

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

Н. С. Буланова

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

Вариант №2

Санкт-Петербург
2011 г

Оглавление

Введение	3
1. Постановка задачи	4
1.1. Задача о роботе, обходящем препятствия.....	4
2. Реализация	5
2.1. Представление особи.....	5
2.2. Оператор скрещивания	5
2.3. Оператор мутации	5
2.4. Функция приспособленности.....	6
2.5. Метод генерации очередного поколения.....	6
3. Результаты работы.....	6
Заключение	10

Введение

В данной лабораторной работе требуется найти зависимость эффективности работы генетического алгоритма построения автомата, решающего задачу о роботе, обходящего препятствия, от числа особей в поколении. Под эффективностью понимается максимальное значение функции приспособленности.

Задача реализована на языке программирования C++ без использования виртуальных лабораторий.

1. Постановка задачи

Задача данной лабораторной работы — исследовать влияние числа особей в поколении на эффективность работы генетического алгоритма, строящего автомат Мили из десяти состояний для решения задачи о роботе, обходящем препятствия. Сгенерированный данным автоматом робот должен доходить до цели не более чем за 200 шагов.

1.1. Задача о роботе, обходящем препятствия

Дано поле 32×32 клетки, представляющее собой фиксированный лабиринт с препятствиями. Робот видит только клетку впереди себя.

За один ход робот может:

- пойти вперед (если впереди препятствие, то робот останется на месте);
- повернуть налево;
- повернуть направо.

Задача робота — добраться до цели за наименьшее число ходов, не превышающее двухсот.

