Київ, Україна

## **Особливості прикладного застосування генетичного алгоритму пошуку оптимальних шляхів на графі**

*Проведено аналіз складності генетичного алгоритму пошуку оптимальних шляхів на графі. Досліджено вплив параметрів генетичного алгоритму на характер збіжності розв'язку.*

**Ключові слова:** маршрутизація, генетичний алгоритм, задача пошуку оптимальних шляхів на графі.

### **Вступ**

Розвиток комп'ютерних мереж вимагає постійного розвитку та покращення алгоритмів, що керують їхньою роботою. Класичні алгоритми маршрутизації, балансування навантаження та відмовостійкості значною мірою залежать від розміру мережі, яку вони обслуговують. Складність цих алгоритмів різко зростає зі збільшенням точок обміну трафіком. Одним із недоліків класичних алгоритмів також є обмеження параметрів мережі, які вони використовують. Збіжність сучасних протоколів маршрутизації значною мірою пов'язана з динамікою змін у системі, а їхня службова інформація використовує значну частину ресурсів мережі. Тому за останні роки було сформовано низку нових протоколів маршрутизації мереж із підвищеною продуктивністю. Покращення роботи алгоритмів пошуку оптимальних шляхів у мережі в цих роботах досягається за рахунок формування нових алгоритмів чи створення паралельних версій уже існуючих [1]. Одним із передових підходів є метод пошуку та оптимізації шляхів із використанням генетичного алгоритму. В роботі [2] запропоновано методуку використання генетичного підходу до розв'язання задачі пошуку на графі та описано особливості застосування генетичних операцій. Метою даної роботи є дослідження ефективності створених методик і їхнього порівняння з класичними алгоритмами, що використовуються у сучасних алгоритмах маршрутизації.

### **Схема генетичного алгоритму**

Розглянемо формалізовану задачу пошуку оптимального шляху на графі, в якій задано зважений орієнтований граф  $G = (V, E)$ , де  $V$  — множина вершин;  $E \in V \times V$  — множина ребер графа. В загальному випадку існує декілька вагових функцій  $\omega_1, \dots, \omega_k : E \rightarrow R$ , кожна з яких відповідає певному критерію оптимізації.

Довільний шлях  $p = v_s \rightarrow v_d$ , де  $v_s$  та  $v_d$  — початкова та кінцева вершини шляху відповідно, складається з послідовності ребер  $\langle v_s, v_i \rangle, \langle v_i, v_j \rangle, \dots, \langle v_k, v_d \rangle \in E$ , і може бути

представлений у вигляді послідовності вершин графа, що належать шляху  $p = \langle v_s, v_i, \dots, v_d \rangle$ . Вершини  $v_s, v_i, \dots, v_d \in V$ , причому кожна вершина входить до шляху лише один раз. Визначимо  $x_{i,j}$  як

$$x_{i,j} = \begin{cases} 1, & \text{якщо ребро } (i, j) \text{ входить до шляху,} \\ 0, & \text{в іншому випадку.} \end{cases} \quad (1)$$

Нехай загальна кількість критеріїв оптимізації задачі  $k$ . За кожним критерієм визначається функціонал (цільова функція), який відповідає якості шляху з точки зору алгоритму маршрутизації і визначається як

$$C_m(p) = F_m(\omega_m(i, j), x_{i,j}), \quad m = 1, \dots, k, \quad (i, j) \in E. \quad (2)$$

На початковому етапі роботи запропонованого генетичного алгоритму задається граф, представлений матрицею суміжностей та вагові функції  $\omega_1, \dots, \omega_k$ , а також індекси  $s$  та  $d$  — початкової та кінцевої вершин відповідно. Формування початкового покоління відбувається випадковим вибором  $M$  шляхів  $p = v_s \rightarrow v_d$  із множини  $P$  усіх можливих шляхів між вершинами  $s$  та  $d$ . Розмір популяції  $M$  залишається незмінним у ході роботи алгоритму.

