

Национальный исследовательский университет информационных
технологий, механики и оптики
Факультет информационных технологий и программирования
Кафедра компьютерных технологий

А.В. Лебедева

**Отчет по лабораторной работе
«Автоматное
программирование и алгоритмы оптимизации»**

Вариант № 17

Санкт-Петербург
2011

Оглавление

Введение	3
1. Постановка задачи	3
1.1 Задача об умном муравье-3	3
2. Автомат Мили	3
3. Алгоритм	3
3.1 Представление особи	3
3.2 Генерация полей	4
3.3 Оценка приспособленности	4
3.4 Мутация	4
3.5 Отжиг Коши	4
3.6 Сверхбыстрый отжиг	4
4. Результаты	5
5. Источники	7

Введение

Цель работы – сравнить эффективность работы отжига Коши и сверхбыстрого отжига на примере задачи об умном муравье-3, построив конечный автомат Мили, решающий ее. Автомат задается с использованием деревьев решений.

Для выполнения работы использовалась виртуальная лаборатория «3genetic». Все исходные коды в данной работе написаны на языке программирования Java.

1. Постановка задачи

Задача данной лабораторной работы – сравнить эффективность работы отжига Коши и сверхбыстрого отжига. Критерий оценки алгоритма заключается в том, что автомат, имея фиксированное число состояний, должен съесть как можно больше еды на поле за ограниченное число шагов.

1.1 Задача об умном муравье-3

В задаче об умном муравье-3 рассматривается поле, располагающееся на поверхности тора и имеющее размер 32×32 клетки. Каждая клетка поля с некоторой, заранее определенной, вероятностью содержит яблоко. Муравей видит восемь клеток перед собой и может выполнять одно из следующих действий:

- повернуть направо;
- повернуть налево;
- сделать шаг вперед, и если в новой клетке есть яблоко, то съесть его.

Максимальное число шагов – 200. Цель задачи – создать муравья, который управляется автоматом с фиксированным числом состояний и съедает максимальное число яблок.

2. Автомат Мили

Автомат Мили – пятерка $A = \langle S, q_0, X, Y, \delta \rangle$, 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. Алгоритм

3.1 Представление особи

Особью генетического алгоритма для решения данной задачи является конечный автомат Мили – детерминированный конечный автомат, каждому переходу которого сопоставлено некоторое действие (в данной задаче это либо шаг вперед, либо поворот вправо или влево).

Входным алфавитом этого автомата являются значения восьми логических переменных, обозначающих наличие или отсутствие еды в каждой из видимых муравьем клеток. Состояния в автомате хранятся в виде дерева решений. В вершинах деревьев находятся входные переменные, в листьях дерева хранятся номера состояний, в которые осуществляется переход, и действие, совершаемое при этом муравьем. Число состояний автомата является настраиваемым параметром алгоритма. Все генерируемые алгоритмом автоматы имеют одинаковое число состояний, не изменяющееся во время выполнения алгоритма.

3.2 Генерация полей

Поля 32×32 генерируются случайным образом. Клетка содержит яблоко с вероятностью 0,05.

3.3 Оценка приспособленности

Автомат обучается на нескольких (в данном случае на трех) полях. Для каждого из полей считается величина равная разности яблок, съеденных муравьем, управляемым автоматом-особью, и отношения номера шага, на котором съедено последнее яблоко, к максимально возможному числу шагов (в данном случае к двумстам). Функцией приспособленности является сумма таких величин по всем полям, на которых обучался автомат.

3.4 Мутация

Начиная с корня дерева (сам корень не мутирует), к каждому узлу применяется следующая операция: с вероятностью, полученной из используемого распределения, узел и его поддеревья заменяются на случайно сгенерированные, иначе с равными вероятностями данная операция применяется к левому или правому поддереву.

3.5 Отжиг Коши

Для реализации отжига Коши используется моделирование, изложенное в лекциях А. С. Лопатина [1].

Для моделирования убывания температуры использовалась формула:

$$T(k) = \frac{T_0}{k^{1/D}},$$

где D – число состояний автомата, k – номер шага.

