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

А. М. Глухов

Отчет по лабораторной работе «Использование генетических алгоритмов для построения управляющих автоматов»

Вариант №8

Принял: Ульянцев В. И.

Санкт-Петербург
2011

Оглавление

Введение

Цель лабораторной работы — исследовать эффективность работы Больцмановского отжига при задании автомата полными и сокращенными таблицами. В качестве примера рассматривается построение конечного автомата Мура, решающего задачу об «Умном муравье – 3».

При выполнении лабораторной работы использовалась программа «Виртуальная лаборатория GlOpt» [1], разработанная студентами кафедры «Компьютерные технологии» НИУ ИТМО, которая позволяет реализовать генетические алгоритмы и особи для них в виде плагинов.

1. Постановка задачи

Целью настоящей работы является исследование эффективности работы Больцмановского отжига при задании автомата Мура полными и сокращенными таблицами на примере задачи об «Умном муравье – 3».

1.1. Задача об «Умном муравье – 3»

В задаче об «Умном муравье – 3» рассматривается поле, располагающееся на поверхности тора и имеющее размер 32 на 32 клетки (рис. 1). Каждая клетка поля с некоторой, заранее определенной, вероятностью содержит яблоко (в данной работе вероятность равна 0,11).

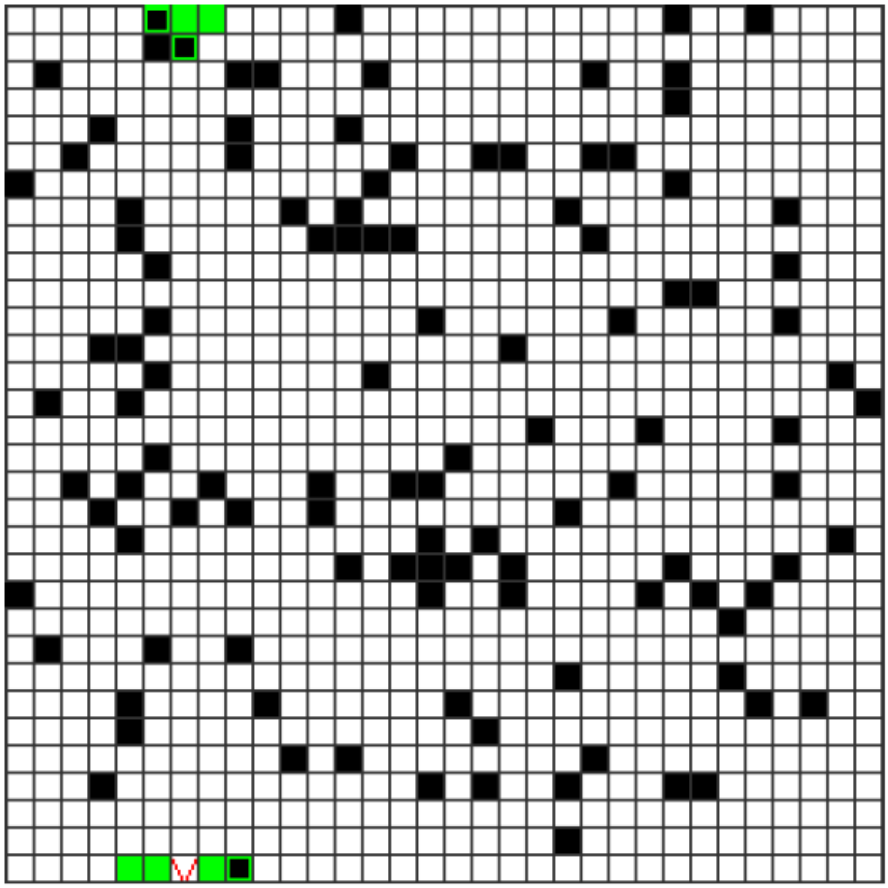

Рис. 1 — Игровое поле

Муравей видит перед собой восемь клеток (рис. 2) и может выполнять одно из следующих действий:

1. повернуть налево;
2. повернуть направо;
3. сделать шаг вперед, и, если в новой клетке есть яблоко, съесть его;
4. ничего не делать.

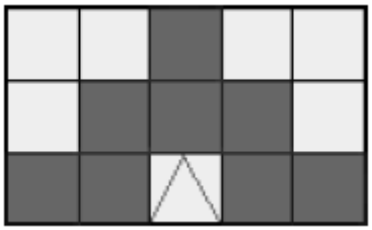

Рис.2 — Видимая область муравья

Число действий, которые позволено совершить муравью по условию задачи, равняется двумстам.

Решением задачи является автомат с фиксированным числом состояний, с помощью которого муравей сможет съесть максимальное число яблок (рис. 3).

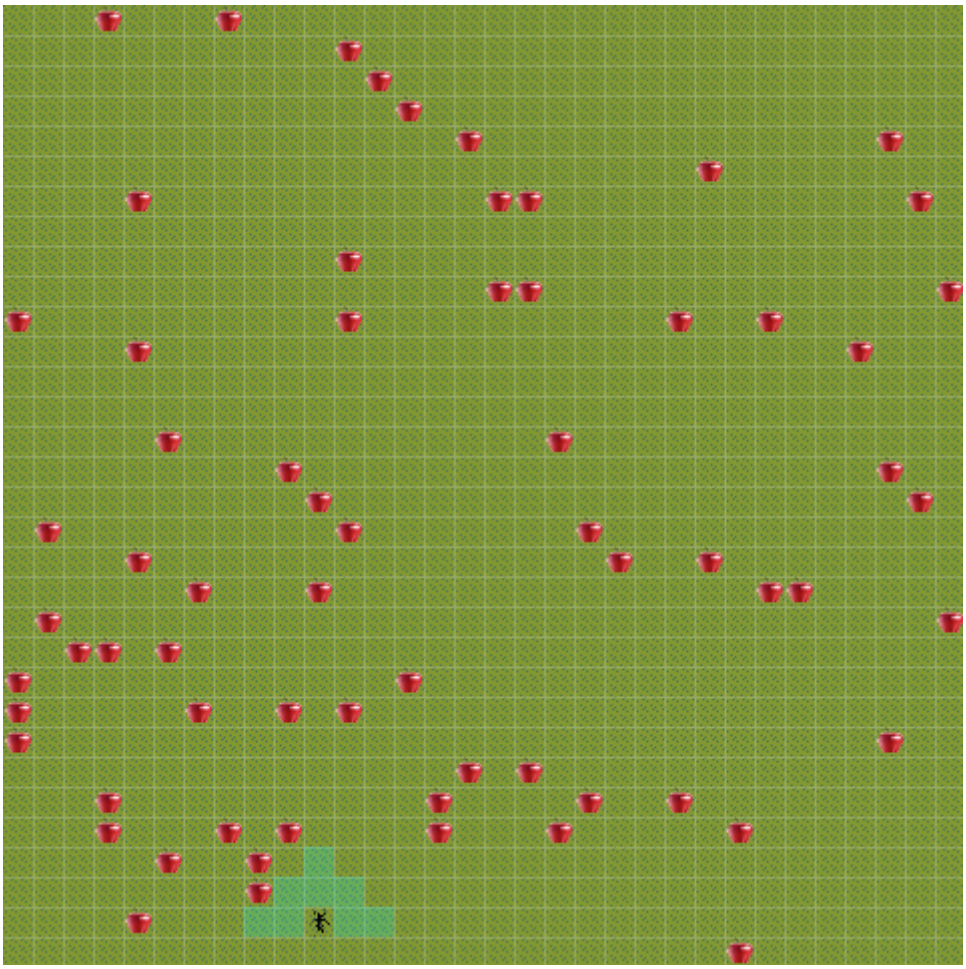

Рис. 3 — Визуализатор построенного автомата
в процессе эмуляции движения муравья по полю

2. Автомат Мура

В этом разделе описывается автомат Мура [2] и его представление в программе.

