

Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики  
Факультет информационных технологий и программирования  
Кафедра «Компьютерные Технологии»

Р. И. Кавецкий

Отчет по лабораторной работе  
«Построение управляющих автоматов с помощью генетических алгоритмов»

Вариант № 13

Санкт-Петербург  
2011 г.

## Оглавление

Введение.....	3
1. Постановка задачи.....	4
1.1. Задача об умном муравье.....	4
2. Автомат Мили.....	5
3. Эволюционная стратегия.....	6
3.1. Функция приспособленности.....	6
3.2. Представление особи.....	6
3.3. Оператор скрещивания.....	7
3.4. Оператор мутации.....	7
4. Результаты.....	8
4.1. Алгоритм без использования гена мутации.....	8
4.2. Алгоритм с использованием гена мутации.....	9
4.3. Сравнение алгоритмов с геном мутации и без.....	10
Заключение.....	12
Источники.....	13

## **Введение**

В лабораторной работе изучается применение эволюционной стратегии для генерации конечных автоматов, решающих задачу об умном муравье. Целью работы является сравнение эффективности работы эволюционной стратегии при фиксированной вероятности мутации и при вероятности мутации, задаваемой в хромосоме особи.

При выполнении работы использовался программный комплекс для изучения методов глобальной оптимизации GLOpt, разработанный студентами кафедры «Компьютерные Технологии» НИУ ИТМО. На основе примеров, содержащихся в этом комплексе, был написан новый плагин, реализовывавший алгоритм эволюционной стратегии. Все исходные коды в данной работе написаны на языке программирования C#.

## 1. Постановка задачи

Задача лабораторной работы — построить автомат Мили из фиксированного числа состояний, решающий задачу об умном муравье для алгоритма с фиксированной вероятностью мутации и с вероятностью мутации, задаваемой в хромосоме особи.

### 1.1 Задача об умном муравье

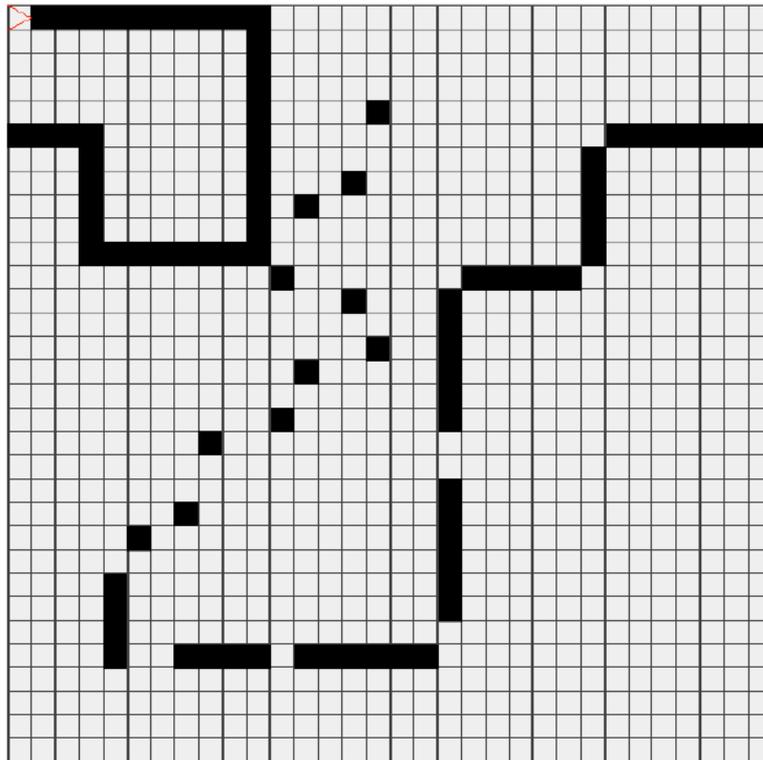


Рис. 1 — Поле, по которому ходит муравей

В задаче об умном муравье рассматривается фиксированное игровое поле размером  $32 \times 32$  клетки, расположенное на поверхности тора (рис. 1). Большая часть клеток пуста, остальные 89 содержат пищу. Муравей начинает движение из клетки, помеченной «Start». За ход муравей может выполнить одно из следующих действий:

- повернуть налево;
- повернуть направо;
- сделать шаг вперед и, если в новой клетке есть еда, съесть ее;
- ничего не делать.