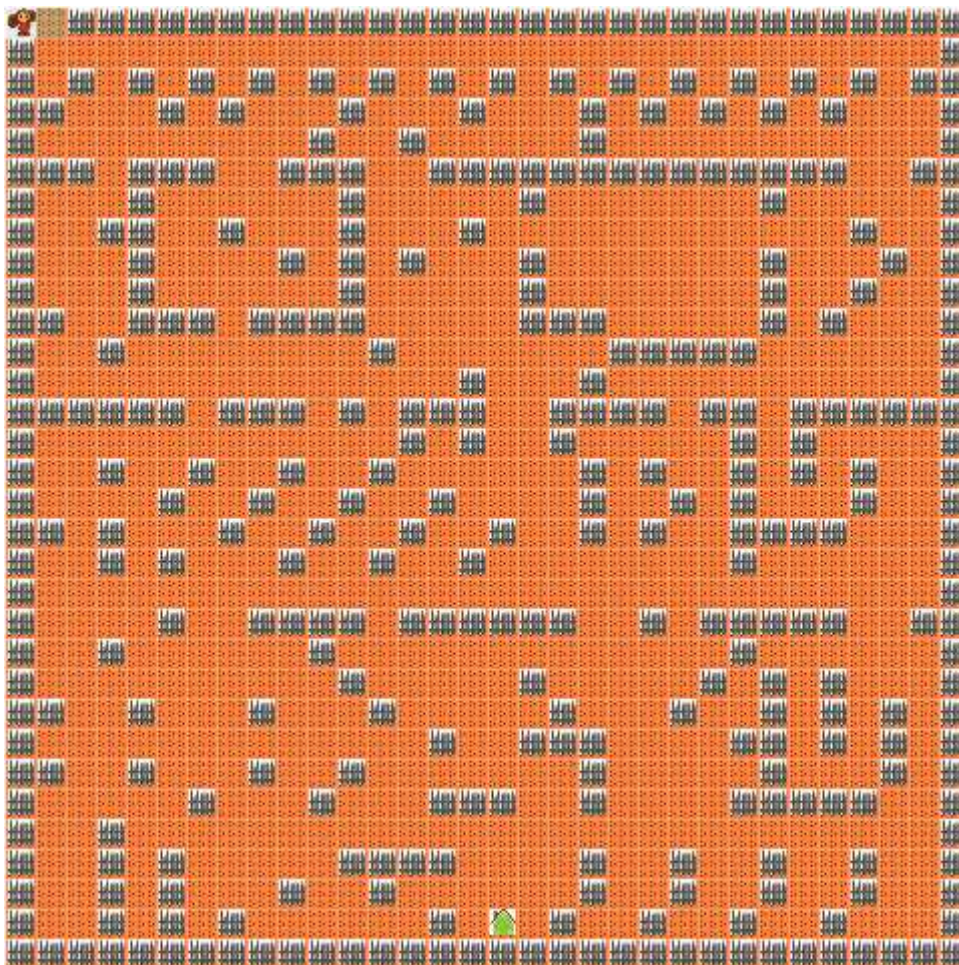


Рис. 1. Поле, по которому ходит робот

2. Реализация

Для задачи поиска оптимального автомата, управляющего роботом, применяется генетический алгоритм. Работа генетического алгоритма состоит из нескольких фаз. Вначале происходит генерация первого поколения особей. Далее алгоритм начинает выполнение итеративного процесса – на каждой итерации строится следующее поколение из предыдущего.

При этом применяются процедуры:

- отбор – из предыдущего поколения выбирается часть особей; для сравнения особей между собой алгоритм использует функцию приспособленности, сопоставляющую каждой особи число, определяющее ее приспособленность;
- скрещивание – из двух особей-родителей создаются две новые особи;
- мутация – случайным образом изменяется строение особи.

Опишем более подробно алгоритм, применяемый в данной работе.

2.1. Представление особи

Особями в данном алгоритме являются автоматы Мили. Автомат Мили представляется в виде графа переходов. Непосредственно в коде программы используется двумерный массив `states[][]`, который хранит переход (выходное воздействие и новое состояние) для каждой пары, состоящей из текущего состояния автомата и входного сигнала. В задаче о роботе, обходящем препятствия, значениями входной переменной являются 0 и 1 – присутствие или отсутствие препятствия в клетке перед роботом.

2.2. Оператор скрещивания

При скрещивании алгоритм порождает две новые особи из двух особей родителей. Состояния автоматов нумеруются числами от 1 до `size` (у всех автоматов одинаковое число состояний, в нашем случае десять).

Рассмотрим процедуру скрещивания: сначала для входного воздействия равного 0, затем для равного 1, пройдясь по всем состояниям с таким входным воздействием, с вероятностью 0,5 поменяем в родительских особях поля выходного воздействия и нового состояния

2.3. Оператор мутации

Оператор мутации реализован следующим образом. Для каждого перехода из каждого состояния действие на этом переходе и номер следующего состояния заменяются случайными с вероятностью 0,05. С той же вероятностью может измениться начальное состояние автомата.

2.4. Функция приспособленности

Вначале исполнения программы рассчитываются длины минимальных путей из каждой из свободных клеток поля до цели. Они хранятся в массиве. Функция приспособленности (ФП) имеет следующий вид:

$$getFitnessFunc(A) = \begin{cases} -dist, \\ 200 - i, \end{cases}$$

где A — автомат, принадлежащий пространству поиска,
 $dist$ — длина минимального пути из точки, в которой робот сделал двухсотый шаг до цели,
 i — число шагов, потребовавшееся роботу для достижения цели.

2.5. Метод генерации очередного поколения

Начальное поколение состоит из фиксированного числа случайно сгенерированных автоматов. Все автоматы в поколении имеют одинаковое наперед заданное число состояний. В данной работе число состояний равно десяти.

Для генерации очередного поколения используется классический генетический алгоритм. Предыдущее поколение хранится в массиве $popul[]$. Пусть в поколении N особей.

