

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное автономное образовательное учреждение  
высшего образования «Санкт-Петербургский политехнический  
университет Петра Великого»

Институт компьютерных наук и кибербезопасности

Высшая школа технологий искусственного интеллекта

Направление: 02.03.01 «Математика и компьютерные науки»

Реферат по дисциплине  
«Генетические алгоритмы»  
«Планирование пути робота»

Студент,  
группы 5130201/20102

\_\_\_\_\_ Тищенко А. А.

Преподаватель

\_\_\_\_\_ Большаков А. А.

« \_\_\_\_\_ » \_\_\_\_\_ 2025г.

Санкт-Петербург, 2025

# Содержание

|                                                     |           |
|-----------------------------------------------------|-----------|
| <b>Введение</b>                                     | <b>3</b>  |
| <b>1 Постановка задачи</b>                          | <b>4</b>  |
| <b>2 Классификация алгоритмов планирования пути</b> | <b>5</b>  |
| <b>3 Генетические алгоритмы для построения пути</b> | <b>7</b>  |
| 3.1 Grid-кодирование пути . . . . .                 | 7         |
| 3.2 Угловое кодирование пути . . . . .              | 12        |
| <b>Заключение</b>                                   | <b>16</b> |
| <b>Список литературы</b>                            | <b>17</b> |

# Введение

Планирование пути мобильного робота — это задача построения коллизие-свободной траектории из начальной точки в целевую в присутствии препятствий и ограничений среды. Актуальность задачи обусловлена широким спектром приложений: от автономной логистики и сельского хозяйства до роботизированной инспекции и сервисной робототехники. Традиционные графовые методы (Дейкстра,  $A^*$ ) обеспечивают точный поиск на известных картах, однако их эффективность снижается при росте размерности, наличии динамики и сложных ограничений. Эволюционные подходы, в частности генетические алгоритмы (ГА), предоставляют гибкую эвристическую альтернативу, позволяя естественно учитывать несколько критериев качества и штрафы за нарушения ограничений.

Цель работы — исследовать применение ГА для планирования пути на дискретном представлении среды (grid) и в параметризации траекторий через направления движения (угловое кодирование), а также сравнить различные способы кодирования хромосом и их влияние на качество маршрутов и вычислительные затраты.

В работе:

- кратко рассмотрена классификация алгоритмов планирования пути по характеру среды (статическая/динамическая), принципу (глобальные/локальные) и полноте (точные/эвристические);
- формализована фитнес-функция с учётом длины пути и штрафов за пересечение препятствий, а для углового кодирования — с дополнительными слагаемыми за число поворотов;
- изучены три варианта grid-кодирования узлов пути (двоичное, десятичное, упорядоченное) и угловое кодирование с 4 и 8 направлениями движения;
- проведены эксперименты в статических сценах и сценарии с поэтапным появлением новых препятствий (offline/online-перепланирование);
- сопоставлены метрики: длина маршрута, число поколений и время решения.

Полученные результаты демонстрируют применимость ГА к задачам планирования как в статических, так и в динамически меняющихся средах, а также выявляют влияние выбора представления пути на качество и стоимость поиска.

# 1 Постановка задачи

Планирование пути робота – вычисление пути, свободного от столкновений, от начальной позиции до конечной среди множества препятствий



Рис. 1. Задача планирования пути

## 2 Классификация алгоритмов планирования пути

На схеме 2 показана классификация алгоритмов планирования пути.