2.1. Описание автомата Мура

В данной работе автомат для управления муравьем является автоматом Мура, то есть совокупностью пяти объектов $A = \{S, X, Y, \delta, \mu\}$, где S — множество вершин, X — множество входных воздействий, Y — множество выходных воздействий, δ — отображение $S \times X \to S$, μ — отображение $S \to Y$.

Автомат Мура можно изобразить графически (рис. 4).

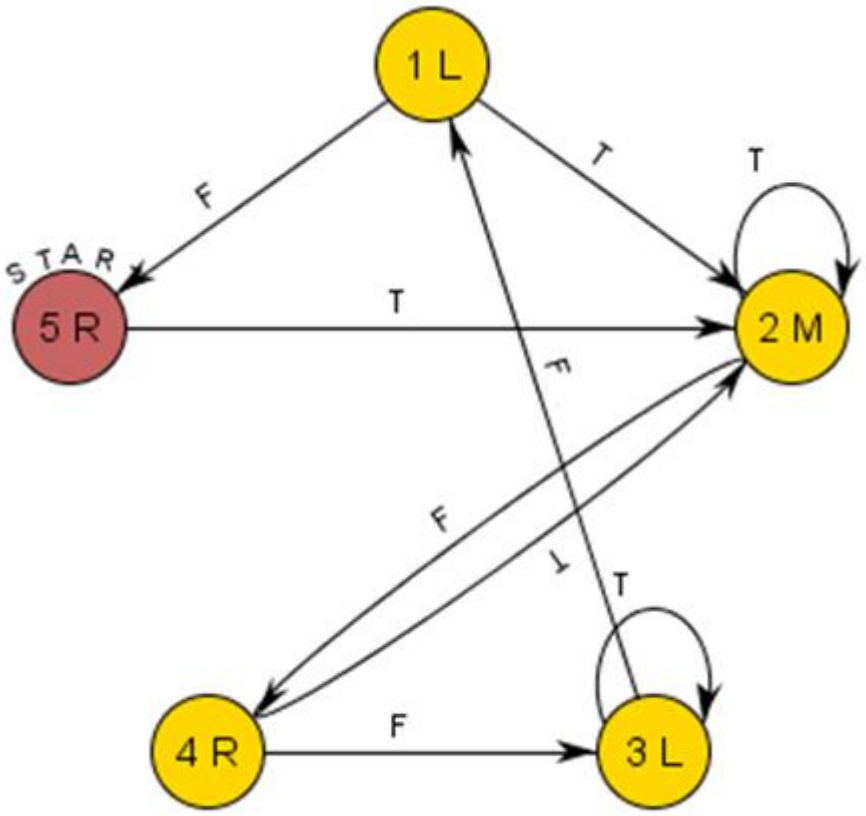

Рис. 4 — Графическое изображение автомата Мура

2.2 Представление автомата

Автомат представлялся тремя массивами.

1. Массив действий *actions*, хранящий действия муравья (*Left*, *Right*, *Move*);
2. Массив переходов *transitions*, который представляет собой матрицу переходов по всем входным воздействиям из состояния и хранит номер состояния, в которое переходим;
3. Битовая маска сокращенной таблицы *mask*, заполненная нулями и единицами. В случае полных таблиц вся маска заполнена единицами.

3.Задание автомата таблицами

Автомат можно задать с помощью полных таблиц, сокращенных таблиц и деревьев решений. В данном разделе описываются первые два способа, потому что именно они используются в работе. Ознакомиться с представлением автомата с помощью деревьев решений можно в [3].

3.1 Полные таблицы

Способ описания автомата, в котором по всем входным воздействиям, состоящим из наличия еды в каждой из восьми видимых муравьем клеток, задается переход в какое-либо состояние, называется полной таблицей переходов. Вид таблицы представлен ниже (рис. 5).

Для каждого из входных воздействий, которых в нашей задаче $2^8 = 256$, и каждого состояния задан переход в некоторое состояние автомата, описанный номером этого состояния. Состояния характеризуются своим номером и действием, совершаемым в них (*L* — поворот влево, *R* — поворот вправо, *M* — движение вперед).

Входные воздействия								Состояния		
x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	(0; L)	(1; R)	(2; M)
0	0	0	0	0	0	0	0	1	2	0
0	0	0	0	0	0	0	1	1	0	2
0	0	0	0	0	0	1	0	2	1	0
0	0	0	0	0	0	1	1	0	1	2
…										
1	1	1	1	1	1	1	1	2	1	0

Рис. 5 — Полная таблица переходов

3.2 Сокращенные таблицы

В реальных задачах число переходов в управляющих автоматах, сформированных вручную, не возрастает экспоненциально с увеличением числа предикатов объекта управления [4]. Причина состоит, по-видимому, в том, что в большинстве задач предикаты имеют «локальную природу» по отношению к управляющим состояниям. В каждом состоянии значимым является лишь определенный, небольшой поднабор предикатов, остальные же не влияют на значение управляющей функции.

Для реализации сокращенных таблиц введем для каждого состояния маску из восьми битов, показывающую, какие из видимых клеток поля являются значимыми для конкретного состояния.

Число значимых клеток у всех состояний равно некоторой задаваемой константе *r*. К примеру, если $r = 4$ (рис. 6), то число всех различных входных воздействий для автомата будет равняться $2^r = 16$.

x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7
0	1	1	0	1	0	1	0

Рис. 6 — Пример маски из четырех значимых битов

Для маски определим операцию мутации (рис. 7). Два случайно выбранных соседних элемента маски меняются местами. Остальные элементы остаются без изменений.

