Санкт-Петербургский государственный университет информационных технологий, механики и оптики

Кафедра компьютерных технологий

А. В. Захарчук

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

Вариант №10

Санкт-Петербург 2011

Оглавление

Введение

В данной работе изучается применение генетических алгоритмов с различными функциями скрещивания для построения конечных автоматов на примере задачи «Умный муравей».

При выполнении работы использовался программный комплекс для изучения методов глобальной оптимизации GlOpt [1], разработанный студентами кафедры «Компьютерные технологии» НИУ ИТМО.

1. Постановка задачи

1.1. Задача об умном муравье

В задаче об «Умном муравье» рассматривается фиксированное поле 32 на 32 клетки(рис. 1), расположенное на поверхности тора. Большая часть клеток пуста, остальные 89 содержат пищу. Имеется агент, муравей, способный видеть, есть ли перед ним пища, или нет. Он начинает движение с клетки, помеченной как «старт» и может совершать следующие действия:

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

Максимальное число ходов – 200. Муравей должен съесть как можно больше пищи, совершив при этом как можно меньше ходов.

В данной лабораторной работе эта задача решается путем построения с помощью генетических алгоритмов конечного автомата, управляющего действиями муравья.

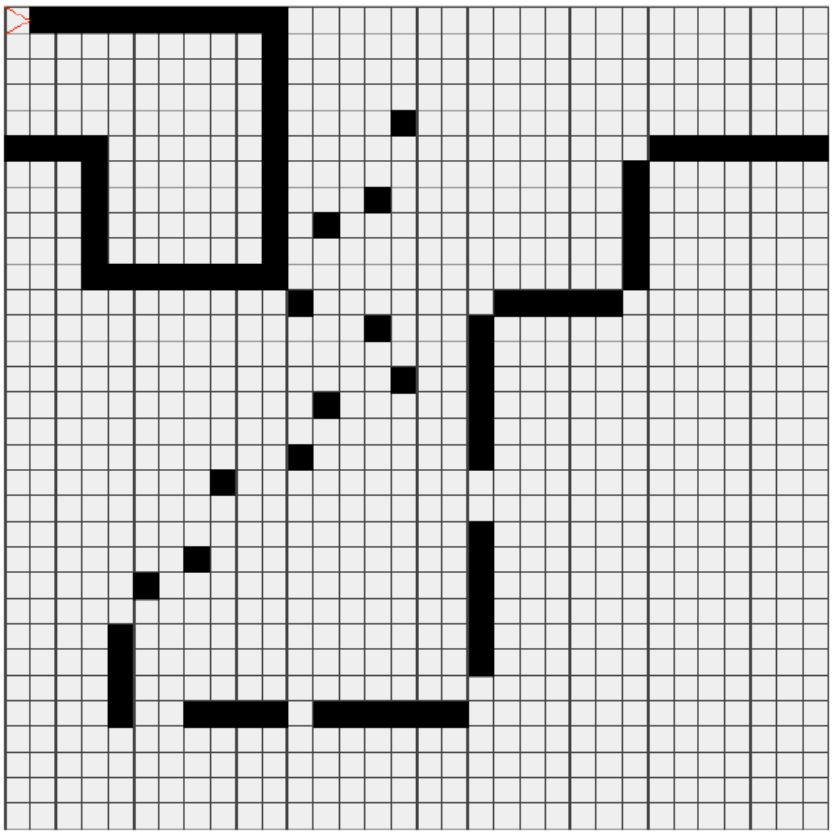

Рис. 1 – Игровое поле

1.2. Конечный автомат Мура

Это конечный автомат, генерирующий выходные воздействия в зависимости от текущего состояния. Функция переходов зависит от входных воздействий и текущего состояния.

Формально, конечным автоматом Мура называют совокупность <S, X, Y, s0, δ, μ >, где:

- S – конечное множество состояний;
- X - конечное множество входных воздействий;
- Y – конечное множество выходных воздействий;
- s0 – стартовое состояние;
- δ - функция перехода. Ее аргументы – состояние и входной символ, результат – состояние: s(t+1) = δ(s(t), x(t));
- μ - функция, ставящая выходное воздействие в соответствие состоянию: y(t) = μ(s(t)); Здесь t = 0, 1, 2, ... - абстрактное время.

