

Соболь Егор Игоревич

магістрант кафедри ПМІ, ДонНТУ

Соболь Егор Игоревич

магістрант кафедри ПМІ, ДонНТУ

Sobol Y.

undergraduate of AMI, DonNTU

Дмитрієва Ольга Анатолівна

зав. кафедри прикладної математики,

д.т.н., ДонНТУ

Дмитриева Ольга Анатольевна

зав. кафедры прикладной математики,

д.т.н., ДонНТУ

Dmitrieva O.

Head Department of Applied Mathematics,

Ph.D., DonNTU

ПОРІВНЯЛЬНИЙ АНАЛІЗ АЛГОРИТМІВ ФОРМУВАННЯ ВИПАДКОВИХ КАРТ СРАВНИТЕЛЬНЫЙ АНАЛИЗ АЛГОРИТМОВ ФОРМИРОВАНИЯ СЛУЧАЙНЫХ КАРТ COMPARATIVE ANALYSIS OF THE ALGORITHMS FOR GENERATING RANDOM MAPS

Анотація. Проведено порівняльний аналіз вибраних біонічних і процедурних методів генерації випадкових карт.

Ключові слова: випадкова карта, лабіринт, біонічний метод.

Аннотация. Проведен сравнительный анализ выбранных бионических и процедурных методов генерации случайных карт.

Ключевые слова: случайная карта, лабиринт, бионический метод.

Summary. A comparative analysis of selected bionic and procedural methods for generating random maps is carried out.

Keywords: random map, labyrinth, bionic method.

В роботі розглянуто сучасні алгоритми та проведено порівняльний аналіз методів, що використовуються для формування випадкових карт. Задачі формування випадкових карт виникають в широкому спектрі завдань, пов'язаних, перш за все, зі створенням тренажерів, проектуванням симуляторів та розробкою комп'ютерних ігор. Випадкова карта в роботі представляється у вигляді двовимірної матриці, де одиниці позначають «стіни», а нулі позначають «підлогу».

Метою даної роботи є проведення порівняльного аналізу роботи базових алгоритмів, модифікованого алгоритму диференціальної еволюції та запропонованого процедурного алгоритму генерації лабіринту. В якості параметра для порівняння використовується

складність результуючого лабіринту[1], яка визначається шляхом вимірювання довжини найкоротшого можливого шляху. Чим більше ця величина, тим складніше вважається лабіринт. Для порівняння в роботі обрано базовий рандомізований алгоритм Прима, який є варіантом алгоритму Прима[2] для пошуку мінімального основного дерева. Етапи алгоритму можна описати наступним чином:

- 1) Розпочати з сіткою, заповненою стінами.
- 2) Обрати клітину, помітити її як частину лабіринту. Додати стіни клітини до списку стін.
- 3) Доки список стін не порожній:
 - вибрати випадкову стіну зі списку. Якщо тільки одна з клітин, які стіна розділяє, переглянута, тоді:

зробити стіну проходом і позначити не відвідану клітку як частину лабіринту; додати стіни помічної клітини до списку стін;

- видалити стіну зі списку

У якості альтернативи для процедурного підходу запропоновано модифікацію методу диференціальної еволюції[3]. У його базовому вигляді алгоритм можна описати таким чином. Спочатку генерується деяка множина векторів, так зване покоління. На кожній ітерації алгоритм генерує нове покоління векторів, випадковим чином комбінуючи вектори з попереднього покоління. Число векторів в кожному поколінні одне й те саме і є одним з параметрів методу. Нове покоління векторів генерується в такий спосіб. Для кожного вектора x_i з старого покоління вибираються три різних випадкових вектори v_1, v_2, v_3 серед векторів старого покоління, за винятком самого вектора x_i , і генерується так званий *мутантний вектор* за формулою (1)

$$v = v_1 + F * (v_2 - v_3). \quad (1)$$

де F — один з параметрів методу, позитивна дійсна константа в інтервалі $[0, 2]$.

Над мутантним вектором v виконується операція «схрещування», яка полягає в тому, що деякі його координати заміщаються відповідними координатами з початкового вектора x_i (кожна координата заміщається з деякою ймовірністю, яка також є ще одним з параметрів цього методу). Отриманий після схрещування вектор називається *пробним вектором*. Якщо він виявляється кращим за векр x_i (тобто значення цільової функції стало меншим), то в новому поколінні вектор x_i замінюється на пробний вектор, а в іншому разі — залишається x_i .