До кожного покоління розв'язків (хромосом) застосовуються операції кросоверу та мутації. Імовірності застосування цих операцій до певної хромосоми в запропонованій моделі не залежать від пристосованості хромосоми і позначаються  $P_c$  та  $P_m$  відповідно. Позначимо порядок покоління генетичного алгоритму індексом  $n$ , а кількість поколінь —  $N$ . Таким чином, запропонована модель роботи алгоритму може бути представлена у вигляді наступної схеми:

- 1) ініціалізація алгоритму ( $G; \omega_1, \dots, \omega_k : E \rightarrow R; s; d$ );
- 2) формування початкового покоління ( $M$ );
- 3) поки ( $n < N$ ):
- 4) вибір  $M \cdot P_c$  хромосом, застосування операції кросоверу;
- 5) вибір  $M \cdot P_m$  хромосом, застосування операції мутації;
- 6) відбір ( $M$  хромосом);
- 7) завершення алгоритму, представлення розв'язку.

Розглянемо операцію мутації, що полягає у формуванні випадкового шляху між двома проміжними вершинами шляху  $p = v_s \rightarrow v_d$ , причому шуканий проміжний шлях не повинен включати вершини основного шляху  $p$ . Позначимо вершини, що відповідають точкам мутації  $m_1$  та  $m_2$ . Введемо додаткове позначення  $V'$  — множина вершин графа  $G$ , що належать проміжним шляхам  $v_s \rightarrow m_1$  та  $m_2 \rightarrow v_d$ . Позначимо також результат мутації шляху  $p$  як  $p_m$ . Тоді схема операції мутації може бути представлена наступним чином:

- 1) формування  $V'$  ( $p = v_s \rightarrow v_d; m_1; m_2$ );
- 2)  $P' = \bigcup p'; \forall p' = m_1 \rightarrow m_2$ ;
- 3) для всіх  $p' \in P'$ :

- 4) якщо  $\exists v \in p'$ , що  $v \in V'$  тоді:
- 5)  $P' = P' / \{p'\}$ ;
- 6) вибір випадкового шляху  $p' \in P'$ ;
- 7)  $p_m = v_s \rightarrow m_1 \cup p' \cup m_2 \rightarrow v_d$ .

### Асимптотична складність генетичного алгоритму

Користуючись асимптотичною оцінкою складності алгоритмів, порівнюємо генетичний алгоритм із класичними алгоритмами пошуку найкоротших шляхів на графі. Слід зауважити, що таке порівняння доцільне лише для функціоналів  $C_m(p)$ , що формуються із сукупності адитивних характеристик з'єднань (ваг ребер графа) або при  $k = 1$ . Для кожного покоління генетичного алгоритму розглянемо операцію формування наступного покоління. Операція кросоверу згідно [2] полягає в обміні проміжними шляхами, і її складність рівна  $O(1)$ . Операція мутації полягає у формуванні простого шляху між довільними вершинами у графі, і її складність в найгіршому випадку рівна  $O(E)$  для представлення графа у вигляді списків суміжності [3]. Операція відбору залежить від складності обчислення функції пристосованості особини  $F(p)$ , яка у найпростішому випадку рівна сумі ваг ребер, що належать шляху  $p$ . Для турнірного способу відбору складність залежить від розміру популяції (кількості операцій порівняння) і рівна  $O(M)$ . Для рангового відбору складність рівна складності операції сортування і рівна  $O(M \cdot \log M)$  [4]. Зважаючи, що  $0 \leq P_c \leq 1$  та  $0 \leq P_m \leq 1$ , загальна складність генетичного алгоритму для турнірного та рангового способів відбору дорівнює відповідно

$$O(N \cdot M \cdot (E + M)) \quad (3)$$

та  $O(N \cdot M \cdot (E + M \cdot \log M))$ . (4)

Порівнюємо асимптотичну оцінку складності алгоритму із класичними алгоритмами пошуку найкоротших шляхів, що використовуються в сучасних алгоритмах маршрутизації (табл. 1).

Таблиця 1. Асимптотична складність алгоритмів пошуку найкоротших шляхів на графі

Алгоритм	Асимптотична складність
Дейкстри	$O(E \cdot \log V)$
Белмана-Форда	$O(E \cdot V)$
Генетичний алгоритм (турнірний відбір)	$O(N \cdot M \cdot (E + M))$
Генетичний алгоритм (ранговий відбір)	$O(N \cdot M \cdot (E + M \cdot \log M))$

Слід зазначити, що завданням генетичного алгоритму є не лише пошук єдиного оптимального розв'язку, а формування певного набору оптимальних шляхів, що можуть бути використані в протоколах маршрутизації та системах балансування навантаження в мережі [5].