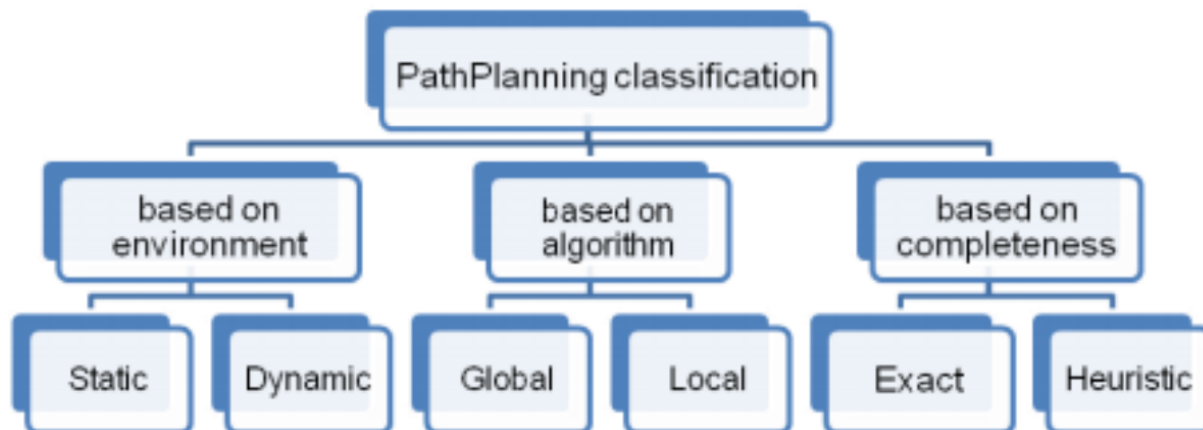


Рис. 2. Классификация алгоритмов планирования пути

Алгоритмы планирования пути можно классифицировать по трём главным признакам:

- По характеру среды (based on environment)
  - **Статические (Static)** Используются, когда карта среды известна заранее и не меняется. Все препятствия заранее заданы, и алгоритм строит маршрут до старта движения. Пример: классические алгоритмы поиска пути на графе ( $A^*$ , Дейкстра).
  - **Динамические (Dynamic)** Применяются в условиях, где препятствия могут появляться или перемещаться во время движения робота. Алгоритм должен уметь перестраивать путь «на лету» (онлайн-планирование). Пример:  $D^*$  Lite, Anytime Repairing  $A^*$ , алгоритмы на основе предсказаний и реактивного управления.
- По принципу алгоритма (based on algorithm)
  - **Глобальные (Global)** Предполагают знание полной карты окружающей среды. Алгоритм ищет оптимальный путь целиком от старта до цели. Пример:  $A^*$ , Дейкстра, волновой алгоритм (Wavefront).
  - **Локальные (Local)** Робот строит путь только на основе данных с датчиков «здесь и сейчас». Такие методы хорошо работают в неизвестных или частично известных средах. Пример: Bug Algorithms, Potential Fields, Dynamic Window Approach.
- По полноте поиска (based on completeness)

- **Точные (Exact)** Гарантируют нахождение решения (если оно существует), а также могут обеспечить оптимальность. Но часто требуют много вычислительных ресурсов. Пример: алгоритм Дейкстры, точный  $A^*$ .
- **Эвристические (Heuristic)** Используют приближённые методы и эвристики для ускорения поиска. Решение находится быстрее, но не всегда оптимальное. Пример: эвристический  $A^*$ , эволюционные методы (генетические алгоритмы, рой частиц, муравьиные алгоритмы).

## 3 Генетические алгоритмы для построения пути

Для решения произвольной задачи с помощью ГА необходимо определить:

- особь и популяцию;
- генетические операторы;
- фитнес функцию;
- параметры ГА.

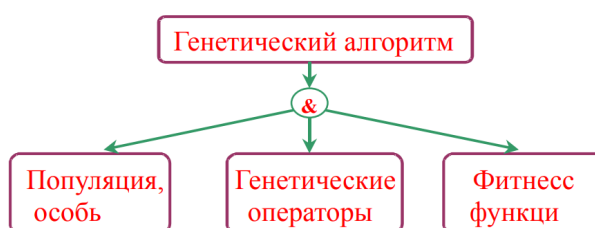


Рис. 3. Составляющие ГА

### 3.1 Grid-кодирование пути

#### Кодирование хромосом

Рассмотрим часто используемое на практике представление окружения робота в виде «решетки» – сетки (grid). Для представления окружения используется:

- двоичное;
- десятичное;
- упорядоченное кодирование;

показанное на рисунке 4 (белые клетки – свободное пространство, серые – препятствия).

Тогда хромосома (особь популяции) представляет потенциальное решение - Путь от начальной точки S (левый верхний угол) до конечной точки- цели Т (правый нижний угол) Хромосома (путь робота) содержит начальную и конечную вершины графа, представляющего решетку, а также вершины, пересекаемые роботом Эти вершины (или шаги) называются генами хромосомы. Различные методы кодирования используются для хромосомы, в зависимости от формы представления окружения:

|          | 0000 (0) | 0001 (1) | 0010 (2) | 0011 (3) | 0100 (4) | 0101 (5) | 0110 (6) | 0111 (7) | 1000 (8) | 1001 (9) |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| 0000 (0) | S<br>0   | 1        | 2        | 3        | 4        | 5        | 6        | 7        | 8        | 9        |
| 0001 (1) | 10       | 11       | 12       | 13       | 14       | 15       | 16       | 17       | 18       | 19       |
| 0010 (2) | 20       | 21       | 22       | 23       | 24       | 25       | 26       | 27       | 28       | 29       |
| 0011 (3) | 30       | 31       | 32       | 33       | 34       | 35       | 36       | 37       | 38       | 39       |
| 0100 (4) | 40       | 41       | 42       | 43       | 44       | 45       | 46       | 47       | 48       | 49       |
| 0101 (5) | 50       | 51       | 52       | 53       | 54       | 55       | 56       | 57       | 58       | 59       |
| 0110 (6) | 60       | 61       | 62       | 63       | 64       | 65       | 66       | 67       | 68       | 69       |
| 0111 (7) | 70       | 71       | 72       | 73       | 74       | 75       | 76       | 77       | 78       | 79       |
| 1000 (8) | 80       | 81       | 82       | 83       | 84       | 85       | 86       | 87       | 88       | 89       |
| 1001 (9) | 90       | 91       | 92       | 93       | 94       | 95       | 96       | 97       | 98       | T<br>99  |

Рис. 4. Grid-кодирование пути

1. Двоичное кодирование – содержит двоичные коды номеров строк и столбцов (координаты  $x,y$ ) «узловых» вершин пути.



Рис. 5. Двоичное кодирование

2. Десятичное кодирование – содержит десятичные номера строк и столбцов (координаты  $x,y$ ) «узловых» вершин пути.



Рис. 6. Десятичное кодирование

3. Упорядоченное кодирование - содержит номера(в общем порядке ячеек - вершин) проходимых «узловых» вершин пути.

## Инициализация популяции

Выполняется случайно При этом некоторые построенные хромосомы соответствуют нереализуемым путям, которые пересекают препятствия В таблице представлены экспериментальные данные по генерации начальной популяции

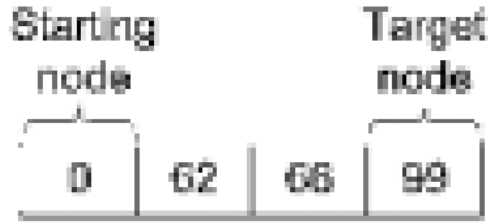


Рис. 7. Упорядоченное кодирование

| Initial population method | Random | Feasible |
|---------------------------|--------|----------|
| Fitness value             | 28.34  | 28.29    |
| Number of generation      | 81     | 36       |
| Solution time(s)          | 0.87   | 0.49     |

Рис. 8. Таблица экспериментальных данных

Далее находятся оптимальные или близкие к оптимальным решения (пути роботов), даже если начальная популяция содержит нереализуемые решения

## Фитнесс функция

Целью планирования пути является построение пути робота между начальной и конечной точками. Оптимальный путь должен иметь минимальную длину, время и минимальные энергозатраты. Прежде всего, минимизируется длина пути. Поэтому все эволюционные методы построения пути включают в фитнесс-функцию длину пути робота, которая может быть определена следующим образом.

Здесь  $P_i$  -  $i$ -й ген хромосомы  $P$ ,  $d$  - расстояние между 2-мя узлами. Фитнесс-функция для реализуемых путей определяется как сумма расстояний между узлами пути. Заметим, что для нереализуемых путей (имеющих пересечения с препятствия) фитнесс-функция имеет дополнительную штрафную функцию. Если между узлами встречается препятствие, то в фитнессфункцию добавляется штраф. Для поиска пути робота используется классический ГА. На последнем шаге итерации (поколения) в соответствии с вычисленными значениями фитнесс-функции отбираются хромосомы для производства потомков с помощью генетических операторов кроссинговера и мутации. Для выбора родителей применяется ранговый отбор

$$f = \begin{cases} \sum_{i=1}^{n-1} d(p_i, p_{i+1}), & \text{для допустимых путей (без столкновений),} \\ \sum_{i=1}^{n-1} d(p_i, p_{i+1}) + \text{штраф}, & \text{для недопустимых путей.} \end{cases} \quad (1)$$

$$d(p_i, p_{i+1}) = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}, \quad (2)$$

где  $p_i = (x_i, y_i)$  — координаты  $i$ -й вершины пути.