В данном случае входными воздействиями являются два взаимоисключающих случая: есть ли перед муравьем еда в данный момент времени, или нет. Выходные воздействия – это те четыре действия, которые может совершать муравей. Каждому состоянию ставится в соответствие одно из них.

В итоге 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 – если ее нет. При переходе в очередное состояние агент выполняет соответствующее ему действие.

2. Реализация

В данной работе используется традиционный генетический алгоритм.

В роли особей выступают конечные автоматы в вышеуказанном представлении.

2.1. Функция приспособленности

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

Чтобы оценить, насколько хорошо полученный автомат управляет действиями муравья, необходимо смоделировать поведение агента и оценить результаты. Таким образом, запустив муравья на имеющемся поле, можно получить следующие показатели:

- a – количество съеденной за 200 ходов пищи;
- t – номер последнего из ходов, на которых была съедена пища (ходы нумеруются с 1).

Примем за значение функции приспособленности следующую величину:

$F = a - t/200$

Легко видеть, что она вполне корректна, и значение ее тем выше, чем эффективней автомат решает задачу.

2.2. Метод мутации

Мутация в генетическом алгоритме – это случайное изменение некоторых признаков особи.

При имеющемся представлении особей у них можно изменить четыре признака:

- стартовое состояние;
- действие, поставленное в соответствие случайно выбранному состоянию;
- состояние, в которое ведет случайно взятое ребро;
- соответствие переходов «есть пища» и «нет пищи» для случайно выбранного состояния.

Последнее означает, что те два ребра, которые ведут из случайной вершины, обмениваются состояниями, в которые они ведут.

Все указанные варианты действий равновероятны.

2.3. Метод скрещивания

Оператор скрещивания принимает на вход две особи – родителей, и возвращает также две особи – потомков.

2.3.1. Одноточеный кроссовер

Представим автомат в виде строки битов.

Сначала, случайным образом выбирается одна из точек разрыва. (Точка разрыва - участок между соседними битами в строке.) Обе родительские структуры разрываются на два сегмента по этой точке. Затем, соответствующие сегменты различных родителей склеиваются и получаются два генотипа потомков.

Например, предположим, один родитель состоит из 10 нолей, а другой - из 10 единиц. Родители и их потомки показаны ниже.

Родитель №1 — 0000000000 — 000~0000000
Родитель №2 — 1111111111 — 111~1111111

Потомок №1 — 000~1111111 — 0001111111
Потомок №2 — 111~0000000 — 1110000000

2.3.2. Двухточечный кроссовер

Представим автомат в виде строки битов.

Сначала, случайным образом выбираются две точки разрыва. (Точка разрыва - участок между соседними битами в строке.) Обе родительские структуры разрываются на три сегмента по этим двум точкам. Затем, соответствующие сегменты различных родителей склеиваются и получаются два генотипа потомков.

Например, предположим, один родитель состоит из 10 нолей, а другой - из 10 единиц. Родители и их потомки показаны ниже.

Родитель №1 — 0000000000 — 000~0000~000
Родитель №2 — 1111111111 — 111~1111~111

Потомок №1 — 000~1111~000 — 0001111000
Потомок №2 — 111~0000~111 — 1110000111

2.3.3. Однородный кроссовер

Обозначим родителей как S1 и S2, а потомков - как P1 и P2.
Однородный кроссовер происходит следующим образом:

1. Если обозначить стартовое состояние особи A как A.start, то с равной вероятностью либо P1.start = S1.start и P2.start=S2.start, либо P1.start = S2.start и P2.start = S1.start.

2. Если обозначить переход из состояния i в автомате A по входному воздействию «впереди есть пища» как A(i, 0), а по «впереди 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)

2.4. Генерация нового поколения

Начальное поколение генерируется из особей, заданных случайным образом. Рассмотрим процесс генерации нового поколения из текущего:

1. В новое поколение отправляется определенная часть всей популяции – элитизм.
2. Из исходного поколения выбирается количество недостающих особей до размера поколения.
3. К выбранным на шаге №2 особям применяется оператор скрещивания и мутации.
4. Восстанавливается размер популяции отсечением худших особей.

3. Результаты работы

Алгоритм был запущен с параметрами:

- 8 состояний в автомате;
- вероятность мутации 0.1;
- размер поколения равен 100 особям;
- элитизм 0.1.

