

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

А. А. Байдаров

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

Вариант №1

Санкт-Петербург
2011 г.

Оглавление

Введение	3
1. Постановка задачи.....	3
1.1. Задача об умном муравье	3
2. Автомат Мили	4
3. Реализация.....	4
3.1. Мутация	4
3.2. Кроссовер	4
3.3. Операторы селекции	5
3.3.1. Пропорциональный отбор	5
3.3.2. Турнирный отбор 1	5
3.3.3. Турнирный отбор 2	5
3.4. Функция приспособленности	6
4. Результаты.....	6
Вывод	9
Источники.....	9

Введение

В лабораторной работе требуется сравнить эффективность работы генетического алгоритма при использовании в качестве оператора селекции пропорционального отбора и турнирного отбора в задаче об умном муравье. При выполнении работы использовался фреймворк *Watchmaker* [1] для работы с генетическими алгоритмами. Весь исходный код, используемый в данной работе, написан на языке *Java*.

1. Постановка задачи

Задача лабораторной работы — сравнить эффективность работы генетического алгоритма при использовании в качестве оператора селекции пропорционального отбора и турнирного отбора, строящего автомат Мили с семью состояниями, который решает задачу об умном муравье.

1.1. Задача об умном муравье

Дано поле размером 32×32 клетки, расположенное на поверхности тора (рис. 1). В некоторых клетках (обозначены на рис.1 черным цветом) находится еда. Она расположена вдоль некоторой ломаной, но не во всех ее клетках. Клетки ломаной, в которых нет еды, обозначены серым цветом. Белые клетки не содержат еду и не принадлежат ломаной. Всего на поле 89 клеток с едой.

В клетке, помеченной меткой *Start*, в начале игры находится муравей. Он занимает одну клетку и смотрит в одном из четырех направлений (север, юг, запад, восток). В начале игры муравей смотрит на восток.

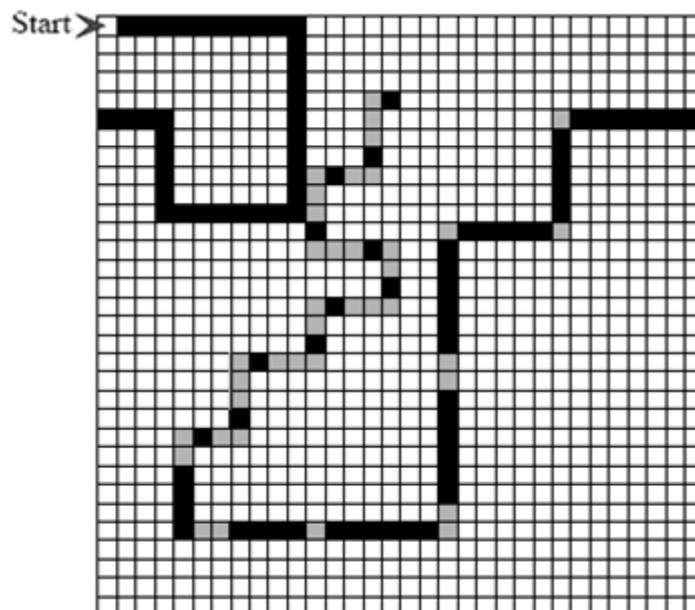


Рис. 1. Игровое поле

Муравей умеет определять, находится ли непосредственно перед ним еда. За один ход муравей может совершить одно из четырех действий:

- повернуть налево (L);
- повернуть направо (R);
- сделать шаг вперед и, если в новой клетке есть еда, съесть ее (M);
- ничего не делать (N).

Игра длится 200 ходов, на каждом из которых муравей совершает одно из четырех действий. По истечении 200 ходов подсчитывается количество еды, съеденной муравьем.

Цель игры — создать муравья, который за 200 ходов съест как можно больше еды (желательно, все 89 единиц). При одинаковом количестве съеденных яблок, лучшим считается муравей, съевший яблоки за меньшее число ходов.

2. Автомат Мили

Автомат Мили — пятерка $A = \langle S, q_0 \in S, X, Y, \delta \rangle$, где

- S — множество состояний;
- q_0 — стартовое состояние;
- X — множество входных воздействий;
- Y — множество выходных воздействий;
- $\delta: S \times X \rightarrow S \times Y$ — функция переходов.