Максимальное число ходов — 200. Задача муравья — за минимальное число ходов съесть как можно больше яблок.

## 2. Автомат Мили

Автомат Мили — конечный автомат, генерирующий выходные воздействия в зависимости от текущего состояния и входного воздействия. Можно представить автомат Мили как пятерку  $A = \langle S, q_0 \in S, X, Y, \delta \rangle$ , где

$S$  — множество состояний;

$q_0$  — стартовое состояние;

$X$  — множество входных воздействий;

$Y$  — множество выходных воздействий;

$\delta: S \times Y \rightarrow S \times Y$  — функция переходов.

### 3. Эволюционная стратегия

Для задачи поиска оптимального автомата, управляющего муравьем, используется алгоритм эволюционной стратегии. В начале случайным образом генерируется первое поколение особей. Далее начинается выполнение итеративного процесса, состоящего из нескольких фаз:

**Отбор элиты.** На первом этапе отбирается фиксированный процент элитных особей, с наибольшей функцией приспособленности. Эти особи переходят в следующее поколение без мутаций.

**Скрещивание.** Каждой особи случайным образом подбирается партнер. Каждая особь за одну итерацию цикла скрещивается только один раз.

**Мутация.** Из всех родителей и их потомков формируется промежуточное поколение, все особи которого, за исключением элиты, мутируют с фиксированной вероятностью или с вероятностью задаваемой в гене особи.

**Новое поколение.** Из промежуточного поколения методом отсечения особей с наименьшей функцией приспособленности, отбирается число особей, равное первоначальному.

Еще стоит отметить, что после каждого фиксированного числа поколений происходит большая мутация, заключающаяся в том, что все особи мутируют с некоторой фиксированной вероятностью большой мутации.

#### 3.1. Функция приспособленности

Функция приспособленности вычисляется для каждой особи и является мерой эффективности решения этой особью итоговой задачи.

Функция приспособленности вычисляется по формуле:

$Fitness = Apples - Steps / 200$ , где

*Apples* — число яблок, съеденных муравьем за 200 шагов.

*Steps* — номер шага, на котором муравей съел последнее яблоко.

Таким образом, муравей, съедающий большее число яблок за 200 шагов, чем другие, всегда будет иметь большее значение функции приспособленности, а из муравьев, съедающих одинаковое число яблок, большая функция приспособленности будет у того, кто потратил на поедание меньше шагов.

#### 3.2 Представление особи

Особь представляется автоматом Мили. Входными воздействиями являются два взаимоисключающих случая: перед муравьем есть еда в данный момент времени, или нет. Выходные воздействия — четыре действия, которые может совершить муравей (пойти вперед, повернуть направо, повернуть налево, ничего не делать). Каждому переходу ставится в соответствие одно из них.

Автомат хранится в виде графа переходов, где вершины — это состояния, ребра — это переходы. Из каждой вершины выходит по два ребра. Переход по первому осуществляется, когда перед муравьем есть еда, переход по второму, когда ее нет. При переходе в очередное состояние агент выполняет соответствующее переходу действие.

Для случая когда мутация задается в гене, в особь добавляется дополнительная переменная, отвечающая за вероятность мутации особи.

### 3.3 Оператор скрещивания

Оператор скрещивания принимает на вход две особи (родители), и возвращает две особи (потомки). В работе применялся однородный метод скрещивания, который происходит следующим образом:

Пусть у нас есть два родителя  $S1$  и  $S2$  и два потомка  $P1$  и  $P2$ .