Одноточечный кроссовер

В ходе работы был получен конечный автомат, достаточно эффективно решающий поставленную задачу: муравей съедает 84 единицы пищи за 199 ходов. На рис. 2 предоставлен график максимального значения функции приспособленности среди особей поколения в зависимости от времени.

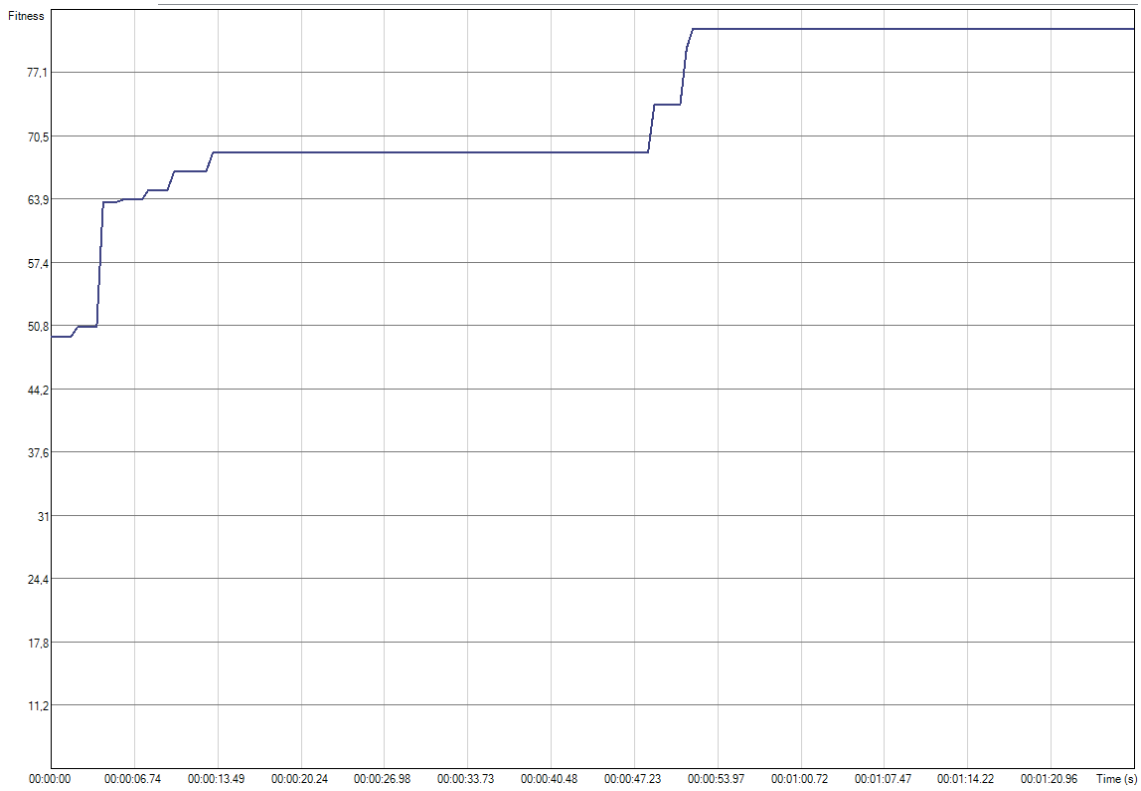

Рис. 2 – максимальное значения функции в зависимости от времени для одноточечного кроссовера

Двухточечный кроссовер

В ходе работы был получен конечный автомат, достаточно эффективно решающий поставленную задачу: муравей съедает 83 единицы пищи за 197 ходов. На рис. 3 предоставлен график максимального значения функции приспособленности среди особей поколения в зависимости от времени.