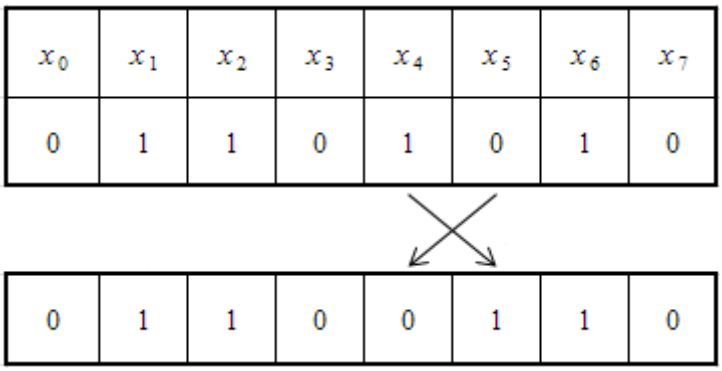

Рис. 7 — Пример мутации

4. Метод имитации отжига

В данном разделе описывается алгоритм имитации отжига в общем виде, а также конкретная реализация Больцмановского отжига.

4.1 Описание метода имитации отжига

Метод отжига – это техника оптимизации, использующая случайный поиск на основе аналогии с процессом образования веществом кристаллической структуры с минимальной энергией при охлаждении [5].

Метод отжига служит для поиска глобального минимума некоторой функции $f(x)$, заданной из некоторого пространства S, дискретного или непрерывного. Элементы множества S представляют собой состояния воображаемой физической системы («энергетические уровни»), а значение функции f в этих точках используется как энергия системы $E = f(x)$ (в нашем случае $f(x)$ является функцией приспособленности).

Предполагается, что процесс протекает при постепенно понижающейся температуре. Находясь в состоянии при температуре T, следующее состояние системы выбирается в соответствии с заданным распределением вероятностей $Q(x; T)$, которое и задает новый случайный элемент $x^l = G(x; T)$. После генерации x^l система с вероятностью $h(\Delta E; T)$ переходит к следующему шагу в состояние x^l. Если переход не произошел, процесс генерации x^l повторяется. Здесь ΔE обозначает приращение функции энергии $f(x^l) - f(x)$. Величина $h(\Delta E; T)$ называется вероятностью принятия нового состояния.

Итак, конкретная схема метода отжига задается следующими параметрами:

1. выбором закона изменения температуры $T(k)$, где k — номер шага;
2. выбором вероятностного распределения $Q(x; T)$;
3. выбором функции вероятности принятия $h(\Delta E; T)$.

4.2 Алгоритм имитации отжига в общем виде

1. Случайным образом выбирается начальная точка $x = x_0$; $x_0 \in S$. Текущее значение энергии E устанавливается в значение $f(x_0)$.

2. k-я итерация основного цикла состоит из следующих шагов:

(a) Сравнить энергию системы E в состоянии x с найденным на текущий момент глобальным минимумом. Если $E = f(x)$ меньше, то изменить значение глобального минимума.

(b) Сгенерировать новую точку $x^l = G(x; T(k))$.

(c) Вычислить значение функции в ней $E^l = f(x^l)$.

(d) Сгенерировать случайное число α из интервала [0; 1]

(e) Если $\alpha < h(E^l - E; T(k))$, то установить $x \leftarrow x^l$; $E \leftarrow E^l$ и перейти к следующей итерации. Иначе повторить шаг (b), пока не будет найдена подходящая точка x^l.

4.3 Больцмановский отжиг

Исторически первой схемой метода отжига является так называемая схема Больцмановского отжига. Именно эта схема использовалась Н. Метрополисом для вычисления многомерных интегралов пути в задачах статистической физики, а также С. Киркпатриком для решения задачи нахождения оптимальной разводки микросхем. В Больцмановском отжиге изменение температуры задается формулой

$$T(k) = T_0 / \ln(1 + k);\ k > 0$$

Вероятностное распределение $Q(x;\ T)$ выбирается как нормальное распределение (рис. 8) с математическим ожиданием $\mu = t(x)$ и дисперсией $\sigma^2 = T$, то есть задается плотностью

$$g(x;\ x^l;\ T) = \frac{1}{\sqrt[2]{2\pi T}} \exp\left(- \frac{(t(x^l) - t(x))^2}{2T} \right)$$

где $t(x)$ — значение для перехода или действия в текущем автомате, а $t(x^l)$ — значение для этого перехода или действия в новом автомате.

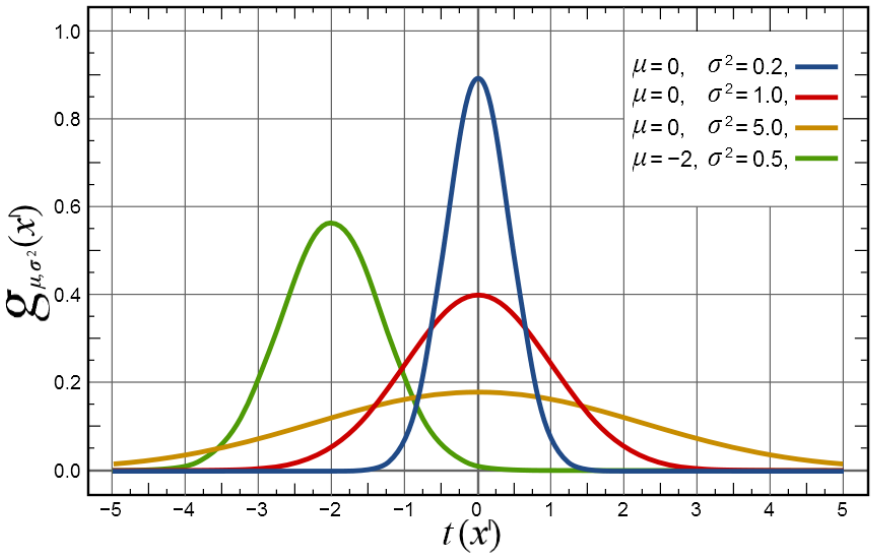

Рис. 8 — Плотность нормального распределения

4.4 Реализация отжига в рассматриваемой задаче

Нормальное распределение моделировалось с помощью центральной предельной теоремы [5].

Если сложить 12 независимых базовых случайных величин, получится грубое приближение стандартного нормального распределения. Тем не менее, с увеличением слагаемых распределение суммы стремится к нормальному.

Если обозначить как α_i независимые реализации равномерного распределения с параметрами $\mu = E\alpha_i = 0;\ \sigma^2 = D\alpha_i = 1/12$, то

$$\frac{\sum \alpha_i - \mu n}{\sigma \cdot \sqrt[2]{n}} = \frac{\sum \alpha_i}{\sqrt[2]{n/12}} \rightarrow N(0;\ 1) \text{ при } n \rightarrow \infty$$

Таким образом, можно сложить n независимых реализаций α_i, и, разделив их сумму на $\sqrt[2]{n/12}$, мы будем иметь достаточно хорошее приближение к нормальному распределению. Как показывает практика, достаточно взять $n = 24$. Для моделирования требуемого $N(0;\ T)$ достаточно домножить получившееся число на $\sqrt[2]{T}$.

Для моделирования убывания температуры использовалась непосредственно формула

$$T(k) = T_0 / \ln(1 + k);\ k = 1, 2, 3\ldots$$

По графику (рис. 9) видно, что, чем больше итераций прошло, тем медленнее меняется температура.