- если обозначить стартовое состояние особи  $A$  как  $A.start$ , то после применения оператора скрещивания с равной вероятностью будет справедливо:

$P1.start = S1.start$  и  $P2.start = S2.start$

либо

$P1.start = S2.start$  и  $P2.start = S1.start$

- если обозначить переход из состояния  $i$  в автомате  $A$  по входному воздействию «вперед есть пища» как  $A(i, 0)$ , а по «вперед нет пищи» как  $A(i, 1)$ , то с равной вероятностью будет справедливо одно из четырех:

$P1(i, 0) = S1(i, 0)$ ,  $P1(i, 1) = S1(i, 1)$ ,  $P2(i, 0) = S2(i, 0)$ ,  $P2(i, 1) = S2(i, 1)$

$P1(i, 0) = S1(i, 0)$ ,  $P1(i, 1) = S2(i, 1)$ ,  $P2(i, 0) = S2(i, 0)$ ,  $P2(i, 1) = S2(i, 1)$

$P1(i, 0) = S2(i, 0)$ ,  $P1(i, 1) = S1(i, 1)$ ,  $P2(i, 0) = S1(i, 0)$ ,  $P2(i, 1) = S2(i, 1)$

$P1(i, 0) = S2(i, 0)$ ,  $P1(i, 1) = S2(i, 1)$ ,  $P2(i, 0) = S1(i, 0)$ ,  $P2(i, 1) = S1(i, 1)$

- если обозначить вероятность мутации, задаваемую в гене особи, как  $A.mutation$ , то вероятность мутации потомков:

$S1.mutation = S2.mutation = (P1.mutation + P2.mutation) / 2$

### 3.4 Оператор мутации

Мутация — это случайное изменение некоторых признаков особи.

При имеющемся представлении особей можно изменить шесть различных признаков:

1. стартовое состояние;
2. выходное воздействие, поставленное в соответствие переходу;
3. входное воздействие, поставленное в соответствие переходу;
4. состояние, в которое ведет ребро;
5. состояние, из которого ведет ребро;
6. мутация, задаваемая в гене.

В ходе предварительных экспериментальных исследований было получено наиболее оптимальный вариант оператора мутации. Он производит с особью одно из трех преобразований:

- меняет стартовое состояние;
- изменяет переход из произвольного состояния в другое произвольное состояние по произвольному входному воздействию с произвольным выходным воздействием (комбинация мутаций с номерами 2–5 из предыдущего списка);
- изменяет вероятность мутации, задаваемой в гене.

## 4. Результаты

Алгоритм запускался по несколько раз с различными параметрами. Постоянными были:

- размер популяции (100);
- число состояний в автомате (8);
- вероятность большой мутации (0,1);
- число поколений, после которых происходит большая мутация (500).

Значение остальных параметров будет указываться для каждого случая отдельно.

### 4.1 Алгоритм без использования гена мутации

Параметры запуска:

- вероятность мутации (0,5);
- процент элиты (0,5).

Был получен автомат, достаточно эффективно решающий поставленную задачу: муравей съедает 88 единицы пищи за 193 хода. На рис. 2 предоставлен график (красного цвета) зависимости максимального значения функции приспособленности среди особей популяции от времени.