Распределение Коши имеет величина $T(k) \operatorname{tg}(\pi\alpha - \frac{\pi}{2})$, где α – равномерно распределенная величина на $[0; 1]$. При выходе значений случайной величины за границы интервала $[0; 1]$ эти значения генерируются заново.

3.6 Сверхбыстрый отжиг

Для реализации сверхбыстрого отжига также используется моделирование, изложенное в лекциях А. С. Лопатина [1].

Температура является вектором размерности D и по каждой из координат может отличаться. Для моделирования убывания температуры использовалась формула:

$$T_i(k) = T_i(k-1) \cdot e^{-c_i},$$

где $c_i = m_i \cdot e^{-\frac{n_i}{D}}$ ($m_i, n_i \geq 0$ – регулируемые параметры).

Требуемое распределение имеет величина

$$\operatorname{sgn}\left(\alpha_i - \frac{1}{2}\right) \cdot T_i \cdot \left(1 + \frac{1}{T_i}\right)^{(2\alpha_i - 1)} - 1,$$

где α_i – независимые случайные величины, распределенные равномерно на интервале $[0; 1]$. При выходе значений случайной величины за границы интервала $[0; 1]$ эти значения генерируются заново.

4. Результаты

Рассматривалась эффективность работы отжигов для различных начальных температур (50 запусков для каждой). Ниже (на рис. 1–5) представлены графики работы алгоритмов для каждой из начальных температур (голубой – сверхбыстрый отжиг, зеленый – отжиг Коши). Наилучшие результаты достигаются при начальных температурах, лежащих в диапазоне $[30; 40]$. В большинстве случаев отжиг Коши оказывается наиболее эффективным. На рис. 6 представлен график зависимости температуры от номера поколения.