3. Реализация

Для реализации генетического алгоритма использован фреймворк *Watchmaker* [1], были написаны операторы мутации, кроссовера и селекции, а также класс вычисляющий функцию приспособленности. Особь кодируется в виде автомата. Каждое состояние содержит 4 параметра: *food* — состояние автомата, в которое ведет переход, если перед муравьем есть еда, *foodAction* — действие, совершаемое при этом, *noFood* — состояние автомата, в которое ведет переход, если перед муравьем нет еды, *noFoodAction* — действие, совершаемое при этом.

3.1. Мутация

Оператор мутации реализован следующим образом. С некоторой вероятностью p изменяем начальное состояние автомата. Далее для каждого состояния и для каждого перехода из этого состояния делаем следующее:

- с вероятностью p изменяем действие на переходе;
- с вероятностью p изменяем состояние, в который ведет переход.

3.2. Кроссовер

В качестве оператора скрещивания используется двухточечный кроссовер (рис.2). Для пары выбранных родителей происходит следующее: случайным образом выбираются номера

состояний A и B (будем считать, что $A \leq B$), родители обмениваются состояниями с номерами от A до B. Получившиеся при этом автоматы и будут потомками.

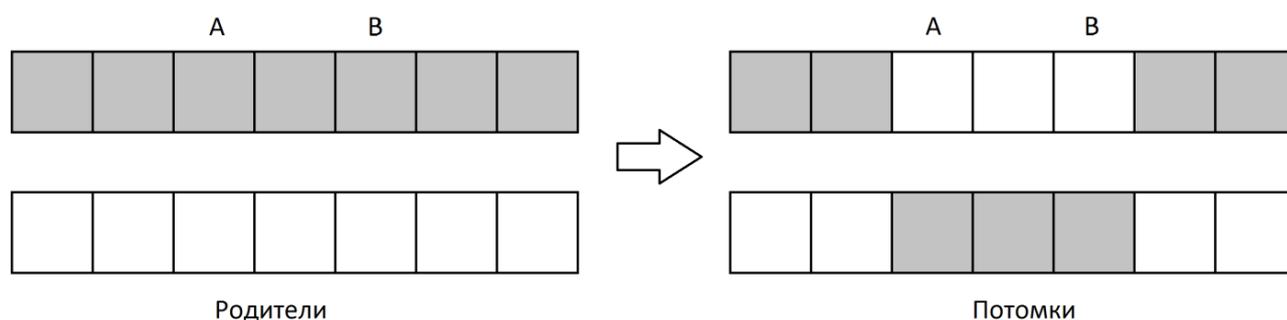


Рис. 2. Двухточечный кроссовер

3.3. Операторы селекции

Пусть нам дана популяция, содержащая n особей. Требуется создать промежуточную популяцию в m особей.

3.3.1. Пропорциональный отбор

Пропорциональный отбор заключается в том, что для каждой особи считается отношение ее приспособленности к средней приспособленности популяции $P_i = \frac{f_i}{f_{cp}}$. Далее происходит отбор m особей согласно этим значениям. В данной работе использовался метод рулетки. Колесо рулетки содержит по одному сектору для каждого члена популяции, причем размер i -ого сектора пропорционален соответствующей величине P_i . Далее происходит m запусков рулетки, и на каждом шаге в промежуточную популяцию берется особь, соответствующая выпавшему сектору.

3.3.2. Турнирный отбор 1

Турнирный отбор происходит следующим образом: случайным образом выбирается k особей и лучшая из них записывается в промежуточную популяцию (таким образом, между выбранными особями проводится турнир). Эта операция повторяется m раз.

3.3.3. Турнирный отбор 2

Существует другой вариант турнирного отбора. В нем выбирается небольшое число k , затем случайным образом выбираются 2^k особи, и между ними происходит турнир по «олимпийской» системе в k раундов. В каждом раунде особи разбиваются на пары, из каждой пары в следующий раунд особь выходит с вероятностью, пропорциональной ее функции приспособленности. Операция повторяется m раз.