## Генетические операторы

Для генерации потомков используется обычный 1-точечный кроссинговер с заданной вероятностью  $P_c$ .



Рис. 9. Одноточечный кроссинговер

Пример на Рис. 9 показывает кроссинговер для упорядоченного кодирования хромосом.

Далее к потомкам (каждому гену) с малой вероятностью  $P_m$  применяется оператор мутации, который вносит небольшое изменение в ген в зависимости от способа кодирования. Мутация расширяет пространство поиска и препятствует преждевременной сходимости в локальном экстремуме

## Эксперименты

**Эксперимент 1.** Построение пути проводилось в следующем окружении (сетка 16x16) с 8 препятствиями с использованием указанных выше методов кодирования хромосом.

Параметры ГА:

- Мощность популяции  $N = 60$ ;
- Вероятности: кроссинговера  $P_c = 1.0$ , мутации  $P_m = 0.1$

После 10 запусков ГА для каждого метода получены средние значения расстояния, число поколений и время решения, которые представлены на Рис. 11.

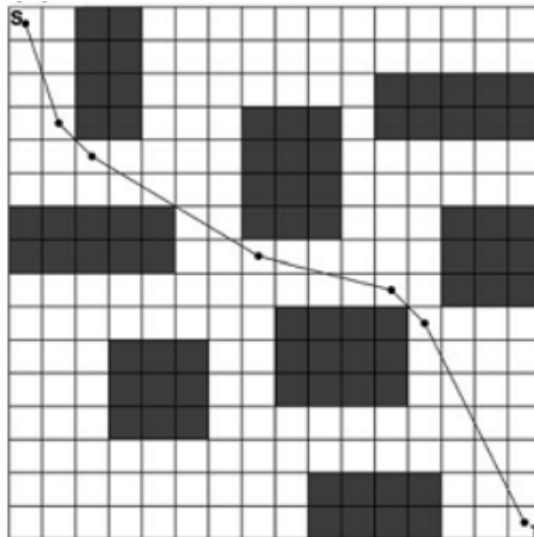


Рис. 10. Эксперимент 1

|                   | Binary | Decimal | Numbered |
|-------------------|--------|---------|----------|
| Path distance     | 28.40  | 25.07   | 26.30    |
| Generation number | 153    | 75      | 54       |
| Solution time(s)  | 2.7    | 1       | 0.7      |

Рис. 11. Таблица эксперимента 1

**Эксперимент 2.** Эксперимент 2 для окружения с 7 препятствиями. Параметры ГА те же.

Данные для различных методов кодирования представлены на рисунке ???. Видно, что десятичное и упорядоченное кодирование показывают лучшие результаты.

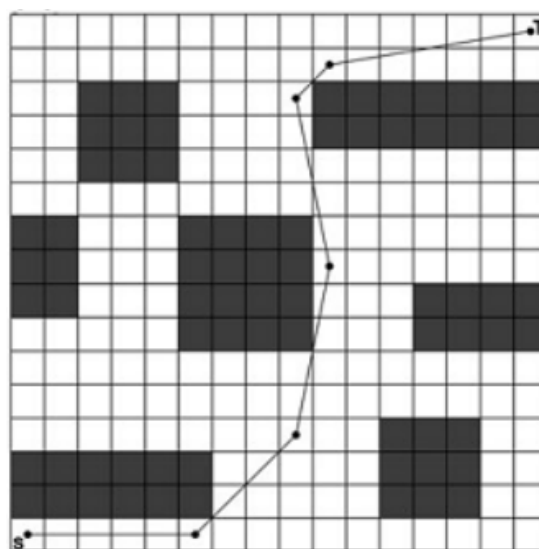


Рис. 12. Эксперимент 2

|                   | Binary | Decimal | Numbered |
|-------------------|--------|---------|----------|
| Path Distance     | 31.9   | 29.6    | 30.6     |
| Generation Number | 101    | 44      | 32       |
| Solution Time(s)  | 1.5    | 0.55    | 0.47     |

Рис. 13. Таблица эксперимента 2

## 3.2 Угловое кодирование пути

### Кодирование хромосом

Для кодирования пути робота эволюционные методы используют различные методы. Альтернативным рассмотренному ранее кодированию (узлов) вершин проходимого пути, является кодирование на основе углов направлений движения робота. Это можно сделать по-разному (с различной точностью). На Рис. 14 представлены 2 варианта кодирования направлений движения:

- а) 4 направления - робот может передвигаться по горизонталям и вертикалям.
- б) 8 направлений- робот может передвигаться также по диагоналям.