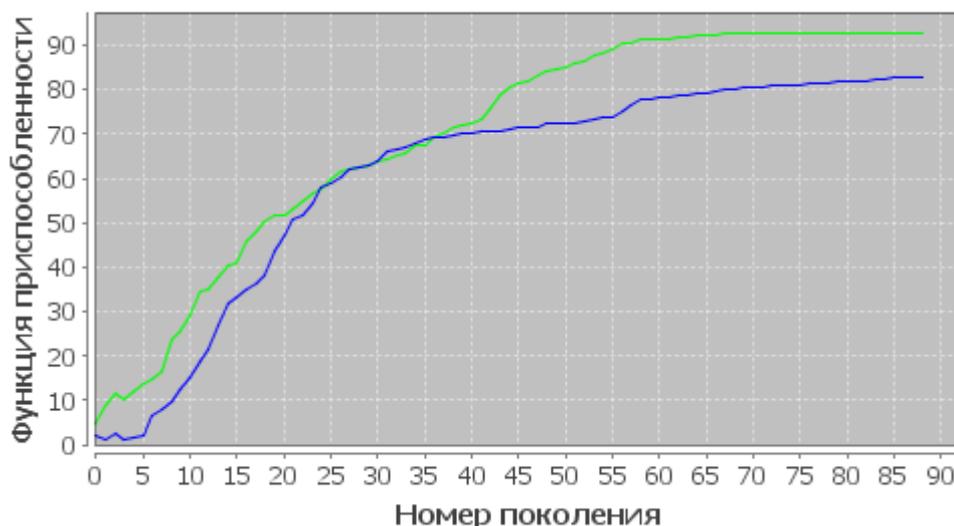


Рис. 1 – $T_0 = 20$

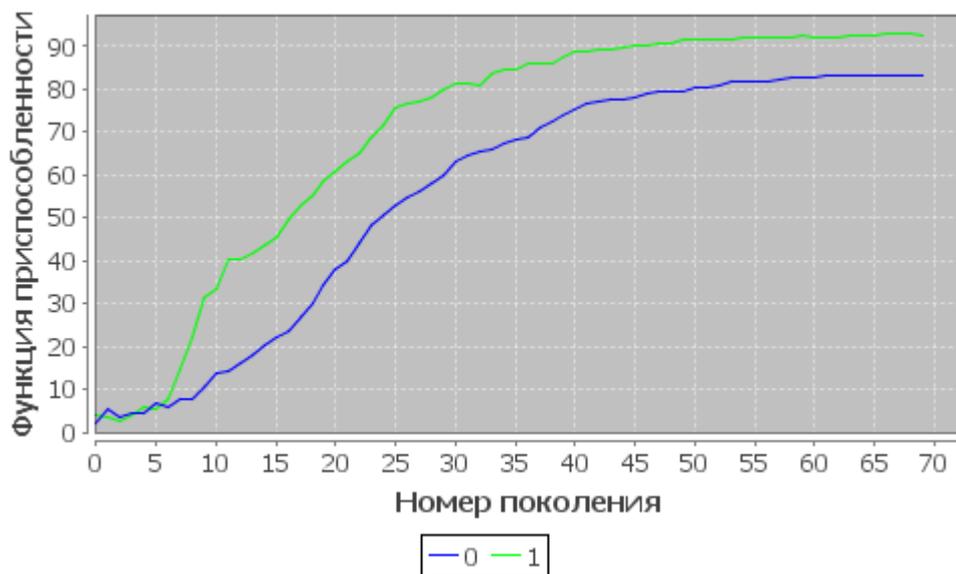


Рис. 2 – $T_0 = 25$

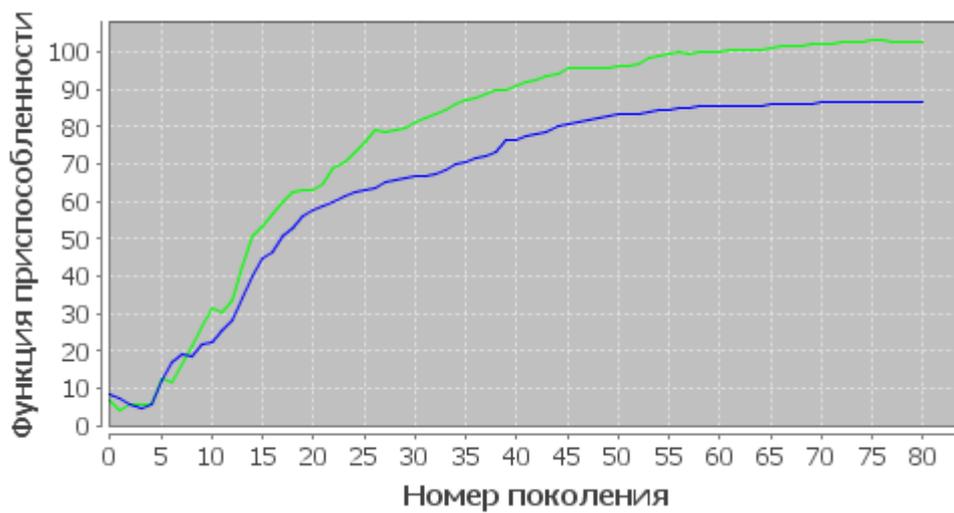


Рис. 3 – $T_0 = 30$

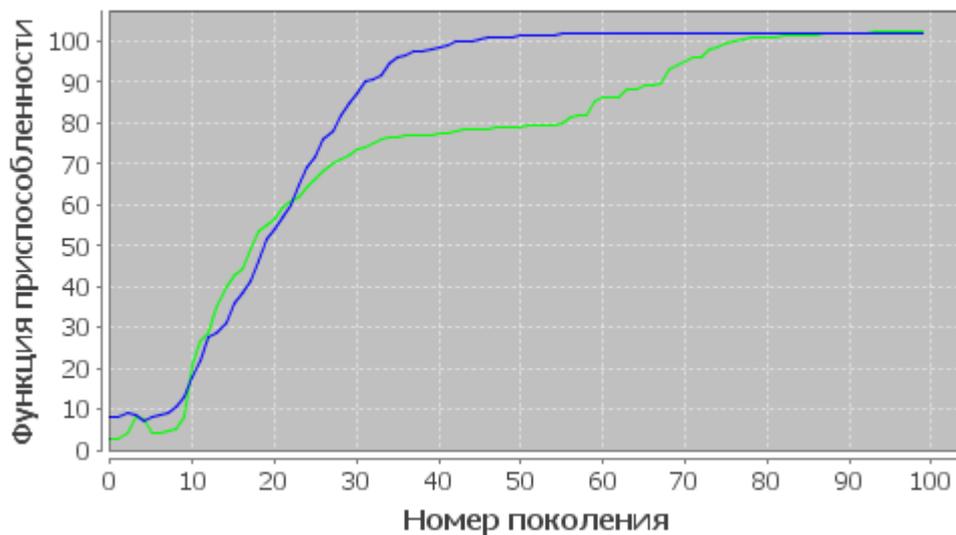


Рис. 4 – $T_0 = 35$