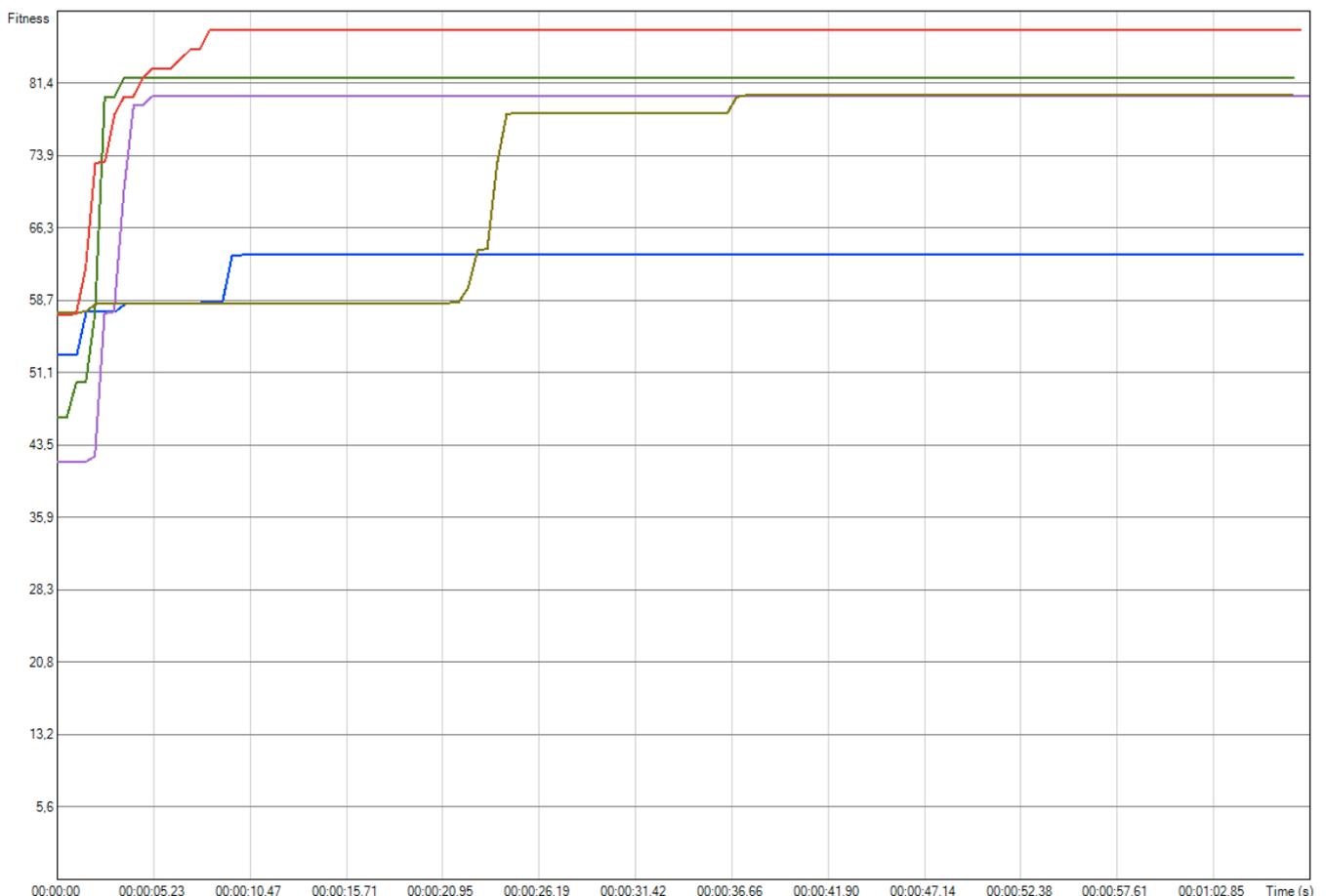


Рис. 2 — Графики зависимости максимального значения функции приспособленности от времени (алгоритм без использования гена мутации)

## 4.2 Алгоритм с использованием гена мутации

Параметры запуска:

- вероятность мутации (задается в гене);
- процент элиты (0,2).

Был получен автомат, достаточно эффективно решающий поставленную задачу: муравей съедает 88 единицы пищи за 189 ходов. На рис. 3 предоставлен график (зеленого цвета) зависимости максимального значения функции приспособленности среди особей поколения от времени.