Опишем используемый алгоритм:

1. Отсортируем $popul[]$ по возрастанию ФП.
2. Произведем ранговый отбор и в качестве элиты возьмем $0,1*N$ особей с наибольшим значением функции приспособленности.
3. Среди не элитных особей сравниваем две случайные и записываем в массив промежуточной популяции ту, у которой ФП больше.
4. Среди особей промежуточной популяции $0,45*N$ раз выбираются случайные две, они скрещиваются, и к детям применяется оператор мутации. Получившиеся особи записываются в $popul[]$ последовательно с первой позиции вместо особей предыдущего поколения. Таким образом, в $popul[]$ окажется новое поколение особей.
5. Далее для каждой особи нового поколения считается её ФП.

3. Результаты работы

В данной работе рассмотрены значения количества особей от 100 до 480. При исследовании каждого из них в поколении было произведено 100 запусков с числом поколений 500. В каждом запуске для количества особей в поколении (100, 150, 200... 400) бралась величина максимальной функции

приспособленности в соответствующем поколении. После получения 100 результатов запусков, данные значения усреднялись. Результаты запусков отражены на рис. 2. Аналогично посчитаны средние значения функции приспособленности, изображенные на рис. 3.

Заметим, что раньше всех достигают цели запуски с количеством особей от 200 до 250. Рассмотрим этот промежуток подробнее с шагом количества особей 10(рис. 4). По графикам, изображенным на рис. 4, можно сделать вывод, что при выборе количества особей равного 230 достаточно всего 73 смены поколений, чтобы робот дошел до цели за 200 шагов.

Можно заметить, что наибольшая эффективность генетического алгоритма в поиске оптимального решения задачи достигается при количестве особей от 300 и выше. Рассмотрим промежуток от 300 до 480 подробнее с шагом количества особей 20(рис. 5). По графикам, изображенным на рис. 5, можно сделать вывод, что при выборе количества особей от 420 до 440 генетический алгоритм работает наиболее эффективно.