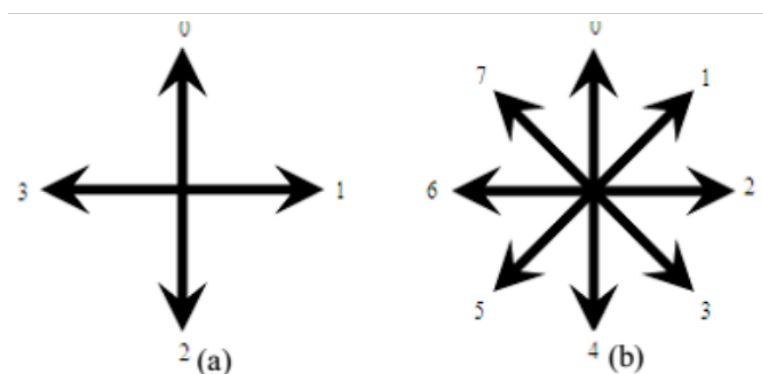


Рис. 14. Для кодирования направлений используются целые числа:  
а) (0,1,2,3), либо б) (0,1,2,3,4,5,6,7)

Тогда каждый ген в хромосоме представляет направление движения робота на следующем шаге.

В этом случае хромосома (см. Рис. 15) представляет «коллекцию» направлений движения робота в каждой точке, начиная от старта, и кончая финалом.

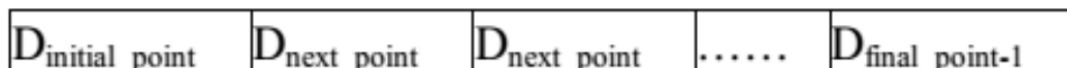


Рис. 15. Хромосома с угловым кодированием пути. D – целое число, определяющее направление движения на следующем шаге.

## Фитнесс-функция

Фитнесс-функция в задаче планирования пути робота определяется следующим образом:

$$\text{Cost}(P) = w_1 D + w_2 O + w_3 C, \quad (3)$$

где:

- $w_1, w_2, w_3$  – коэффициенты (константы), задающие вклад каждой компоненты;
- $D$  – расстояние (длина) пути;
- $O$  – число пересекаемых ячеек препятствий;
- $C$  – число изменений направлений движения.

Особь (маршрут) с минимальным значением фитнес-функции соответствует оптимальному пути. Очевидно, что увеличение числа изменений направления приводит к увеличению времени движения, а каждое пересечение препятствия сопровождается значительным штрафом.

## Эксперименты

**Эксперимент 1.** Метод апробирован на различного типа препятствиях с разным числом направлений движений (4 или 8). Начальная точка – (0,0). Конечная точка – (15, 9). В 1-м эксперименте использовались только статические препятствия.

На Рис. 16-17 представлены результаты для 2- вариантов направлений движений (4 или 8), которые показывают способность ГА решать данную проблему.