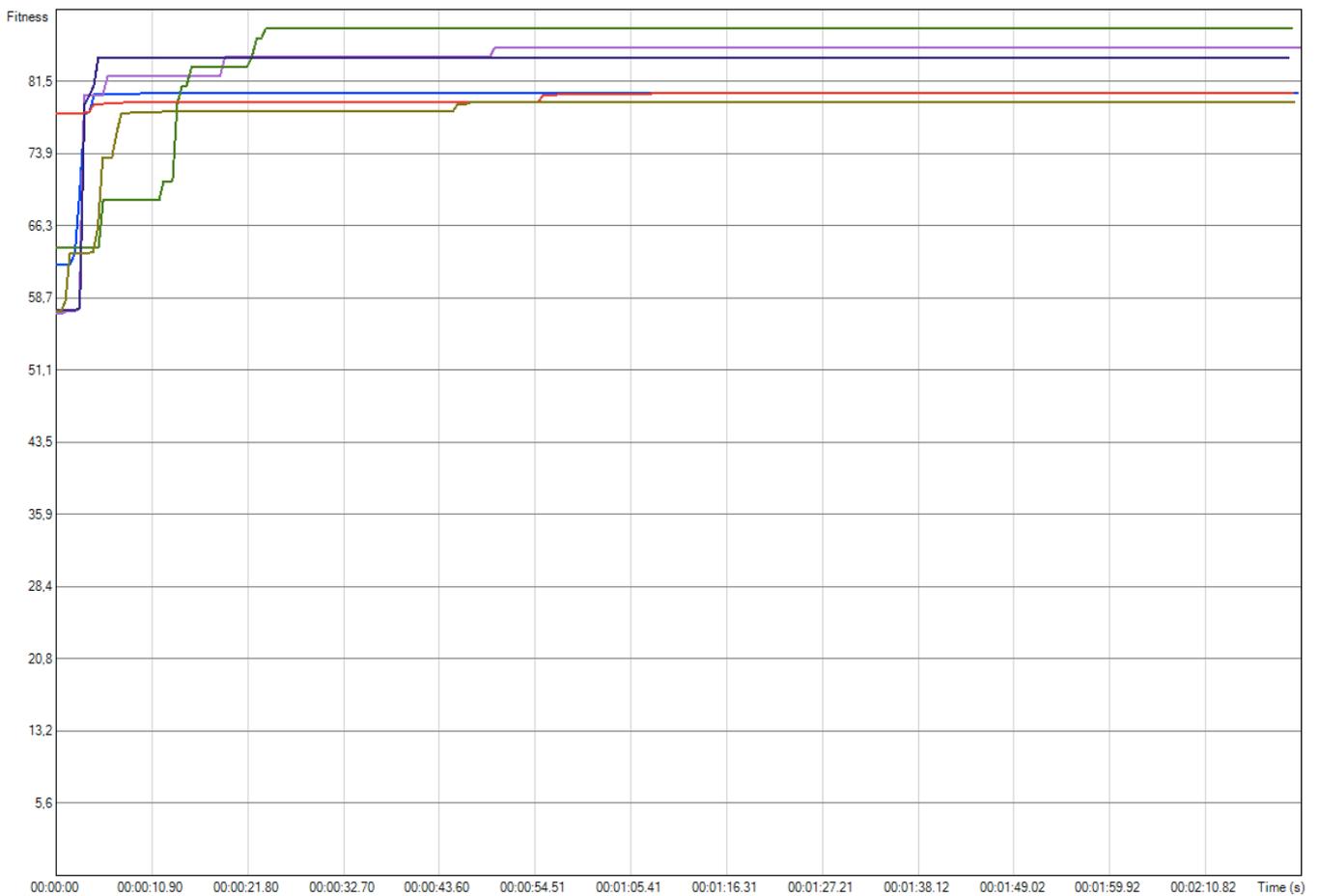


Рис. 3 — Графики зависимости максимального значения функции приспособленности от времени (алгоритм с использованием гена мутации)

### 4.3 Сравнение алгоритмов с геном мутации и без

Алгоритмы с использованием гена мутации и без использования гена мутации были запущены одновременно, с одинаковыми параметрами. На рис. 4–6 графики зависимости максимальной функции приспособленности от времени. Каждые две пары слева различаются лишь использованием гена мутации (первый из пары — без гена, второй — с геном). Красным овалом выделен лучший результат из пары.