Цей метод використовує ідеї генетичних алгоритмів[4] і є біонічним методом. Базовий метод призначений для роботи з набором векторів і мета модифікації полягає в тому, щоб зробити можливою роботу з набором матриць за допомогою даного методу.

Формула отримання нового нащадка (2) виглядає наступним чином

$$m = C2(m1, C1(m2, m3, m4)), \quad (2)$$

де m — результуюча матриця;

$m1, m2, m3, m4$ — матриці-предки;

$C2$ — функція вторинного схрещування;

$C1$ — функція первинного схрещування.

Функція первинного схрещування повертає матрицю $m2$, яка заповнюється наступним чином:

$$m2[i, j] = 1, \text{ if } m3[i, j] == 1 \ \&\& \ m4[i, j] == 1 \ \&\& \ p < F, \quad (3)$$

де F є одним з параметрів методу, що відповідає за ймовірність первинного схрещування, приймає значення від 0,0 до 1,0.

Функція схрещування представлена наступним співвідношенням:

$$m2[i, j] = m1[i, j], \text{ if } p < Cr, \quad (4)$$

де Cr відповідає за вірогідність вторинної передачі генів, і приймаючим значення від 0,1 до 0,9.

Також важливими параметрами методу є:

W — щільність заповнення нульового покоління карт стінами;

N — кількість поколінь методу диференціальної еволюції.

Процес відбору партнерів для схрещування відбувається випадковим чином для кожної матриці поточного покоління. В рамках одного схрещування матриці не повторюються.

Порівняння результатів роботи алгоритмів

На графіку (рис. 1) представлені результати роботи алгоритму Прима та метода диференціальної еволюції на картах різного розміру. На графіку позначені середні значення із серії з десяти експериментів для кожного розміру карти. Для метода диференціальної еволюції було використано два набори параметрів:

D1, де $W = 0.4, C1 = 0.5, C2 = 0.5, N = 500$

D2, де $W = 0.4, C1 = 0.7, C2 = 0.55, N = 500$.

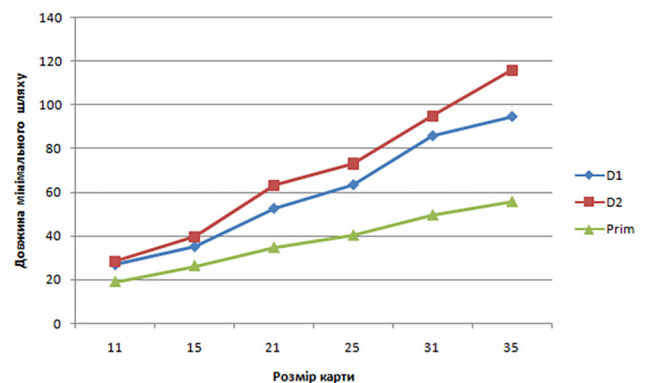


Рисунок 1. Результати роботи алгоритмів

Висновки

Довжина мінімального шляху в картах, згенерованих за допомогою біонічного методу, перевищує довжину мінімального шляху в картах, згенерованих ран-

домізованим алгоритмом Прима. Як можна побачити на графіку, зі збільшенням розміру карти складність лабіринтів, згенерованих методом диференціальної еволюції, збільшується швидше, ніж складність лабі-

ринтів, згенерованих за допомогою алгоритму Прима. Підбір параметрів методу дозволяє ще більше підвищити складність випадкових лабіринтів, що генеруються.

Література

1. Мозговой М. Занимательное программирование: Самоучитель / Мозговой М. // СПб: Питер, 2004. — 208 с. — ISBN 5-94723-853-5.
2. Pettie, S. An optimal minimum spanning tree algorithm / S. Pettie, V. Ramachandran // Journal of the ACM. — 2000. — Vol 1. — P. 16–34.
3. Storn R. Differential Evolution A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces / R. Storn, K. Price // Journal of Global Optimization. — Vol. 11, № 4. — 1997. — P. 341–359.