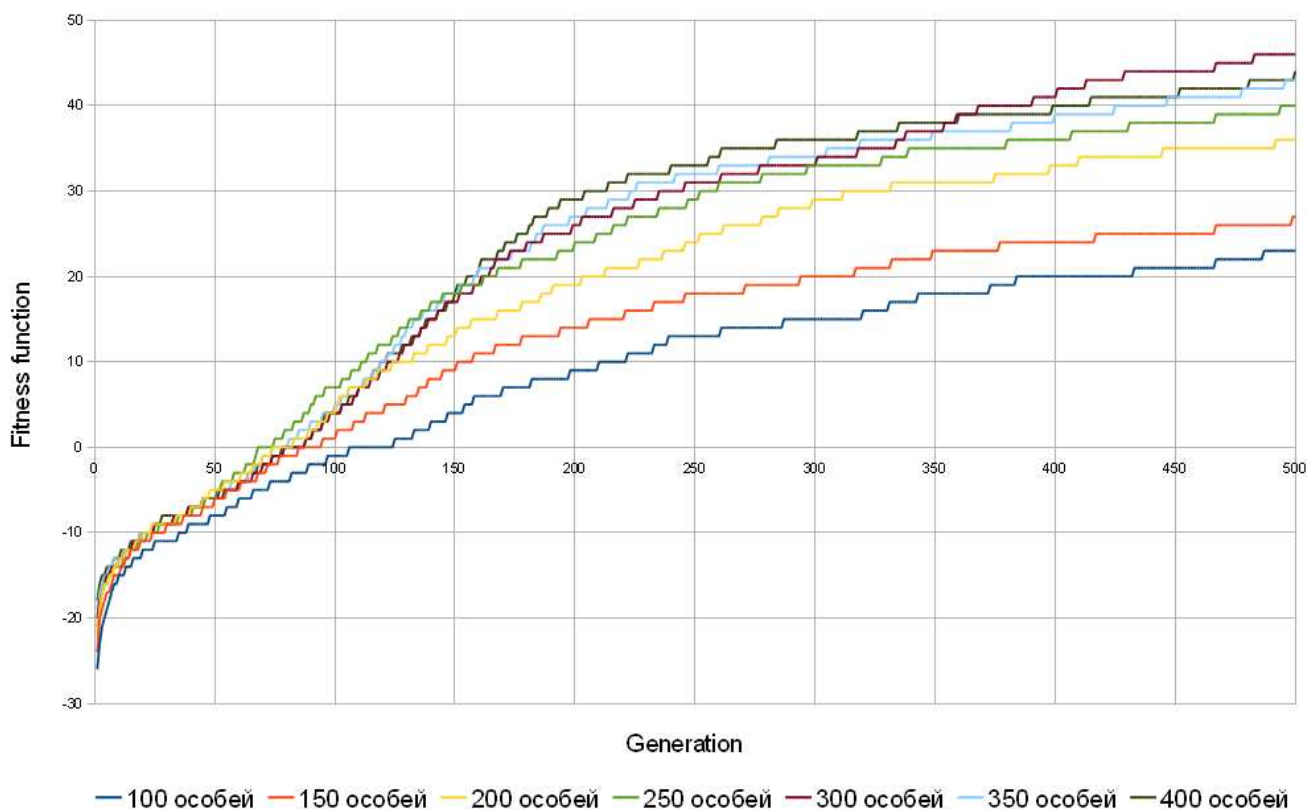


Рис. 2. Графики максимального значения функции приспособленности: 100 – 400 особей