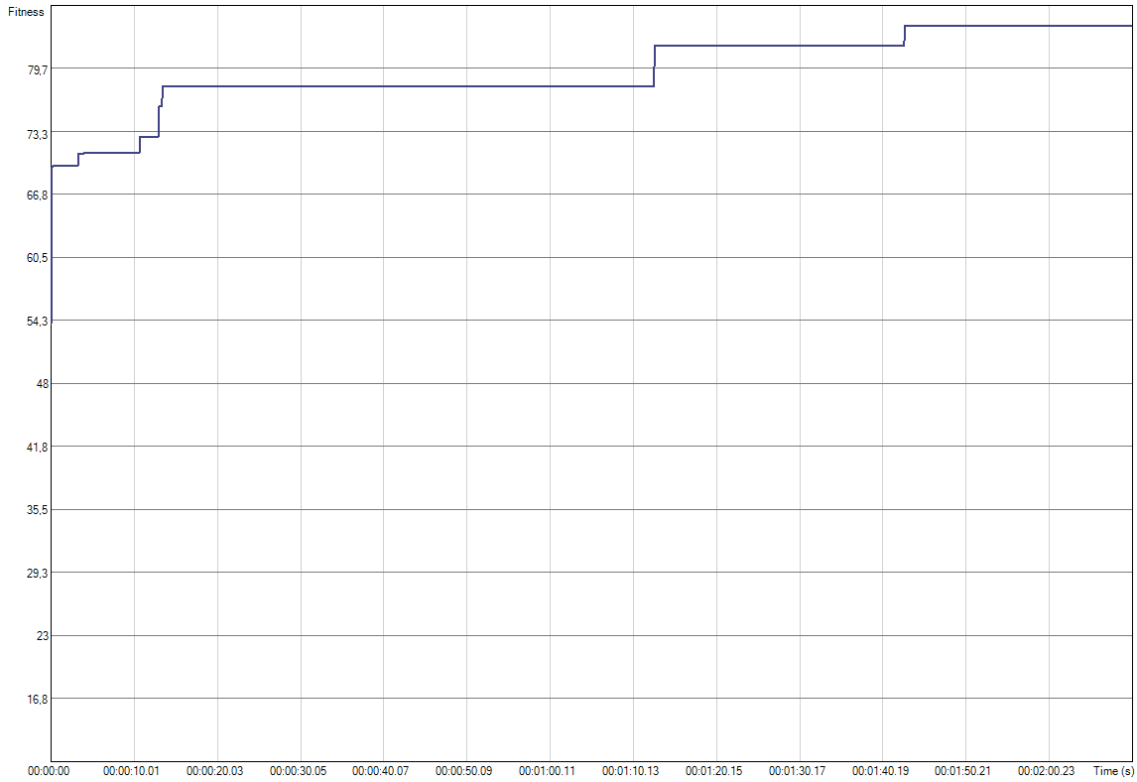

Рис. 3 – максимальное значения функции в зависимости от времени для двухточечного кроссовера

Однородный кроссовер

В ходе работы был получен конечный автомат, достаточно эффективно решающий поставленную задачу: муравей съедает 83 единицы пищи за 196 ходов. На рис. 4 предоставлен график максимального значения функции приспособленности среди особей поколения в зависимости от времени.