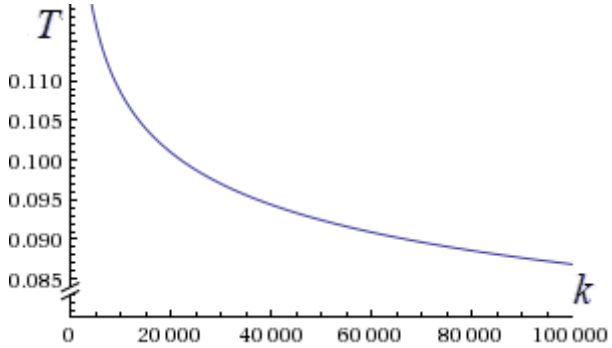

Рис. 9 — График $1 / \ln(1 + k)$

Для вычисления вероятности принятия нового состояния использовалась формула, подробнее описанная в [6]

$$h(\Delta E;\ T) = e^{\Delta E / T}$$

Из графика (рис. 10) видно, что, если $E^l - E > 0$ (новая функция приспособленности больше старой), то функция вероятности принятия $h > 1$, следовательно, новый автомат обязательно заменит старый. Если же $E^l - E < 0$ (то есть старый автомат лучше), то новый автомат имеет шанс заменить старый с тем меньшей вероятностью, чем больше разность функций приспособленности.

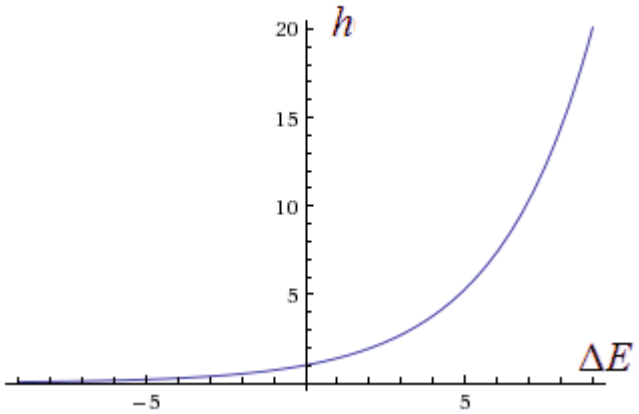

Рис. 10 — График $h(\Delta E;\ T) = e^{\Delta E / T}$

Данная реализация этих трех параметров вкупе с алгоритмом, описанным в пункте 4.1, используется в коде для решения задачи.

4.5 Функция приспособленности

Функция приспособленности $f(x)$ вычисляется следующим образом: проводится эксперимент с участием муравья с поведением, заданным текущим автоматом Мура. Генерируются k полей и эмулируются перемещения муравья в зависимости от расположения еды: он проходит по всем полям, каждый раз подсчитывается число съеденных яблок. Полученные результаты усредняются (сумма делится на k) и выдаются в качестве ответа.

5. Результаты

По итогам измерений подтвердилось предположение, что автоматы, построенные с использованием сокращенных таблиц, работают намного лучше полных. То есть в конкретном состоянии не нужно использовать все входные данные.

5.1 Результаты измерений функции приспособленности

Ниже представлен усредненный график зависимости числа съеденных яблок от числа поколений по десяти запускам метода имитации отжига (рис. 11). В каждом запуске число итераций равняется полумиллиону. В каждом запуске генерировалось только одно случайное поле. Позже будет произведено сравнение с более универсальным автоматом, построенным по десяти случайным полям.

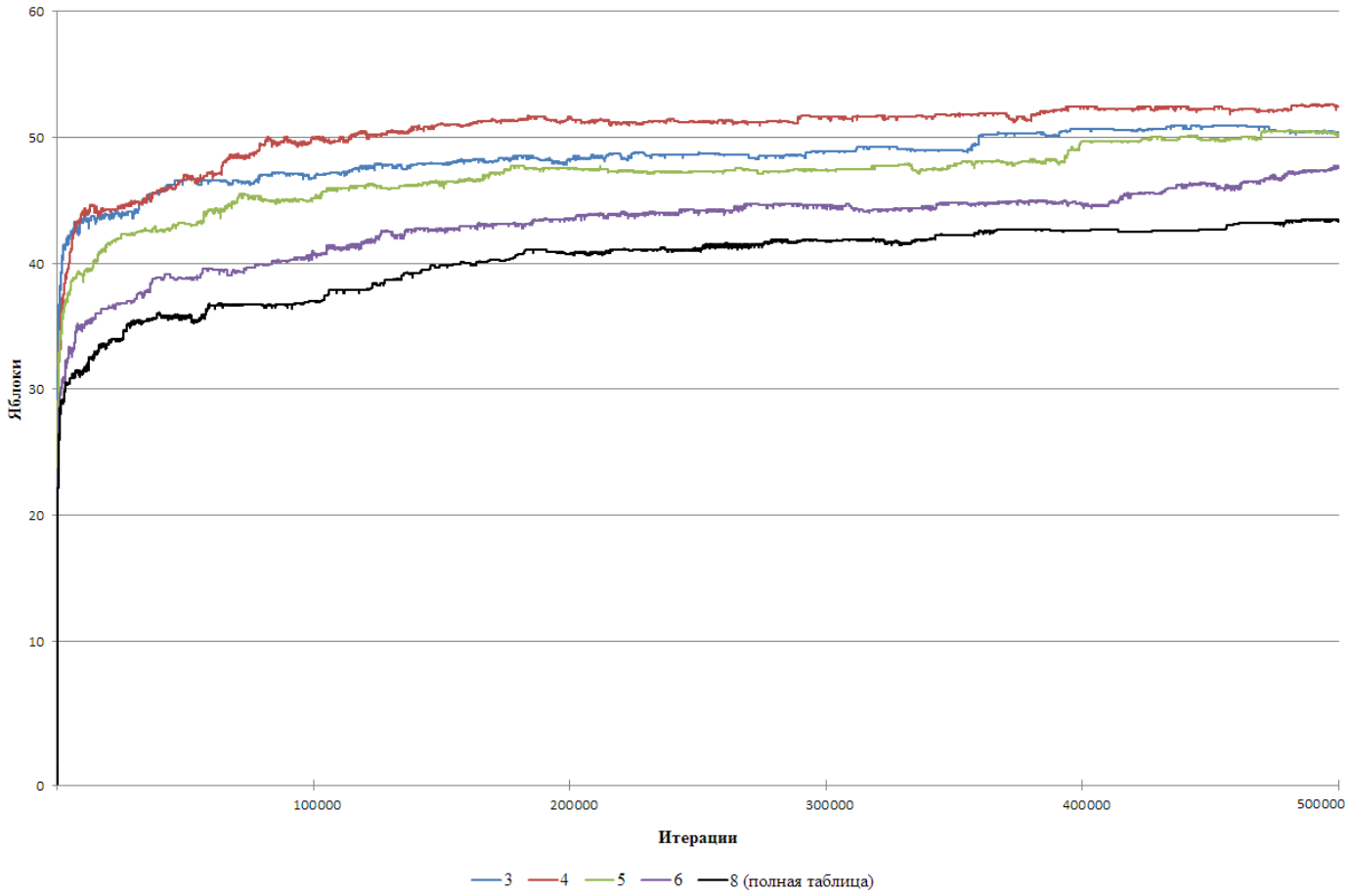

Рис.11 — Усредненный график, 10 запусков, 500 000 итераций

Из графика видно, что десять запусков не дают ясной картины происходящего, поэтому были произведены сто запусков с теми же параметрами (рис. 12).