3.4. Функция приспособленности

Определим функцию приспособленности следующим образом:

$$A + \frac{200 - T}{200},$$

где A — суммарное количество еды, съеденной муравьем за игру, T — номер хода, на котором муравей съедает последнюю единицу еды.

4. Результаты

Для каждого метода отбора был выбран некоторый диапазон значений коэффициента k , и для каждого k из этого диапазона было произведено 100 запусков генетического алгоритма с использованием выбранного метода отбора и выбранного значения k . В каждом запуске бралось наибольшее значение функции приспособленности в соответствующем поколении. После получения результатов запусков, значения усреднялись. Графики зависимости значения функции приспособленности от номера поколения для каждого метода селекции представлены ниже.



Рис. 3. График зависимости среднего значения функции приспособленности от номера поколения при использовании пропорционального отбора

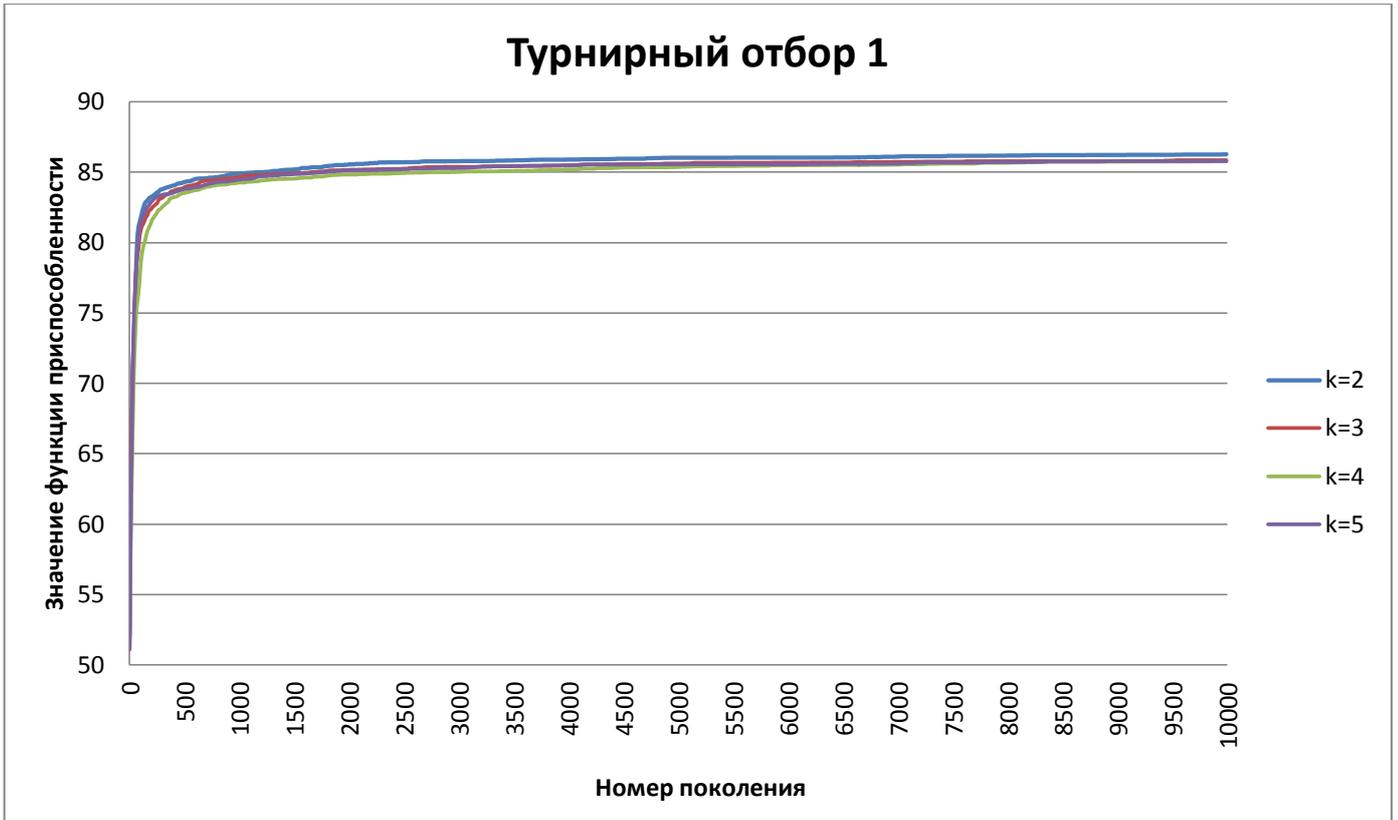


Рис. 4. График зависимости среднего значения функции приспособленности от номера поколения при использовании турнирного отбора 1