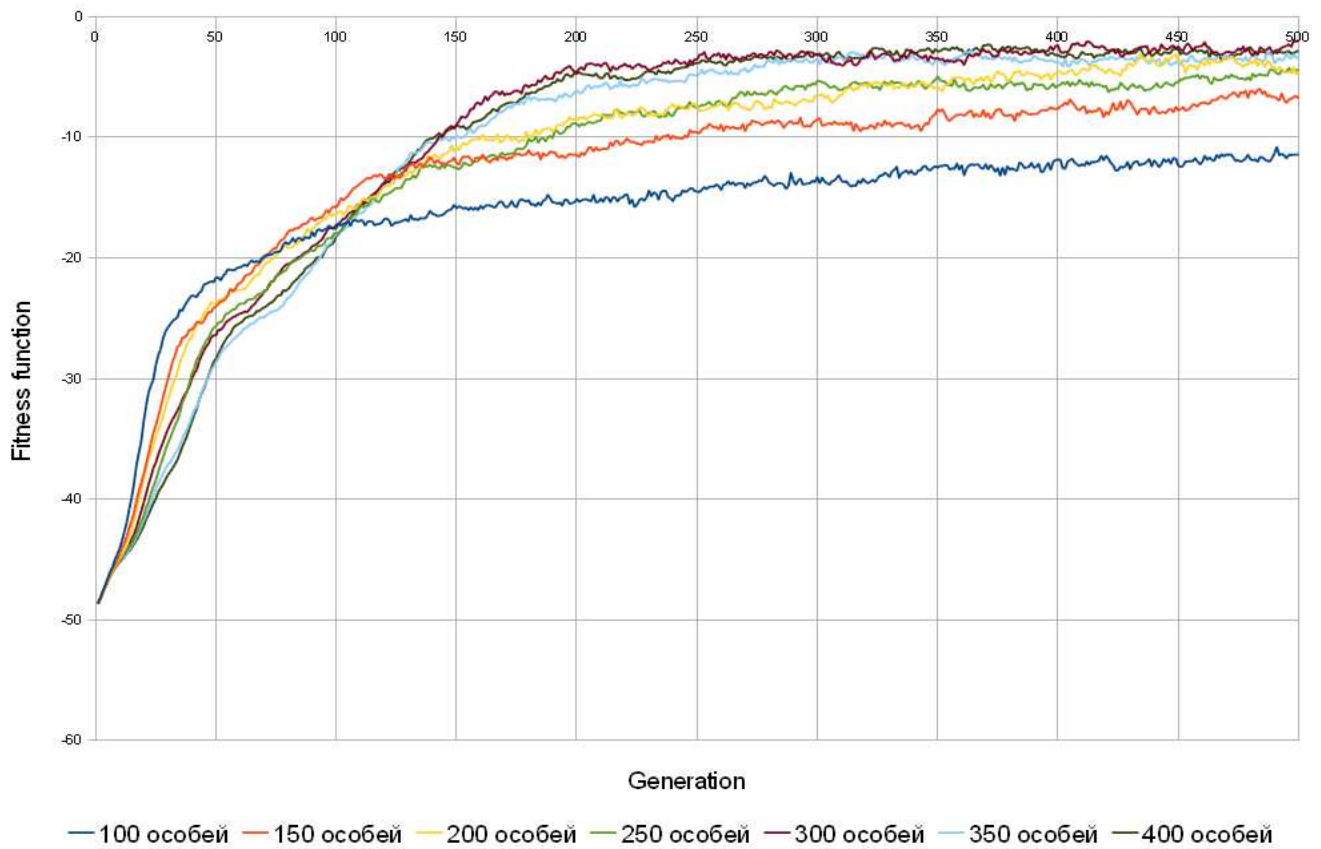


Рис. 3. Графики среднего значения функции приспособленности: 100 – 400 особей

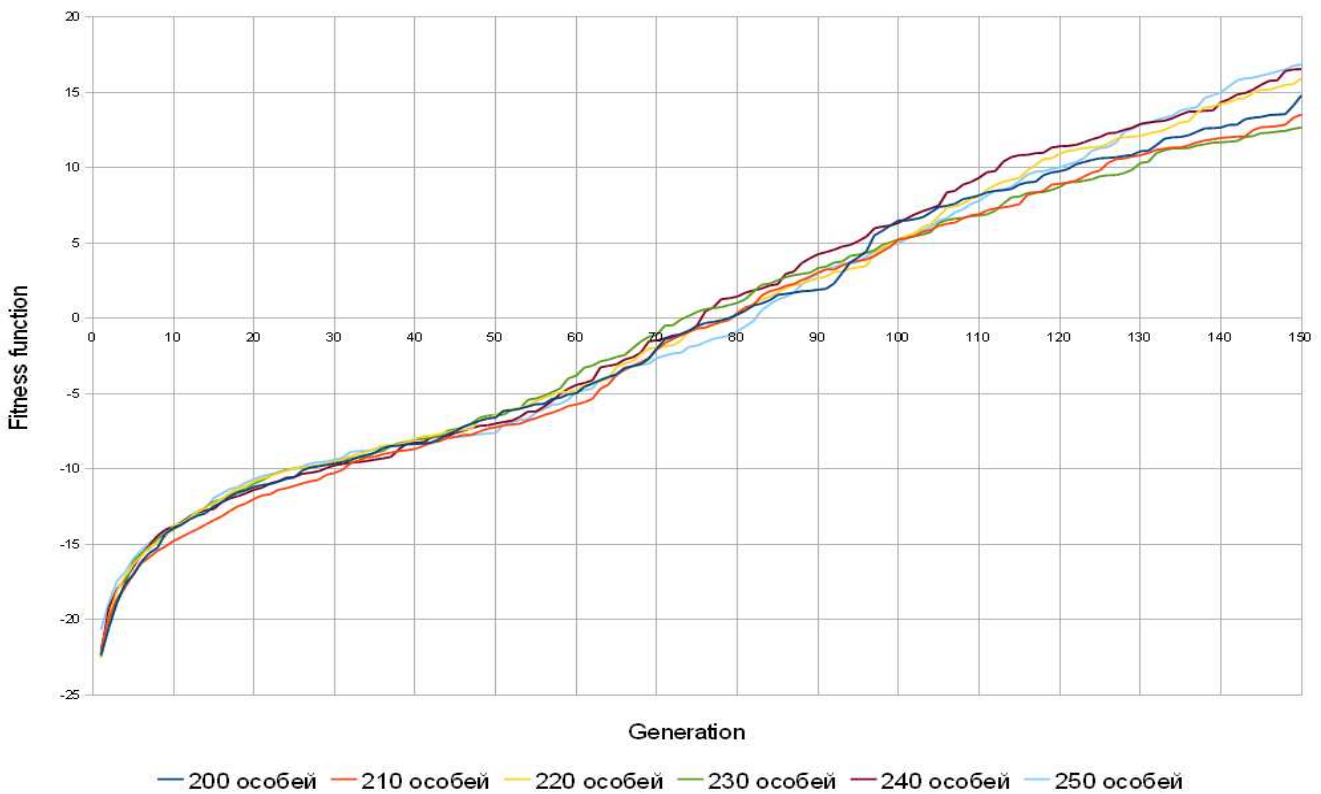