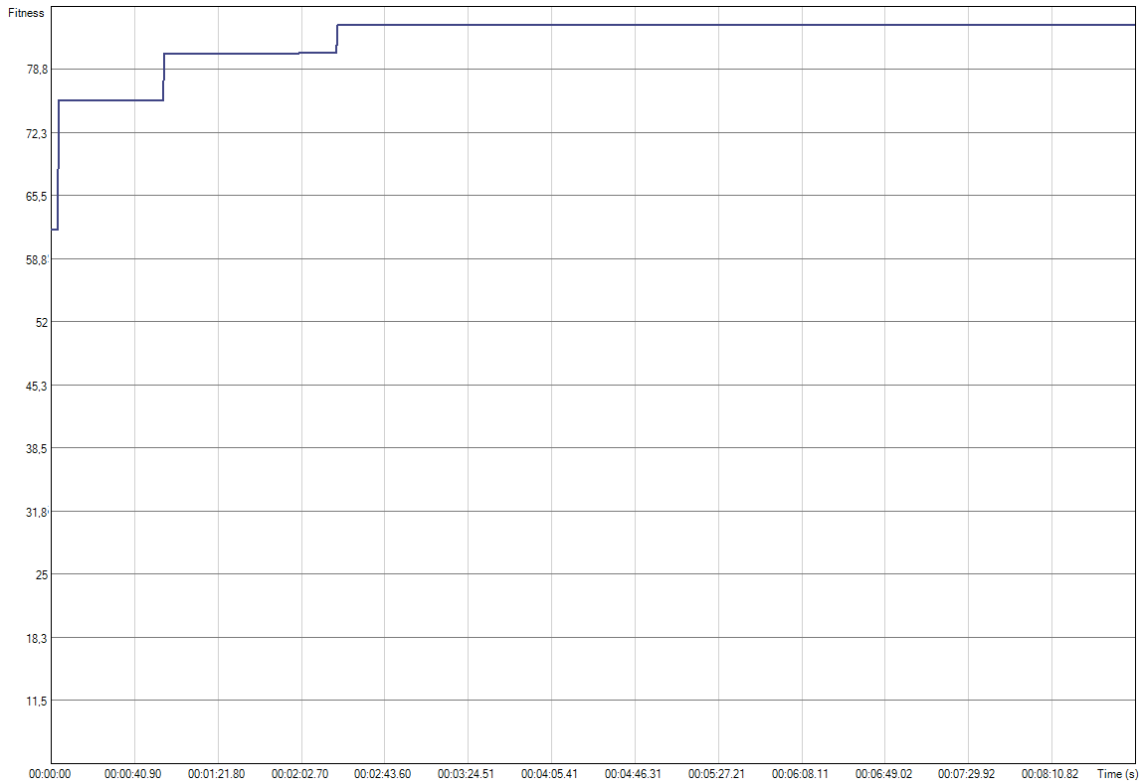

Рис. 4 – максимальное значения функции в зависимости от времени для однородного кроссовера

Сравнение трех кроссоверов

Для каждого из кроссоверов были созданы 3 особи. Практически все особи справились с поставленной задачей. Результат запуска предоставлен на рис. 5.

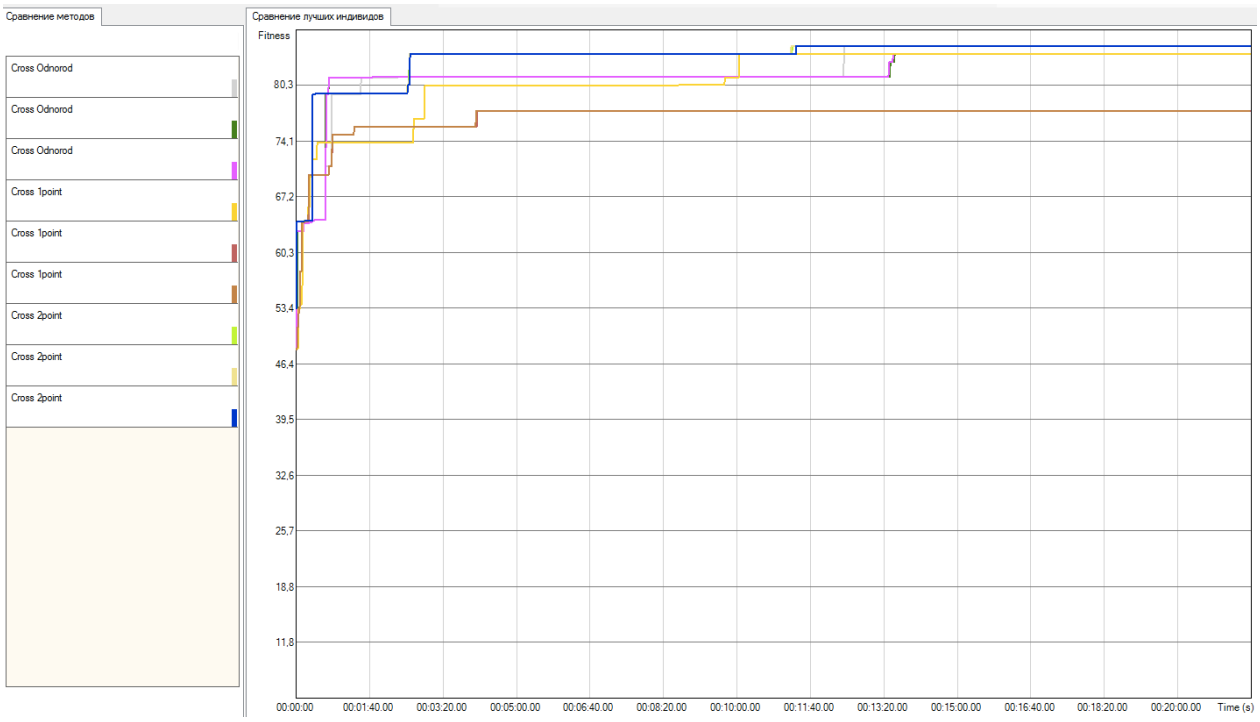

Рис. 5 – Сравнение трех кроссоверов

Полученный автомат

На рис. 6 представлен один из полученных автоматов Мура в виде графа переходов, в котором F (¬F) над переходом – наличие (отсутствие) еды перед муравьем.