### Залежність збіжності від параметрів генетичного алгоритму

Для оцінки залежності збіжності генетичного алгоритму від параметрів обрано один параметр оптимізації та функціонал  $C_m(p) = \sum_{i,j} \omega_m(i, j), x_{i,j}$ . Таким чином, для

знаходження точного розв'язку задачі можна застосувати один із класичних алгоритмів (алгоритм Дейкстри). Позначимо мінімальне відхилення функціонала  $C(p)$  від точного розв'язку в межах популяції як  $r_{\min}$ . Типова залежність  $r_{\min}$  від порядку покоління  $n$  наведена на рис. 1. Усі залежності усереднені для 100 запусків програми тестування.

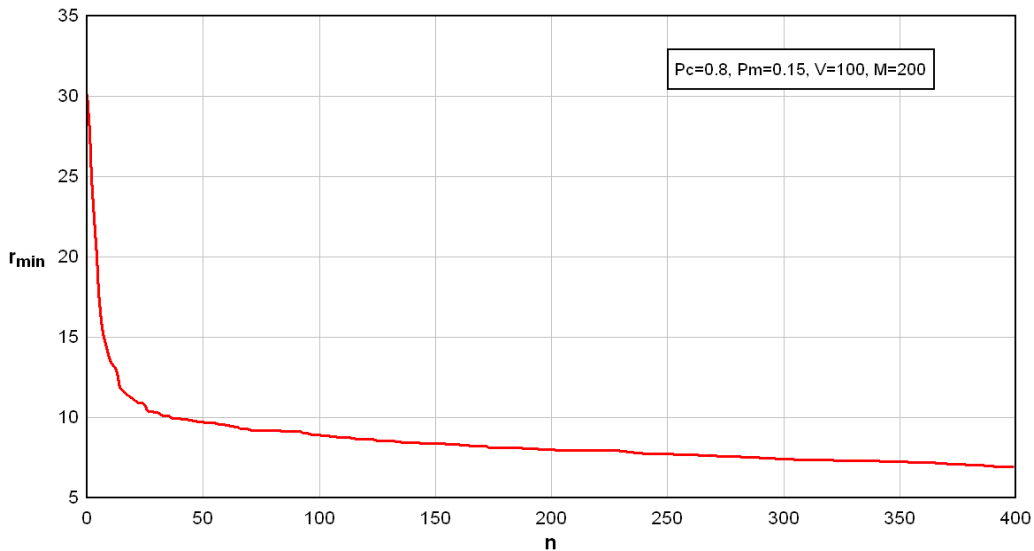


Рис. 1. Залежність  $r_{\min}$  від порядку покоління  $n$

Основними параметрами дослідження збіжності генетичного алгоритму є розмір популяції, імовірності кросоверу та мутації. Максимальний порядок покоління роботи алгоритму  $N$  обирається залежно від вимог практичного застосування (часу роботи алгоритму маршрутизації).

Зафіксуємо значення параметрів мутації та кросоверу, а також кількість вершин графа, що відповідає структурі досліджуваної мережі. Побудуємо залежність  $r_{\min}$  від  $n$  для різного розміру популяції  $M$  ( $M = 0,1V, 0,2V, 0,5V, V, 2V, 4V, 6V$ ) (рис. 2).

