

ЗАДАЧА РОЗФАРБУВАННЯ ГРАФУ ДЛЯ КЕРУВАННЯ РУХОМ НА ПЕРЕХРЕСТІ

Вінницький національний технічний університет

Анотація

Керування рухом на перехресті подано як задачу розфарбування графу, для вирішено якої запропоновано використати генетичний алгоритм. У роботі наведено загальний алгоритм роботи генетичного алгоритму для розв'язання поставленої задачі та приклади реалізації методів кросоверу та мутації.

Ключові слова: еволюційний алгоритм, розфарбування графу, генетичний алгоритм, контроль руху

Abstract

Traffic control at the intersection is presented as a problem of coloring the graph, for which it is decided to use a genetic algorithm. The paper presents a general algorithm for the operation of the genetic algorithm for solving the problem and examples of the implementation of the crossover and mutation methods.

Keywords: evolution algorithm, graph coloring, genetic algorithm, traffic control

Вступ

Зростання кількості транспортних засобів призвело до неминучого збільшення заторів, тривалості пересування та збільшенні небезпеки для пішоходів через неефективне управління світлофорами. В зв'язку з цим з'явилась необхідність використання нових методів при проектуванні моделі керування світлофорами для транспортних засобів та пішохідних переходів [1, 2].

Вирішення даної задачі полягає у пошуку оптимального рішення для ефективного планування руху транспорту та пішоходів на перехресті. Слід звернути увагу, що рух транспорту і пішоходів можливий у будь-якому порядку, але вони не можуть рухатись одночасно, так як це призведе до хаосу та спричинить небезпечну ситуацію на дорозі. Враховуючи дані фактори, задачу керування рухом на перехресті представимо у вигляді задачі розфарбування графу.

Запропонований підхід

Задача розфарбовування графу (Graph Coloring Problem - GCP) - це добре відома NP-повна задача. Суміжні вершини повинні бути забарвлені у різні кольори, при чому, їх кількість обмежена. Другочергова мета полягає в тому, щоб знайти мінімальну кількість різних кольорів, необхідних для розфарбування певного графу, не порушуючи умову суміжності [3]. У нашому випадку вершинами графу є рух транспорту чи пішоходів, а ребрами – конфлікт у їх виконанні. Мінімальну кількість кольорів визначимо як мінімальний час перетину перехрестя усіма учасниками руху.

В зв'язку з тим, що дана задача є NP-повною, застосуємо генетичний алгоритм для її розв'язання. Вони мають загальну структуру та процес виконання, але відрізняються по певним деталям в залежності від конкретної задачі. Загальний алгоритм роботи генетичного алгоритму наведено на рисунку 1.

На початку роботи алгоритму створимо популяцію і тоді у циклі декодуємо наші дані у двійковий код та обрахуємо функції пристосованості кожної хромосоми. Після цього виконаємо селекцію, кросовер та мутацію створеної популяції. Повторюємо дані дії поки алгоритм не закінчить свою роботу [4].

Розроблений тестовий приклад методу кросоверу та мутації для поставленої задачі наведено нижче:

```

def crossover(color_parent1, color_parent2):
    crosspoint = random.randint(0, len(color_parent1))
    child = color_parent1[0:crosspoint] + \
            color_parent2[crosspoint:len(color_parent2) - 1]
    return child

def mutation(chromosome, adjacency_matrix, available_colors_list):
    for index, vertex_color in enumerate(chromosome):
        if adjacency_matrix[index][index+1] == adjacency_matrix[index+1][index] \
            and vertex_color == chromosome[index+1]:
            available_colors_list.pop(vertex_color)
            new_color = available_colors_list.get(
                random.randint(0, len(available_colors_list) - 1)
            )
            chromosome[index] = new_color
    return chromosome

```

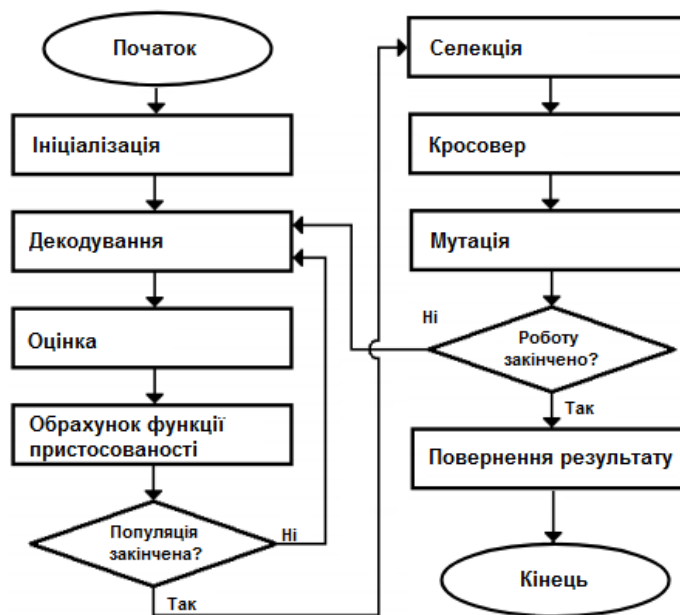


Рисунок 1 – Загальний алгоритм роботи генетичного алгоритму

Представимо хромосому як простий масив з довжиною, що дорівнює кількості вершин в графі [5]. Кожній комірці в масиві присвоїмо значення від 0 до кількості кольорів – k (рис. 2). Суміжності між вершинами представимо матрицею розміром $n \times n$, де n – кількість вершин (табл. 1).