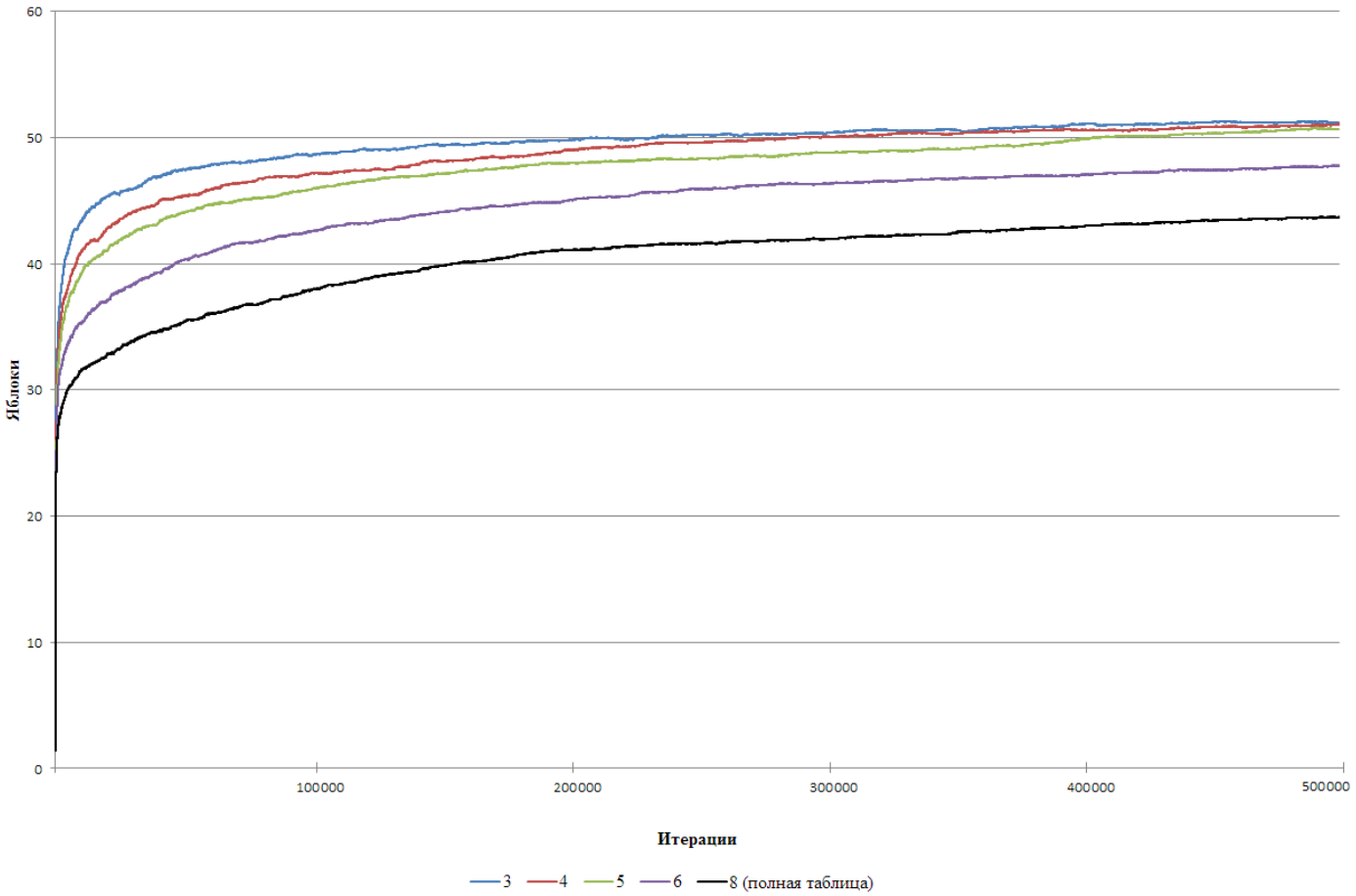

Рис. 12 — Усредненный график, 100 запусков, 500 000 итераций

Теперь видно явное преимущество автоматов на сокращенных таблицах над автоматами на полных. Также исчезла большая разница функции приспособленности между муравьем на таблице с четырьмя переменными и муравьями на таблицах с тремя и пятью.

Далее была произведена тысяча запусков с теми же параметрами. График усредненных значений функции приспособленности не приводится по причине его похожести на график сотни запусков.

На следующем графике показано максимальное количество съеденных яблок среди тысячи запусков для каждой итерации алгоритма (рис. 13).