Рис. 4. Графики максимального значения функции приспособленности: 200 – 250 особей

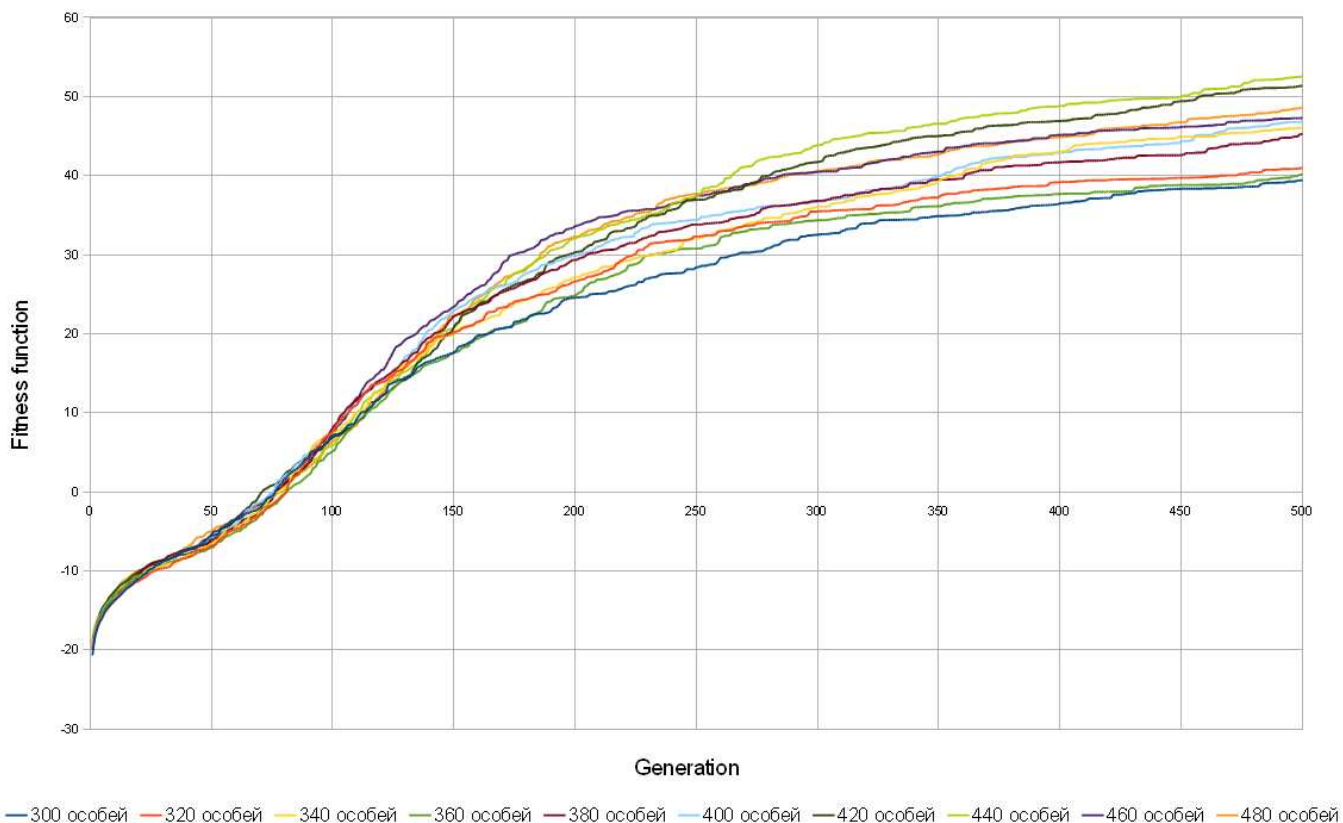


Рис. 5. Графики максимального значения функции приспособленности: 300 – 480 особей