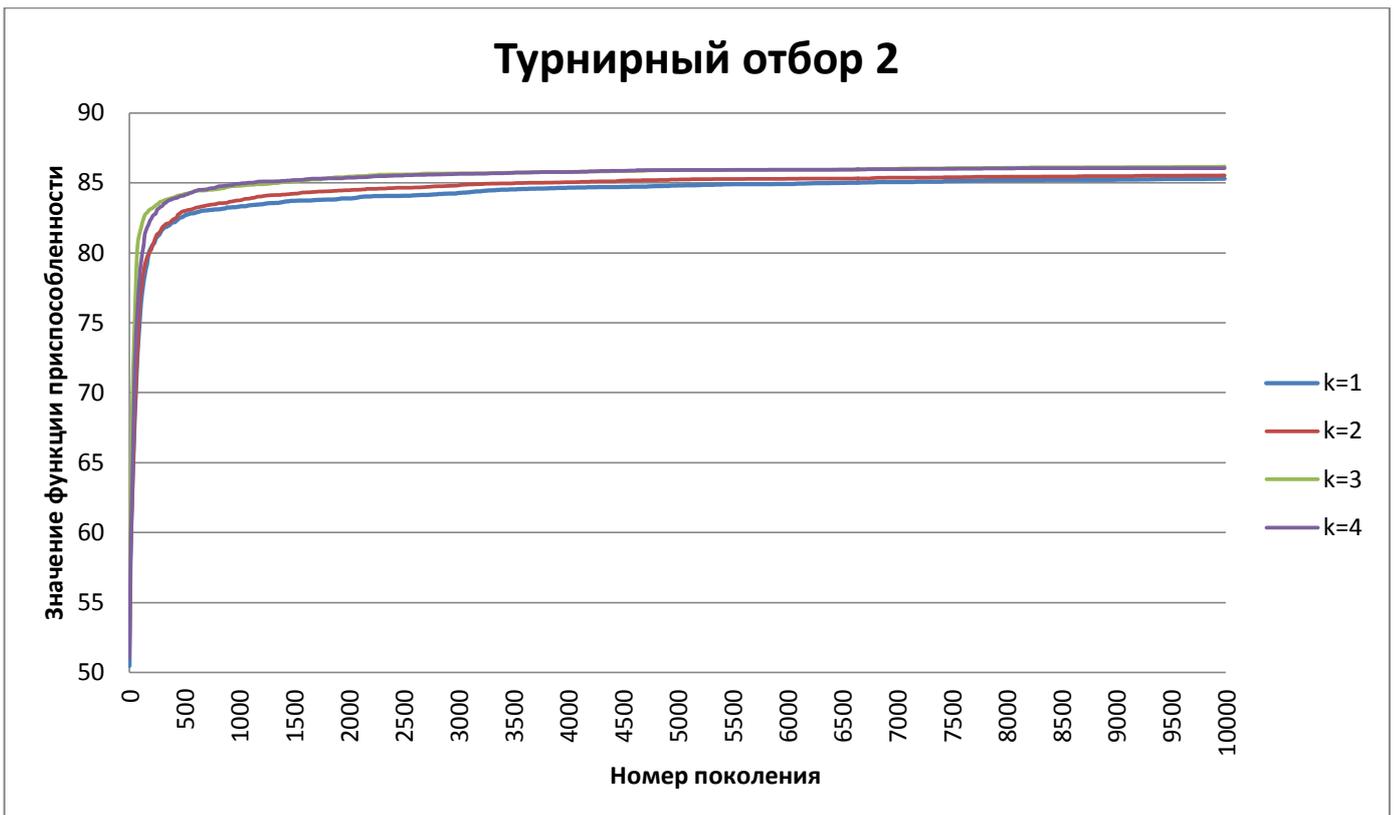


Рис. 5. График зависимости среднего значения функции приспособленности от номера поколения при использовании турнирного отбора 2