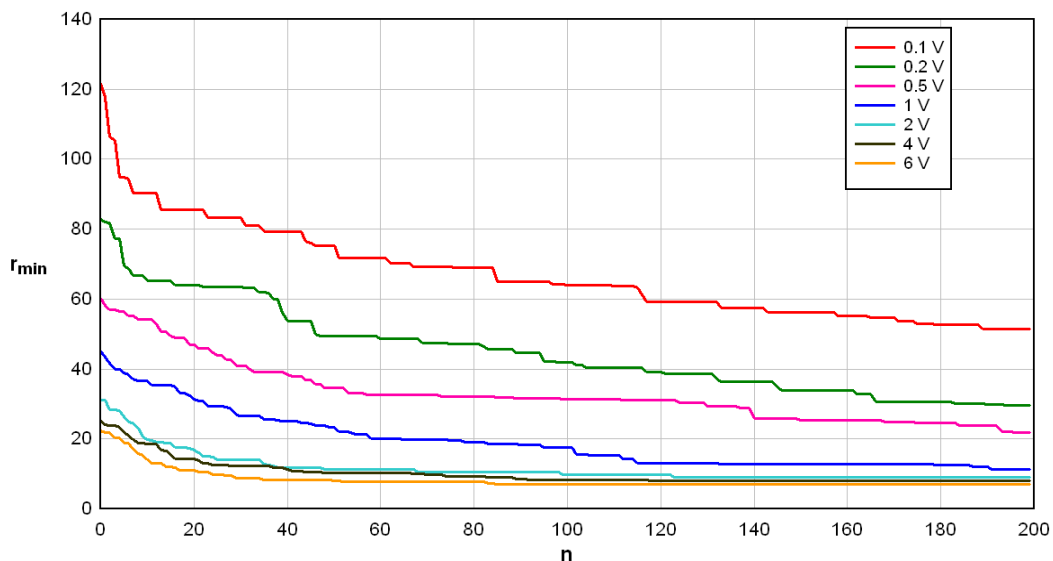


Рис. 2. Залежність збіжності від розміру популяції  $M$

З одержаних результатів випливає, що для розміру популяції  $M \approx 2V$  подальше збільшення  $M$  не приводить до покращення збіжності генетичного алгоритму. Розмір популяції значною мірою впливає на характеристики початкового покоління алгоритму — зі збільшенням розміру вибірки початкових випадкових шляхів відхилення значення цільової функції «найкращої» особи покоління від точного розв'язку зменшується. Важливим висновком з одержаних залежностей є також те, що незалежно від критерію зупинки роботи алгоритму ефективно значення максимального порядку покоління генетичного алгоритму не залежить від розміру популяції  $M$  при  $M \approx 2V$ .

Важливим питанням є формування критерію завершення алгоритму та залежність максимального порядку покоління роботи алгоритму від розмірності вхідного графа. Для одного параметра оптимізації максимальним порядком покоління вважатимемо покоління, при якому досягається точний розв'язок задачі. Залежність збіжності генетичного алгоритму від розміру вхідного графа зображено на рис. 3.

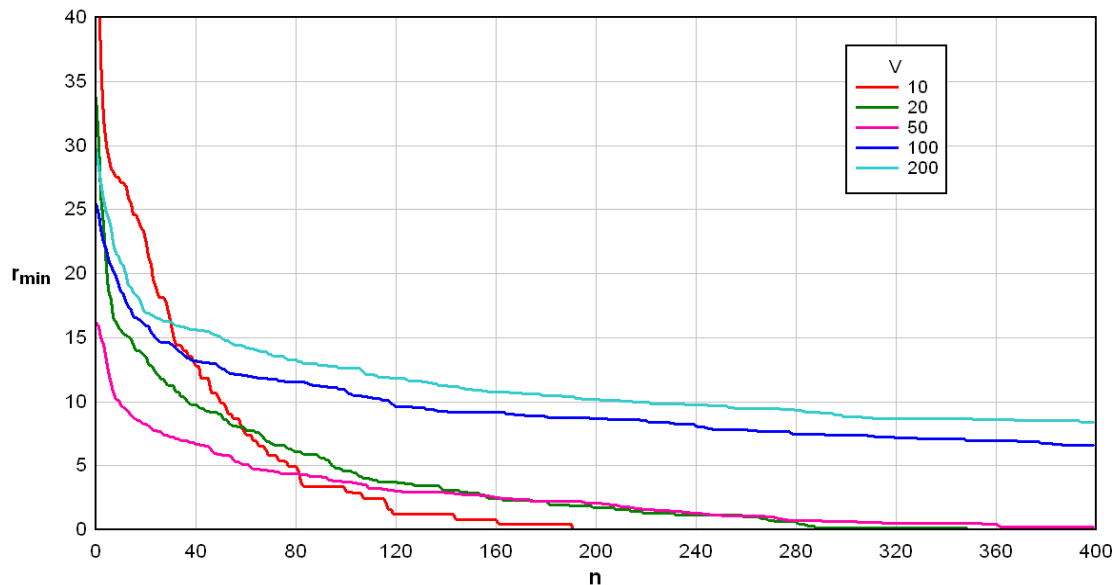


Рис. 3. Залежність збіжності від кількості вершин вхідного графа

Одержані результати показують, що збіжність алгоритму значною мірою залежить від розміру вхідних даних (табл. 2). Позначимо порядок покоління, при якому алгоритм збігається до точного розв'язку як  $N_{eff}$ .