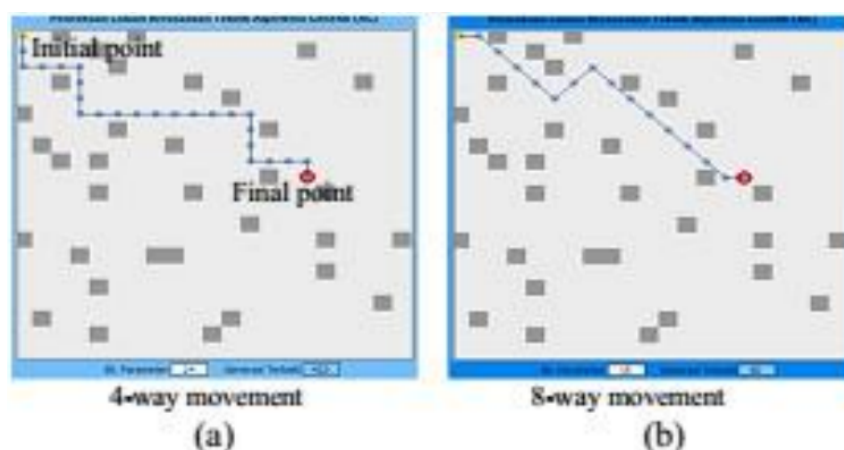


Рис. 16. Загроможденные препятствия.  
(a) 4 направления, (b) 8 направлений

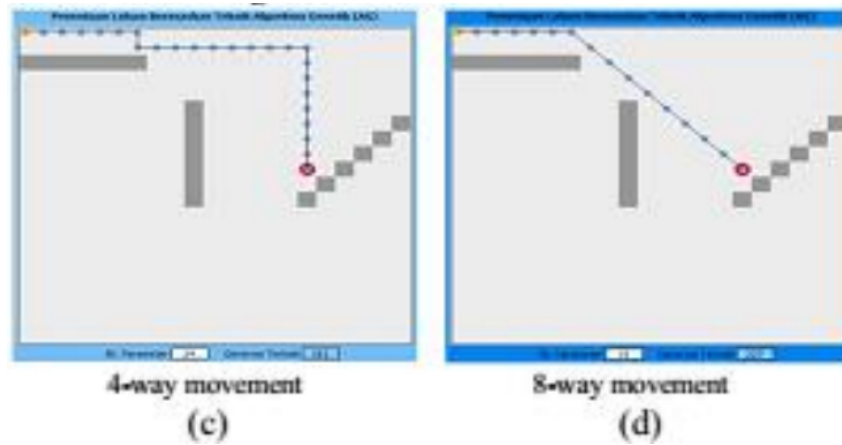


Рис. 17. Региональные препятствия.  
(с) 4 направления, (d) 8 направлений

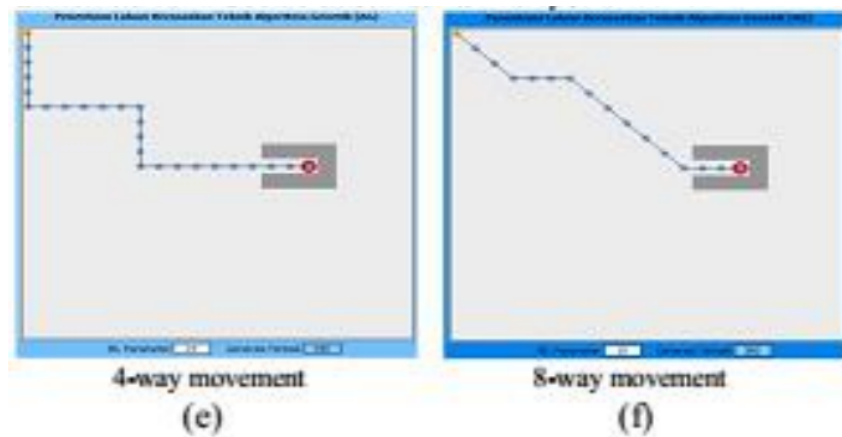


Рис. 18. Препятствия окружают конечную точку.  
(е) 4 направления, (f) 8 направлений

**Эксперимент 2.** Динамическую систему поиска пути можно разделить на две компоненты.

- Ищем маршрут по заранее заданной карте ещё до запуска робота (offline компонента).
- Если робот встретил новые препятствия, которых не было в исходной карте, то запускается новый алгоритм поиска с актуальной точки (online компонента).

Сначала генетический алгоритм был запущен для статической среды (см. Рис. 19).

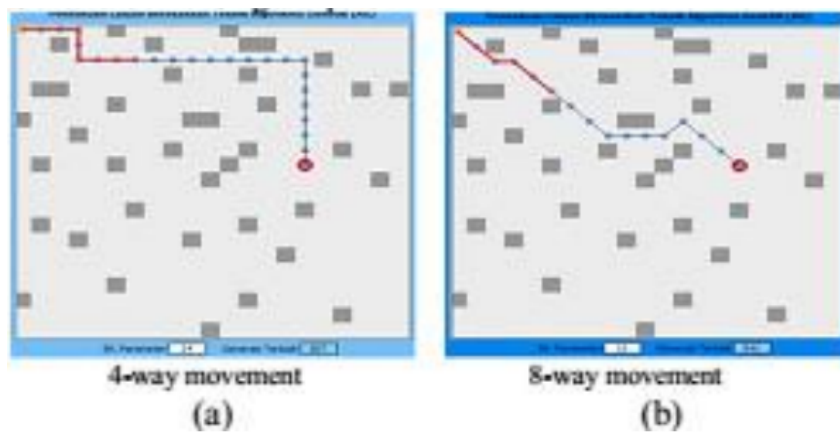


Рис. 19. Запуск для статической среды (offline компонента)

И далее добавлялись (случайно) 1 препятствие (см. Рис. 20).