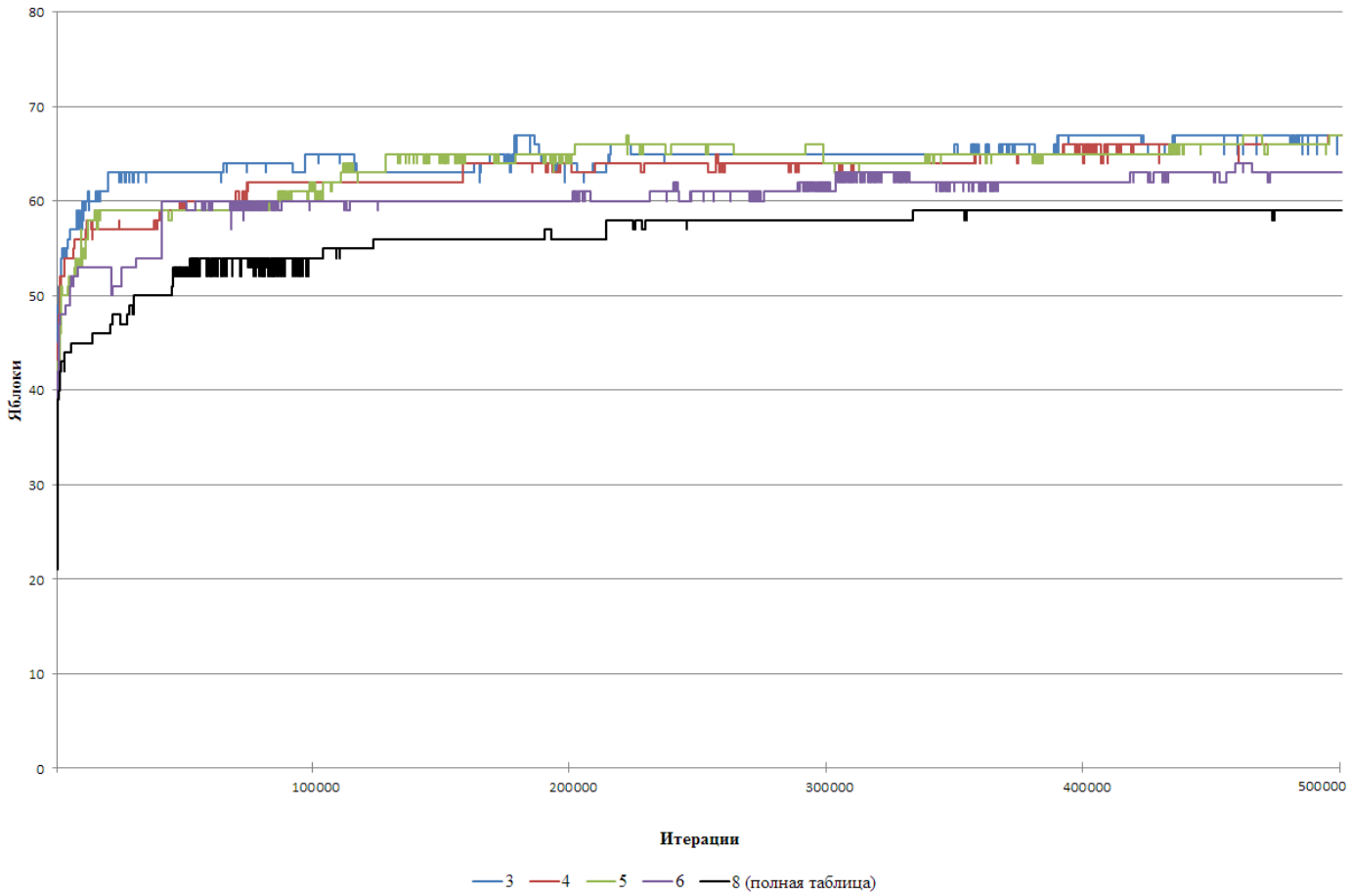

Рис. 13 — График максимумов, 1000 запусков, 500 000 итераций

К концу работы алгоритма автоматы на сокращенных таблицах с тремя, четырьмя и пятью переменными пришли к одному равному максимальному значению, поэтому было решено увеличить число итераций.

Так как существенных изменений при увеличении числа запусков до тысячи не произошло, далее будет производиться работа с числом запусков, равным ста.

Были произведены сто запусков с числом итераций, равным двум миллионам. Это позволило несколько улучшить результат. Ниже представлен график (рис. 14).

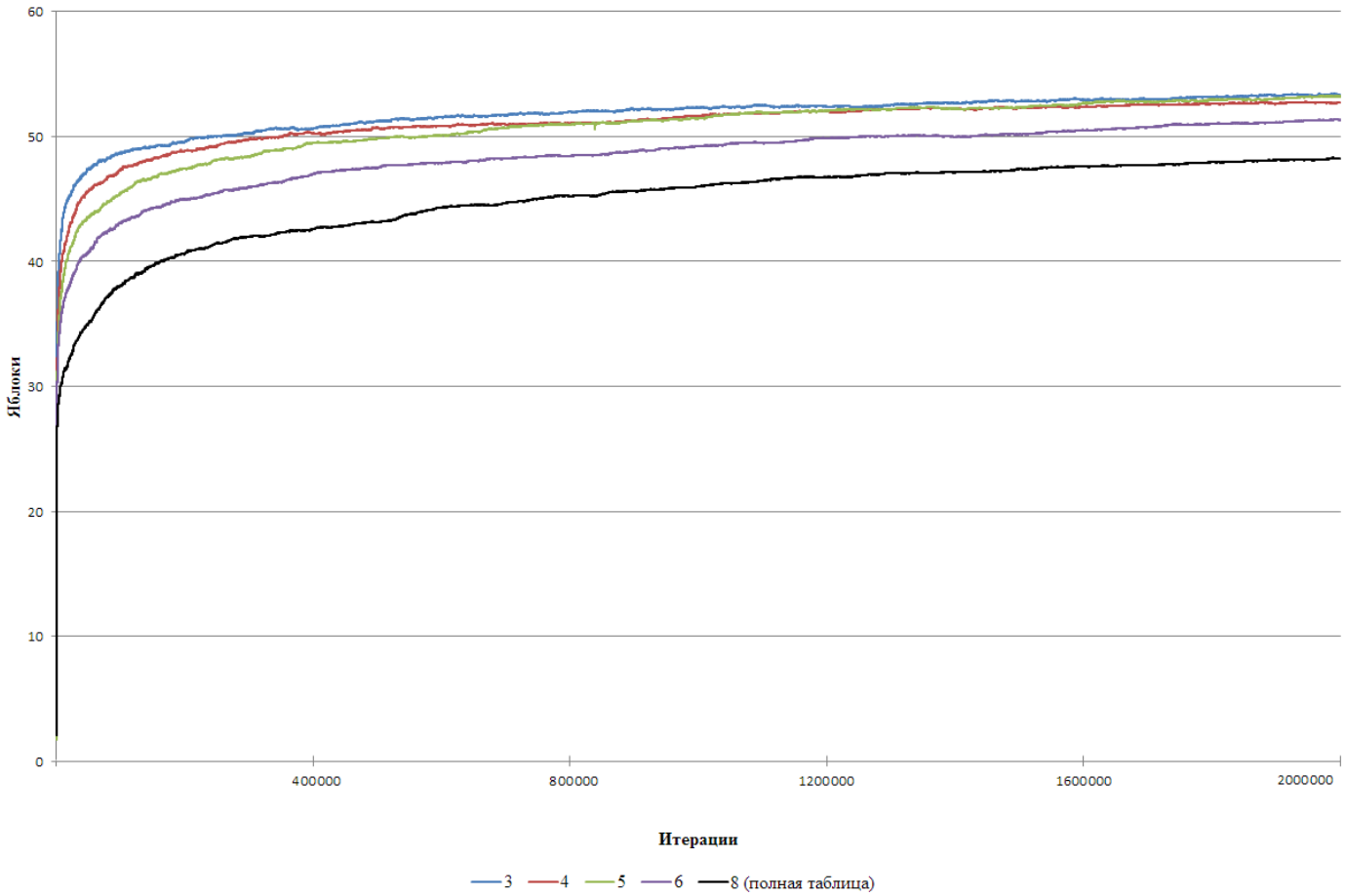

Рис. 14 — Усредненный график, 100 запусков, 2 000 000 итераций

Если посмотреть на графики максимумов для этих запусков (рис. 15), станет видно, что автомат, заданный таблицей из пяти значимых переменных, достиг наибольшего значения функции приспособленности раньше всех. Автомат, использующий сокращенную таблицу с четырьмя значимыми переменными, достиг того же максимума функции приспособленности, что и автомат с пятью, хоть и несколько позже.