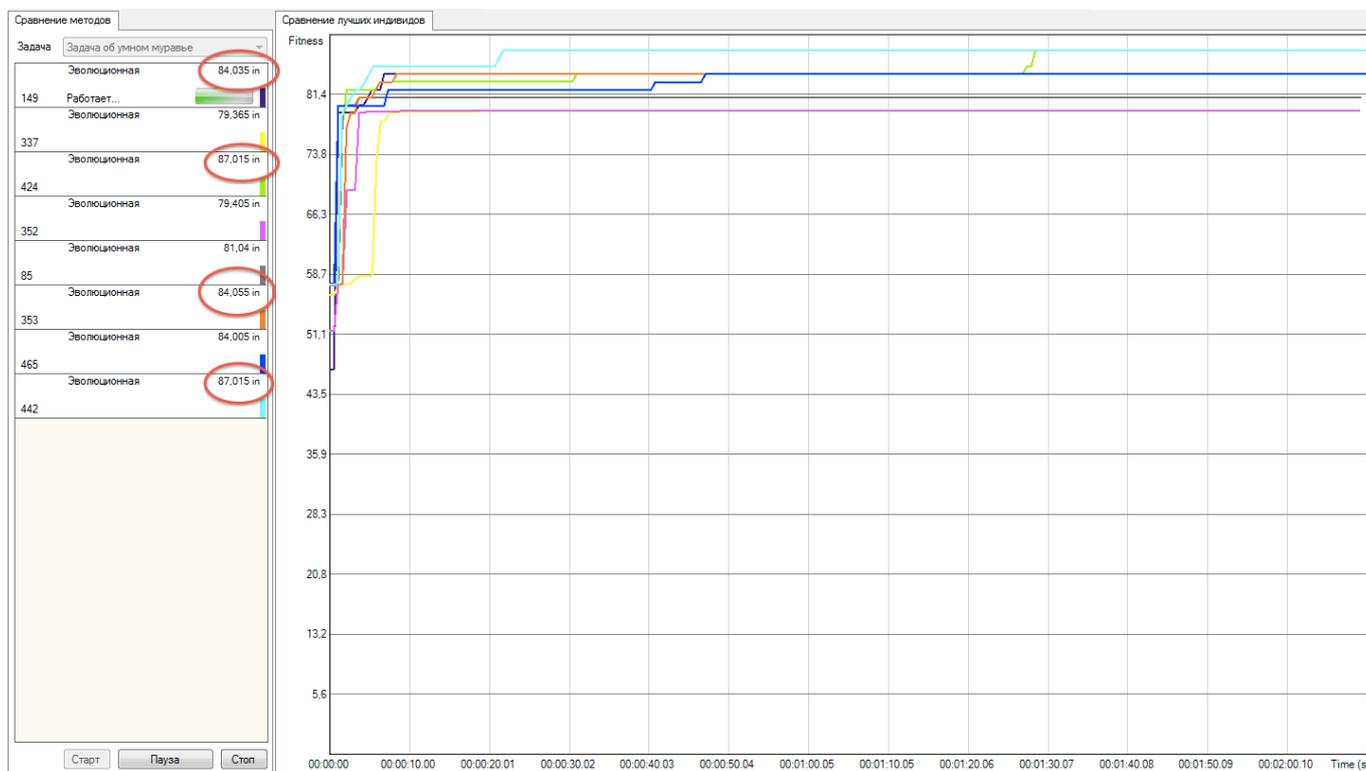


Рис. 4 — Графики зависимости максимальной функции приспособленности от времени

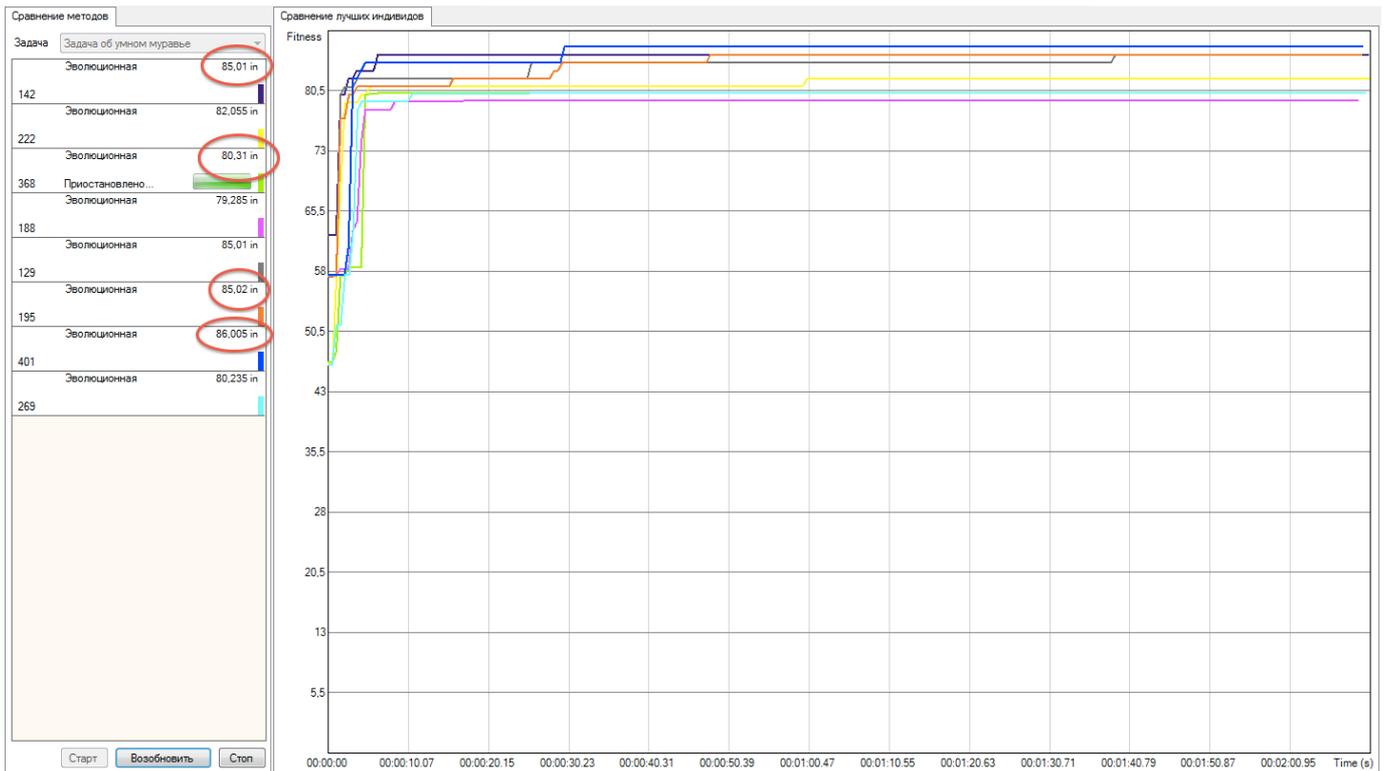


Рис. 5 — Графики зависимости максимальной функции приспособленности от времени

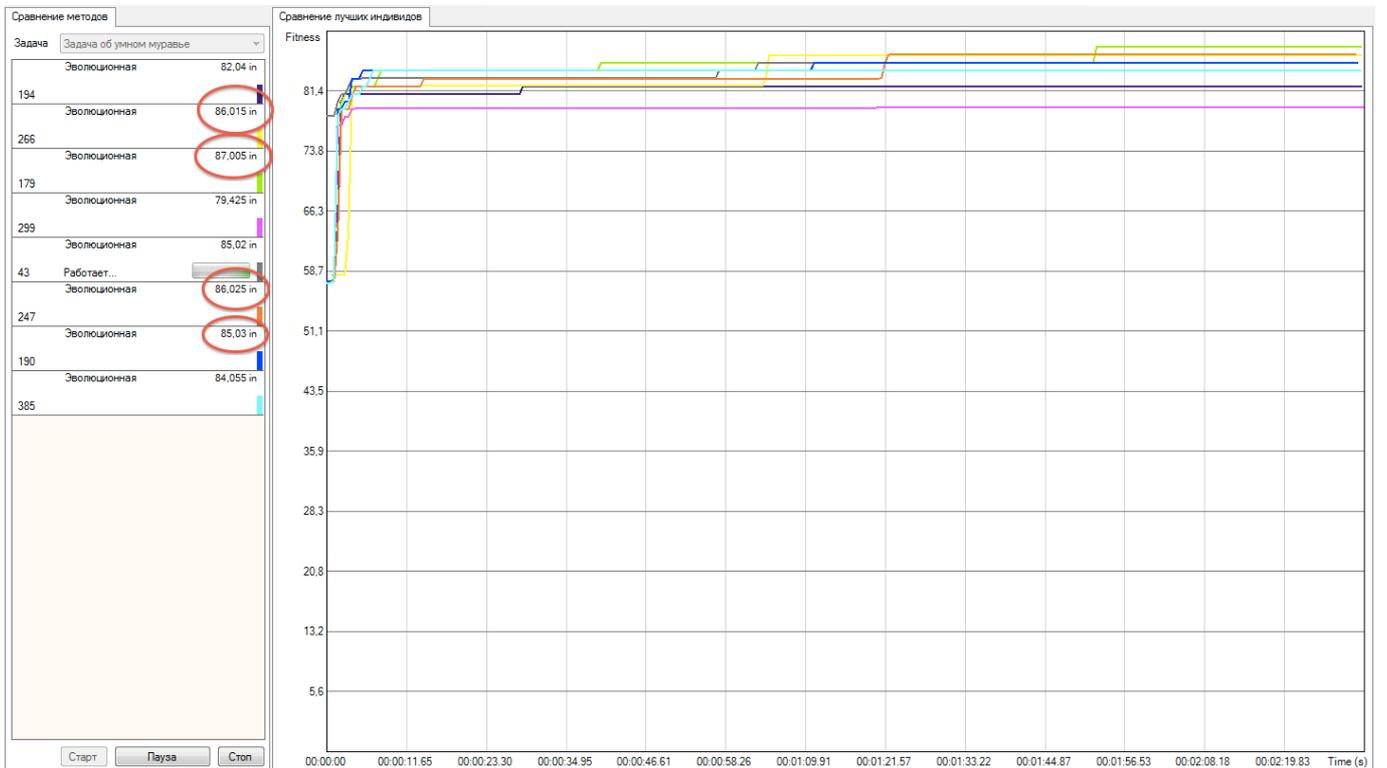


Рис. 6 — Графики зависимости максимальной функции приспособленности от времени

## **Заключение**

Результаты лабораторной работы показали, что алгоритм эволюционной стратегии достаточно эффективен для поиска решения задачи об умном муравье и имеет очень высокую скорость поиска оптимального