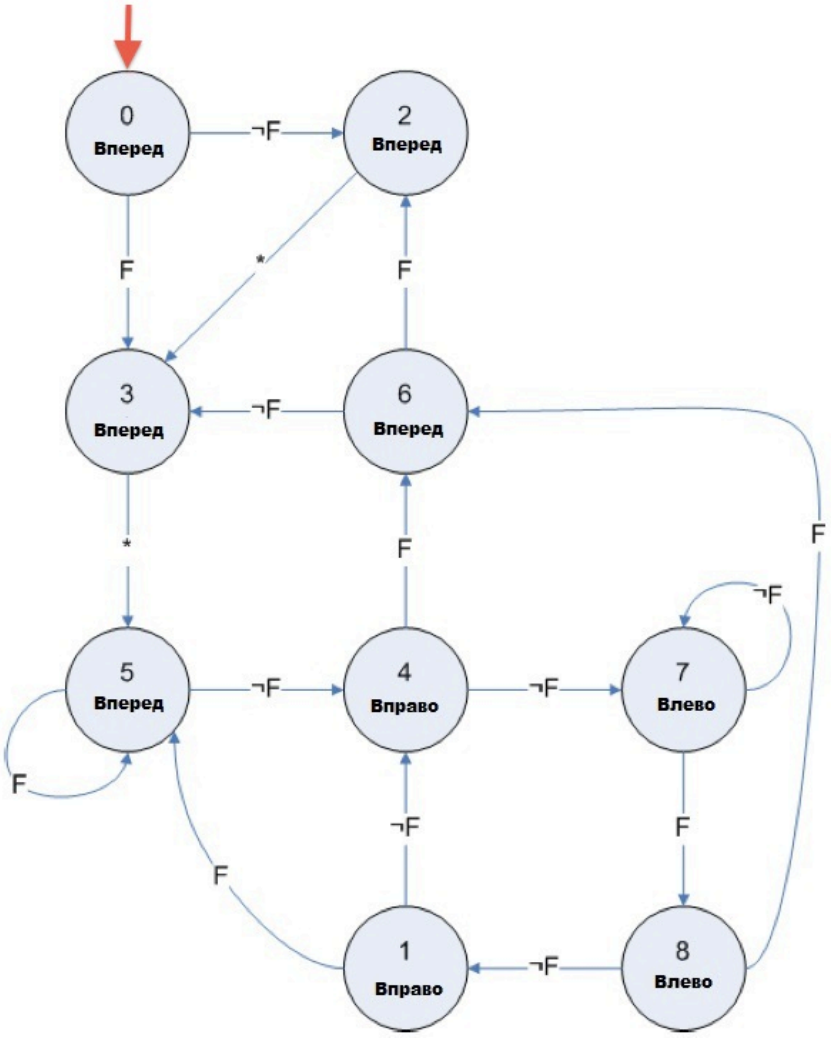

Рис. 6 – Полученный автомат

Заключение

Результаты лабораторной показывают, что использование любого из трех кроссоверов имеет довольно неплохую эффективность для решения данной задачи.

Изменив параметры алгоритма на следующие:

- 8 состояний в автомате;
- вероятность мутации 0.1;
- размер поколения равен 10000 особям;
- элитизм 0.1;

Было замечено, что самая быстрая сходимость особей присуще одноточечному кроссоверу. За ним следует двухточечный. Наименьшую сходимость показал алгоритм с однородным кроссовером.

Источники

1. Документация к комплексу для изучения методов глобальной оптимизации GlOpt. http://is.ifmo.ru/courses/_giopt-src.rar
2. Бедный Ю. Д., Шалыто А. А. Применение генетических алгоритмов для построения автоматов в задаче «Умный муравей». http://is.ifmo.ru/works/_ant.pdf
3. Яминов Б. Генетические алгоритмы. http://rain.ifmo.ru/cat/view.php/theory/unsorted/genetic-2005