Рис. 5. наглядно демонстрирует, что оптимально алгоритм работает при количестве особей в поколении от 420 до 440. Используем при построении автомата количество особей равное 430. Получим автомат из десяти состояний, который обеспечивает достижение роботом цели за 112 шагов. На приведенной ниже схеме автомата зеленым отмечены переходы при условии, что перед роботом нет препятствия, иначе переход отмечен красным. Буква на переходе показывает действие, которое должен совершить робот: F – идти прямо, L – повернуть налево, R – повернуть направо. Стартовым является состояние под номером десять.

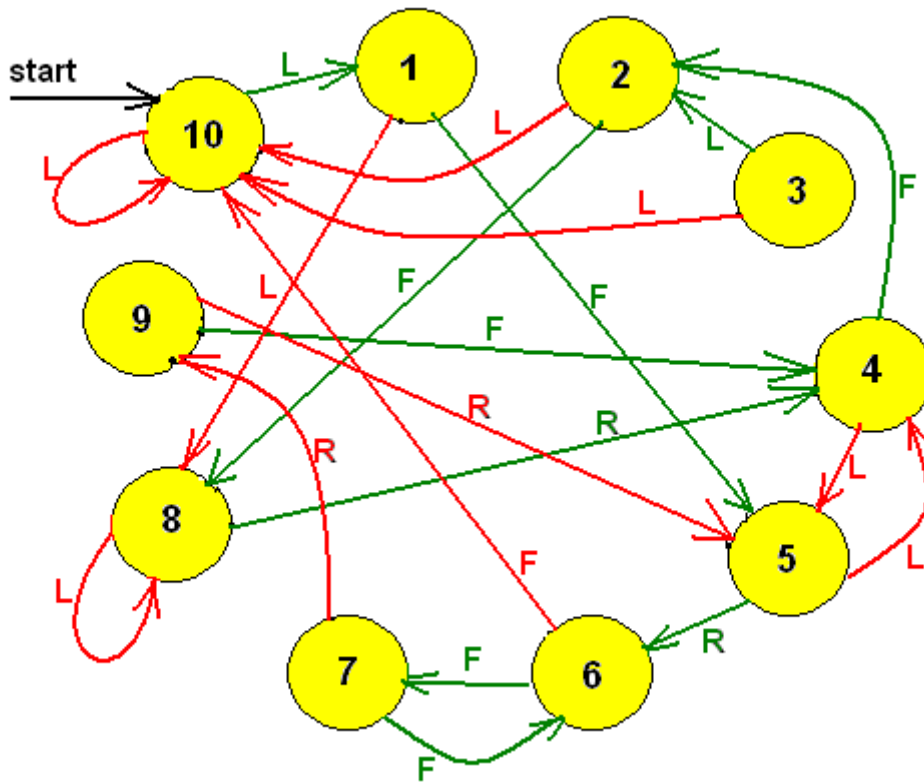


Рис. 6. Схема автомата Мили из десяти состояний для робота, достигающего цель за 112 шагов.

Заключение

В результате выполнения лабораторной работы получено, что при построении конечного детерминированного автомата Мили из десяти состояний с помощью генетического алгоритма для решения задачи о роботе, обходящем препятствия, быстрее всех доходят до цели автоматы, при построении которых количество особей в поколении было равно 230. Наибольшая эффективность алгоритма в поиске автомата, находящего цель за минимальное число шагов, замечена на промежутке от 420 до 440. В качестве оптимального значения числа особей в поколении можно порекомендовать 430.