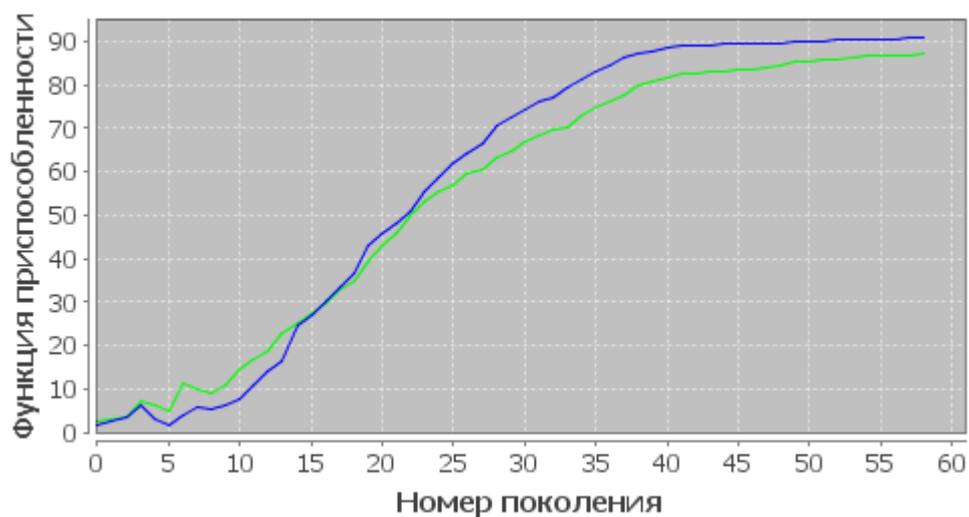


Рис. 5 – $T_0 = 40$

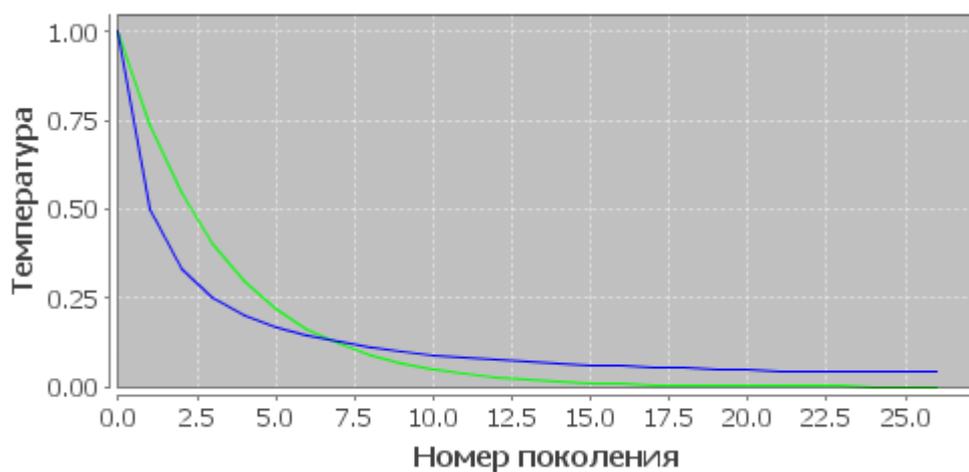


Рис. 6 – График зависимости температуры от номера поколения

5. Источники

1. Метод имитации отжига. Конспект лекций А. Лопатина.
<http://rain.ifmo.ru/~buzdalov/lab-2011/books/annealing.pdf>