Таблиця 2. Залежність  $N_{eff}$  від  $V$

$V$	$N_{eff}$
10	190
20	350
50	755
100	4152
200	9675

Розглянемо залежність характеру збіжності генетичного алгоритму від імовірності мутації хромосом  $P_m$  (рис. 4).

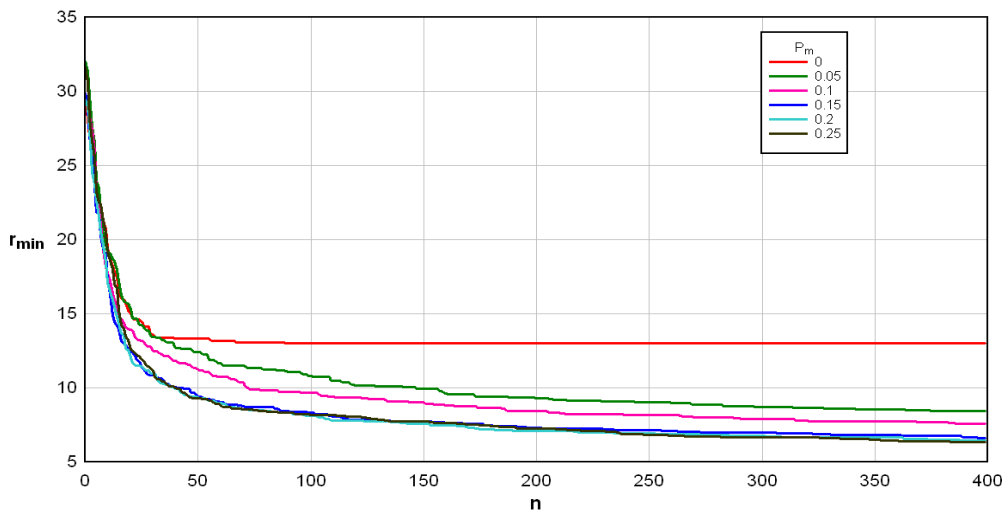


Рис. 4. Залежність збіжності від імовірності мутації  $P_m$

Залежності показують, що за відсутності мутації ( $P_m = 0$ ) алгоритм досить швидко збігається до локального екстремуму. Експериментальні дані свідчать, що оптимальним є значення параметра  $P_m = 0,15$ . При подальшому збільшенні цього параметра характер збіжності алгоритму не змінюється. Слід також зауважити, що складність алгоритму значною мірою визначається складністю операції мутації, що впливає на часові характеристики роботи генетичного алгоритму для вхідних графів малої розмірності.

Розглянемо залежність характеру збіжності генетичного алгоритму від імовірності кросоверу хромосом  $P_c$  (рис. 5).