решения. Использование гена мутации не особенно влияет на результат решения данной задачи, и в зависимости от значения остальных параметров, может как улучшить результат, так и сделать его значительно хуже. Как выяснилось в ходе работы, гораздо больше на результат влияет реализация оператора мутации, а не вероятность мутации особи.

## Источники

1. Тяхти А. С., Чебатуркин А. А. Программный комплекс для изучения методов искусственного интеллекта «Виртуальная лаборатория «GLOpt». [http://is.ifmo.ru/genalg/labs\\_2010-2011/GLOpt\\_instruction.pdf](http://is.ifmo.ru/genalg/labs_2010-2011/GLOpt_instruction.pdf).
2. Бедный Ю. Д., Шалыто А. А. Применение генетических алгоритмов для построения автоматов в задаче «Умный муравей». [http://is.ifmo.ru/works/\\_ant.pdf](http://is.ifmo.ru/works/_ant.pdf).
3. Яминов Б. Генетические алгоритмы. <http://rain.ifmo.ru/cat/view.php/theory/unsorted/genetic-2005>.
4. Рутковская Д., Пилиньский М. Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы. [http://www.sernam.ru/book\\_gen.php](http://www.sernam.ru/book_gen.php).
5. Генетический алгоритм на примере бота Robocode. <http://habrahabr.ru/blogs/DIY/111811>.