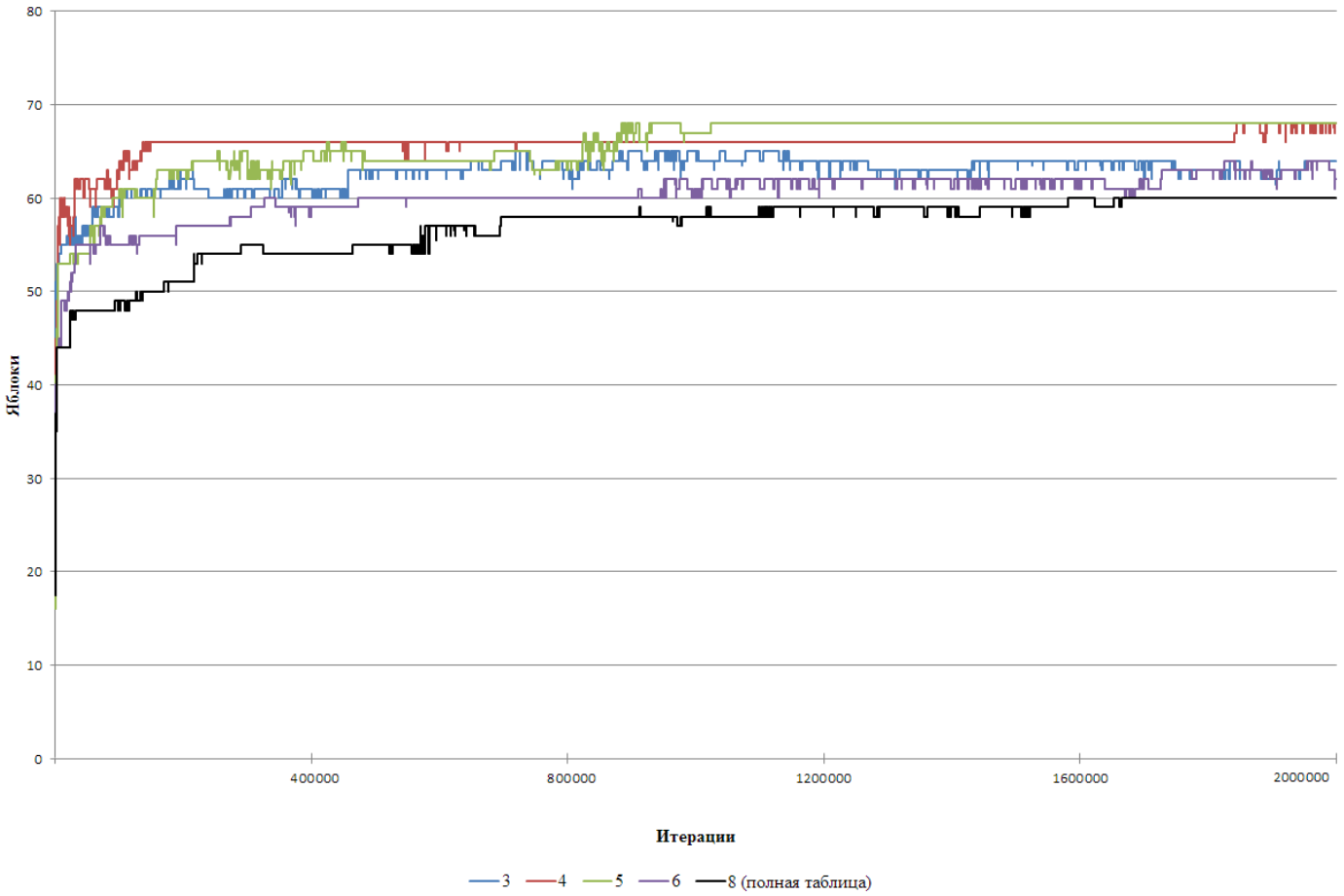

Рис. 15 — График максимумов, 100 запусков, 2 000 000 итераций

Далее число полей, сгенерированных в начале каждого запуска, было увеличено до десяти. За число съеденных яблок бралось среднее значение по всем полям.

Ниже представлен усредненный график по сотне запусков с десятью полями (рис. 16), а также график максимального числа съеденных яблок среди всех запусков для каждой итерации (рис. 17).