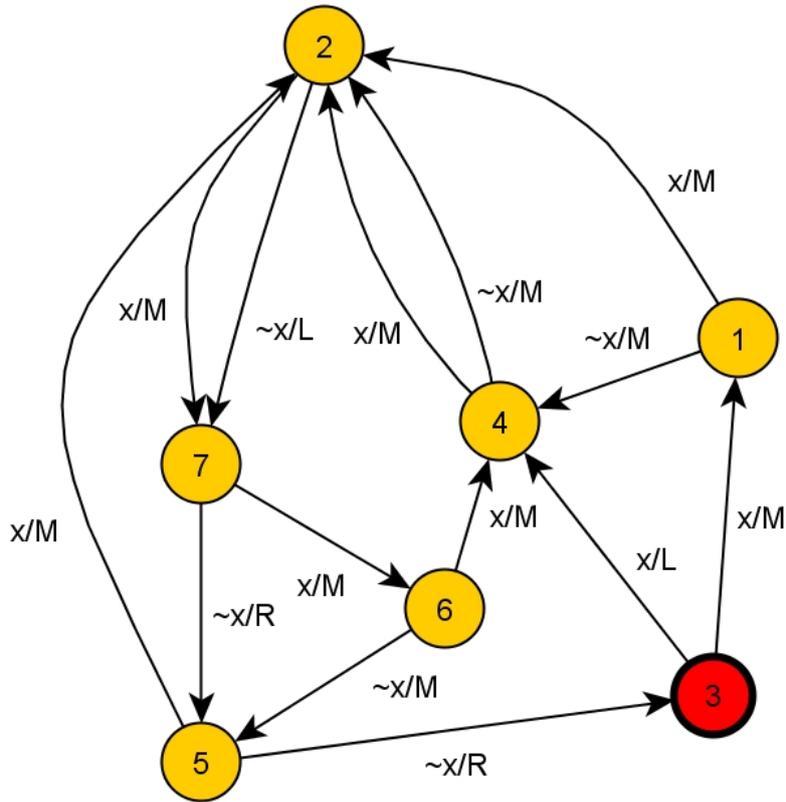


Рис. 6. Автомат, полученный при пропорциональном отборе. Съедает 86 яблок за 191 шаг.

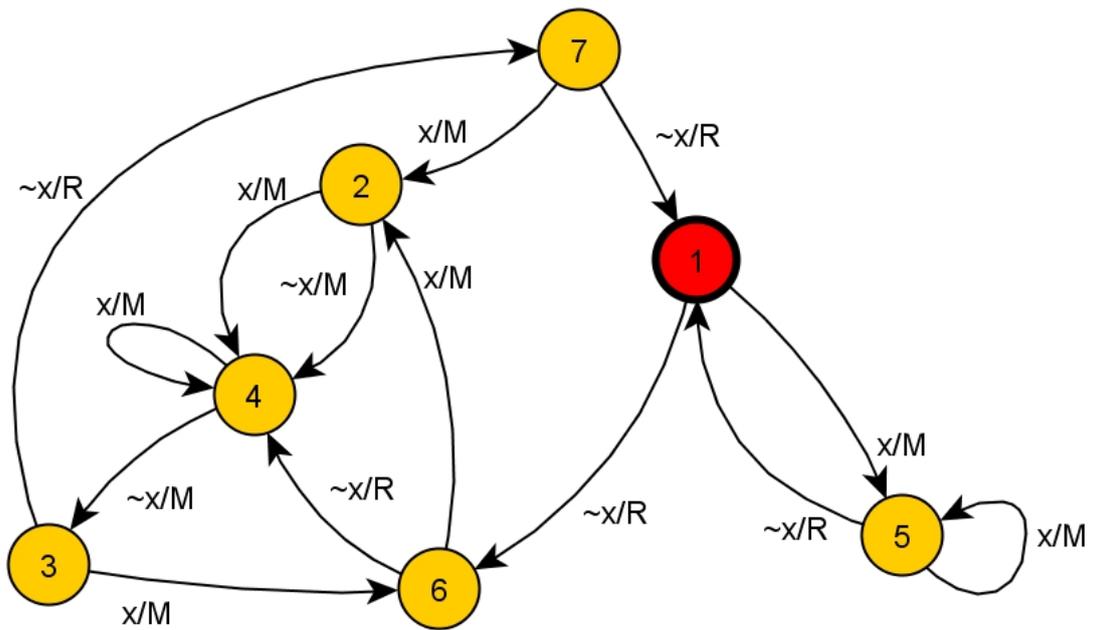


Рис. 7. Автомат, полученный при турнирном отборе 1 ($k=2$). Съедает 88 яблок за 194 шага.