Таблиця 1 – Матриця суміжностей між вершинами

	x1	x2	x3	x4	x5	x6	x7
x1		1	1	1	1		1
x2	1		1	1	1	1	1
x3	1	1			1	1	
x4	1	1			1	1	
x5	1	1	1	1		1	1
x6		1	1	1	1		1
x7	1	1			1	1	

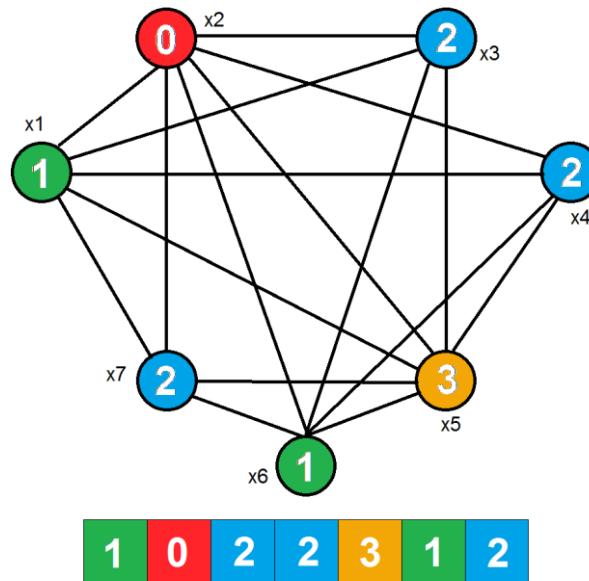


Рисунок 2 – Хромосомне представлення розфарбованого графу

Кінцевою ціллю, при якій задача розфарбування графу вважається вирішеною, є досягнення ситуації, в якій не існує двох суміжних вершин, які були б пофарбовані в однакові кольори. Тому генетичний алгоритм будемо виконувати до тих пір, поки він не знайде рішення (тобто будуть відсутні конфлікти) або допоки алгоритм не виконає попередньо визначену кількість ітерацій.

Висновки

Керування рухом на перехресті було представлено як задачу розфарбування графу, для вирішення якої використано генетичний алгоритм. Для цього розроблено тестову програму, яка генерує покоління та виконуючи такі операції генетичного алгоритму як селекцію, кросовер, декодування, мутацію та інші, здійснює розфарбування. В ході тестування було виявлено, що коли задано фіксовий час перемикання світлофору, це може спричинити до пустування одних маршрутів і незначної переповненості інших. В зв'язку з цим, метою подальшого дослідження є розробка функціоналу для керування рухом у динамічному середовищі, наприклад на основі розміру потоку автомобілів або пішоходів, з метою адаптації та оптимізації перемикання світлофору у різні періоди часу.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. T. Royani Control of Traffic Light in Isolated Intersections Using Fuzzy Neural Network and Genetic Algorithm // International Journal of Computer and Electrical Engineering. – 2013. – Volume 5, No. 1, P. 142-146 [Електронний ресурс]. – Режим доступу: <http://www.ijcee.org/papers/682-X101042.pdf>
2. Ayad M. Turk The Use of Genetic Algorithm for Traffic Light and Pedestrian Crossing Control // International Journal of Computer Science and Network Securit. – 2009. – Volume 9, No. 2, P. 88-96 [Електронний ресурс]. – Режим доступу: http://paper.ijcsns.org/07_book/200902/20090212.pdf
3. Sidi Mohamed Douiri Solving the graph coloring problem via hybrid genetic algorithms // Journal of King Saud University - Engineering Sciences. – 2015. - Volume 27, Issue 1. P. 114-118 [Електронний ресурс]. – Режим доступу: <https://www.sciencedirect.com/science/article/pii/S1018363913000135#b0055>
4. Генетичний алгоритм. Матеріал з Вікіпедії [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Генетичний_алгоритм

5. Musa M. Hindi Genetic Algorithm Applied to the Graph Coloring Problem // Musa M. Hindi, Roman V. Yampolskiy – Louisville: Computer Engineering and Computer Science J.B. Speed School of Engineering, 2012. – 7p. [Електронний ресурс]. – Режим доступу: http://ceur-ws.org/Vol-841/submission_10.pdf

Тасьмук Дмитро Ігорович — студент групи 2КН-17м, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця, e-mail: dimatasmuk@gmail.com

Володимир Іванович Месюра — канд. техн. наук, доцент, професор кафедри комп'ютерних наук, Вінницький національний технічний університет, м. Вінниця.

Dmytro I. Tasmuk — student of Information Technologies and Computer Engineering Department, 2CS-17m, Vinnytsia National Technical University, Vinnytsia, e-mail: dimatasmuk@gmail.com

Volodymyr I. Mesyura — Cand. Sc., Assistant Professor, Professor of the Computer Science Chair, Vinnytsia National Technical University, Vinnytsia.