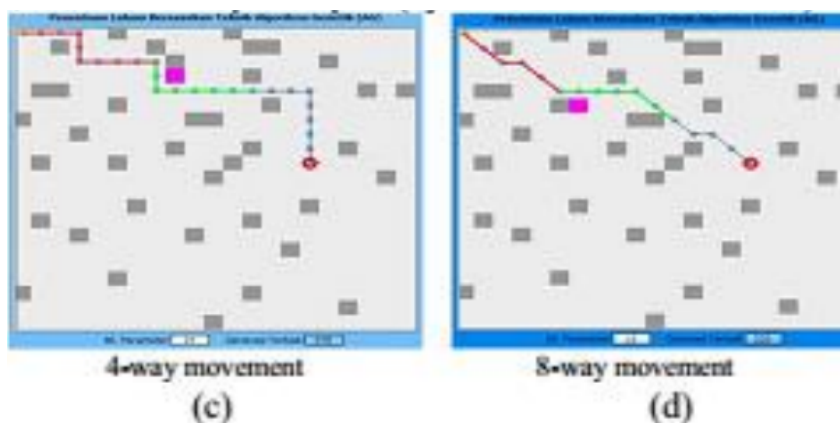


Рис. 20. Обновлённый путь после добавления ещё одного препятствия.

И ещё одно препятствие (см. Рис. 21).

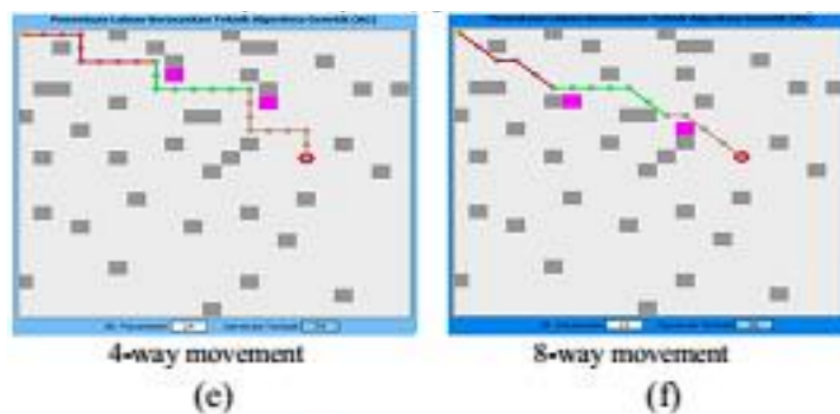


Рис. 21. Обновлённый путь после добавления второго препятствия.

# Заключение

В работе рассмотрено применение генетических алгоритмов к задаче планирования пути робота на дискретной карте и в форме углового (направленного) кодирования. Показано, что:

1. Для grid-представления десятичное и упорядоченное кодирование обеспечивают в среднем более короткие пути и/или меньшее время поиска по сравнению с двоичным кодированием при сопоставимых параметрах ГА ( $N = 60$ ,  $P_c = 1.0$ ,  $P_m = 0.1$ ).
2. Угловое кодирование с 8 направлениями, как правило, даёт более гибкие и короткие траектории, чем с 4 направлениями, однако требует контроля числа поворотов (штраф  $C$ ) для предотвращения «ломаных» маршрутов.
3. В динамическом сценарии поэтапное появление препятствий эффективно обрабатывается за счёт online-перепланирования: повторный запуск ГА из текущего положения восстанавливает коллизие-свободный путь без полной переразметки карты.

Ограничения подхода включают чувствительность к настройкам операторов и вероятностей, риск преждевременной сходимости и вычислительные затраты при частых проверках коллизий.

ГА представляют собой практичный и переносимый инструмент планирования пути в условиях неопределённости и динамики, при этом выбор кодирования и архитектуры фитнес-функции критически влияет на компромисс между качеством траектории и затратами вычислений.

## Список литературы

- [1] М.В. Прохоров, Ю.А. Скобцов. Многокритериальный генетический алгоритм построения пути робота в сложной среде. — Санкт-Петербург: СПбПУ, 2016. — 36с.
- [2] Матвеева А.В. Оптимизация построения маршрута с помощью генетического алгоритма // Финансовый университет при Правительстве Российской Федерации, 2022. — С. 10–12.