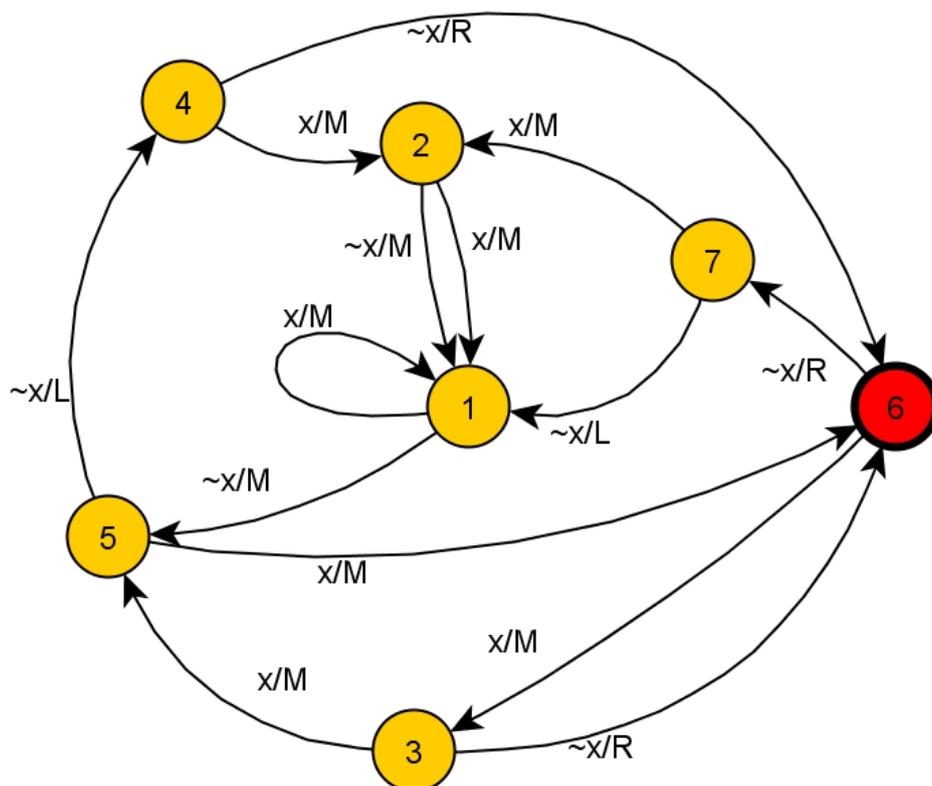


Рис. 8. Автомат, полученный при турнирном отборе 2 ($k=3$) . Съедает 88 яблок за 198 шагов.

Вывод

Результаты лабораторной работы показали, что эффективное построение автомата Мили с семью состояниями, решающего задачу об «Умном муравье», наблюдается при использовании турнирного отбора в качестве оператора селекции, причем в случае первого варианта турнирного отбора эффективность чуть выше при значении $k=2$, а в случае второго варианта — при $k=3$ и $k=4$. Лучшие автоматы (рис. 6-8), соответствующие различным методам отбора, которые были получены в процессе выполнения, также показали, что турнирный отбор эффективней использовать в качестве оператора селекции, чем пропорциональный отбор.

Источники

1. The Watchmaker Framework for Evolutionary Computation. <http://watchmaker.uncommons.org/>.
2. Поликарпова Н.И., Шалыто А.А. Автоматное программирование. <http://is.ifmo.ru/books/ book.pdf>
3. Бедный Ю. Д., Шалыто А. А. Применение генетических алгоритмов для построения автоматов в задаче «Умный муравей». <http://is.ifmo.ru/works/ ant.pdf>