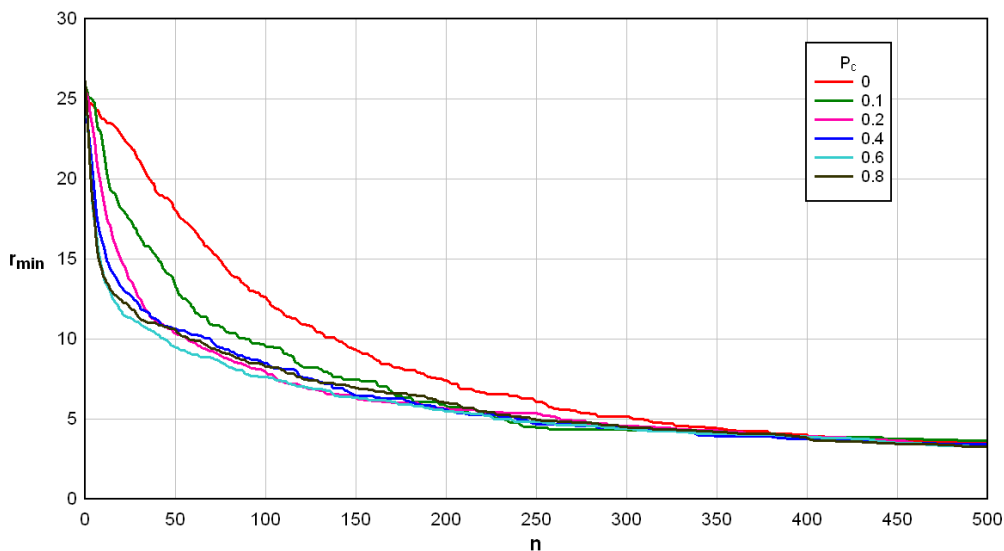


Рис. 5. Залежність збіжності від імовірності кросоверу  $P_c$

Проаналізувавши одержані залежності, було отримано оптимальне значення параметра  $P_c = 0,6$ . Слід зауважити, що реальний відсоток хромосом, до яких успішно застосовано операцію мутації дещо менший величини  $P_c$ , що обумовлено особливостями запропонованої моделі [2].

Незважаючи на залежність характеру збіжності алгоритму від розмірів вхідного графа (рис. 3), одним із основних критеріїв зупинки алгоритму є обмеження на порядок покоління, яке не залежить від розмірності оброблюваного графа. Такий вибір пов'язаний зі значним зростанням порядку покоління, при якому алгоритм збігається до точного розв'язку при збільшенні розміру вхідного графа. Необхідно також зазначити, що для задач з багатьма критеріями оптимізації точного розв'язку може взагалі не існувати, і основним завданням є формування низки ефективних шляхів між двома вершинами за обмежену кількість ітерацій генетичного алгоритму. Для такого підходу

$$N \neq f(V). \quad (5)$$

Підставивши одержане раніше ефективне значення розміру популяції  $M = 2V$  в (3) і (4), та зважаючи на умову (5), одержимо значення складності генетичного алгоритму:

$$O(V \cdot E), \quad (6)$$

$$O(V \cdot (E + V \cdot \log V)). \quad (7)$$

Таким чином, складність генетичного алгоритму для турнірного способу відбору співпадає зі складністю алгоритму Белмана-Форда.

## Висновки

У роботі проведено теоретичні оцінки складності генетичного алгоритму та експериментально перевірено залежність його збіжності від вхідних параметрів задачі. Запропоновано чисельні оцінки ефективності генетичного підходу порівняно з класичними алгоритмами маршрутизації в мережах.

У результаті проведених експериментальних досліджень запропоновано найбільш ефективні значення параметрів генетичного алгоритму: розмір популяції, імовірності кросоверу та мутації залежно від розміру вхідних даних задачі.

Одержані результати можуть бути використані для проектування та впровадження протоколів маршрутизації, що базуються на генетичних алгоритмах пошуку оптимальних шляхів на графі.

1. *Погорілий С.Д.* Формування та аналіз паралельних схем алгоритму Дейкстри / С.Д. Погорілий, Ю.В. Бойко, Р.В. Білоус // Математичні машини і системи. — 2008. — Т. 4. — С. 61–71.
2. *Погорілий С.Д.* Генетичний алгоритм розв'язання задачі маршрутизації в мережах / С.Д. Погорілий, Р.В. Білоус // Проблеми програмування. — 2010. — № 2–3: Спец. вип. — С. 171–178.
3. *Седжвик Р.* Фундаментальные алгоритмы на C++. Алгоритмы на графах / Р. Седжвик; пер. с англ. — СПб: ООО «ДиаСофтЮП», 2002. — 496 с.
4. *Ахо А.* Структуры данных и алгоритмы / А. Ахо, В. Хопкрофт, Дж. Д. Ульман. — М.: Диалектика, 2003. — 384 с.
5. *Gen M.* Genetic Algorithms and Engineering Optimization / M. Gen, R. Cheng. — New York: Wiley, 2000.
6. *Goldber D.E.* Genetic Algorithms in Search, Optimization, and Machine Learning / D.E. Goldber— MA: Addison-Wesley, 2000.
7. *Sateesh Kumar P.* Genetic Zone Routing Protocol / P. Sateesh Kumar, S. Ramachandram // Journal of Theoretical and Applied Information Technology. — 2008.

Надійшла до редакції 17.06.2010