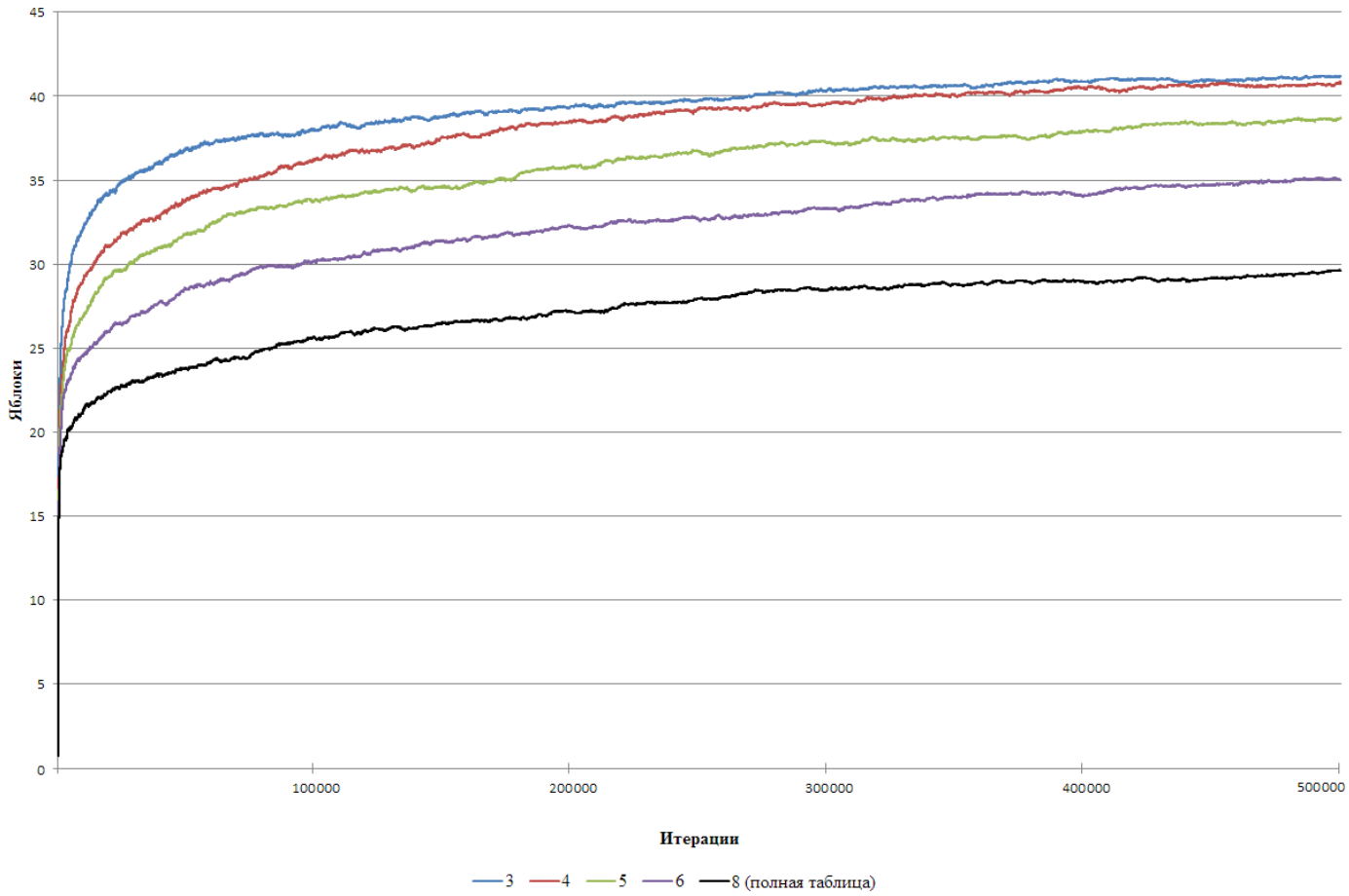

Рис. 16 — Усредненный график, 100 запусков, 500 000 итераций, 10 полей

Большое число полей позволило обойти проблему заточенности автомата лишь под одно поле, он стал более универсальным. Но эта универсальность ожидаемо привела к снижению максимума функции приспособленности.

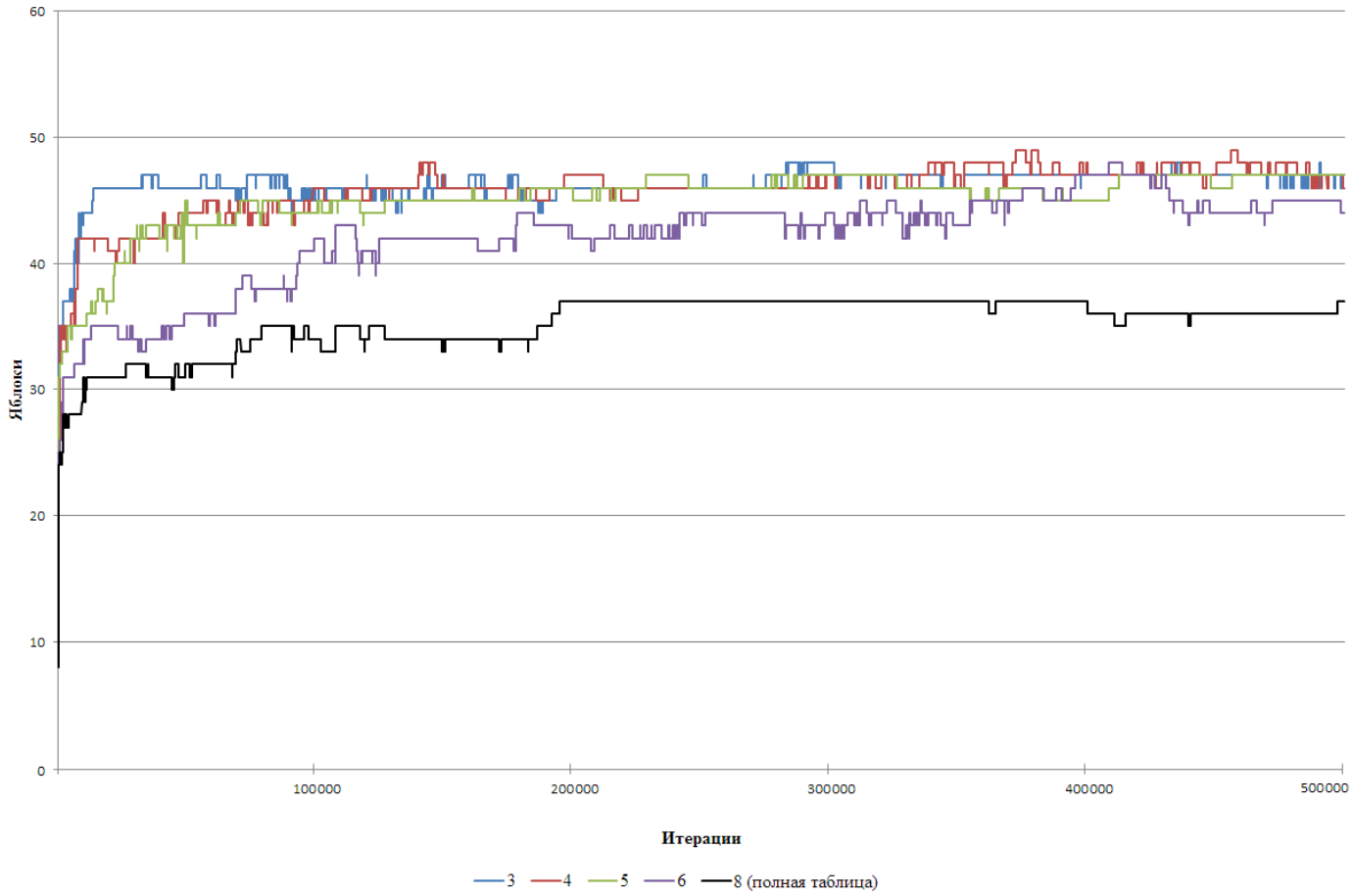

Рис. 17 — График максимумов, 100 запусков, 500 000 итераций, 10 полей

По итогам этого эксперимента оказалось, что автомат, использующий сокращенную таблицу с четырьмя значимыми переменными, вновь достиг наибольшего значения функции приспособленности, теперь и при вычислении ее на десяти полях.

5.2 Полученный автомат

Наилучшая особь получилась при использовании сокращенной таблицы с пятью значимыми переменными (табл. 1). Муравей съел 68 яблок из 112 на одном поле.

В автомате восемь состояний. Каждое состояние имеет свой номер от нуля до семи и действие, совершаемое в нем. Начальным состоянием автомата является третье. В каждом состоянии из восьми видимых клеток только пять являются значимыми, и в каждой клетке может присутствовать или отсутствовать яблоко. Таким образом, у автомата всего имеется $2^5 = 32$ входных воздействия. Для каждого из них в таблице указан переход в новое состояние.

В масках, приведенных ниже, восемь цифр отвечают за восемь клеток, видимых муравьем (рис. 18). К примеру, если первая цифра в маске — единица, то клетка с номером один на рисунке является значимой.

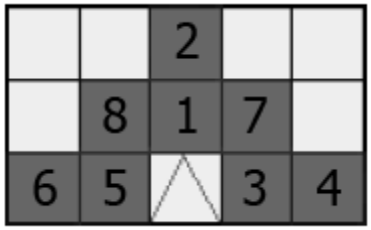

Рис. 18 — Номера видимых клеток поля в маске

Маски состояний:
0: 10111010
1: 01100111
2: 01110011
3: 01101101
4: 11110001
5: 11001110
6: 10100111
7: 01011101

Таблица 1 — Таблица переходов автомата для одного поля

№	(0;Move)	(1;Move)	(2;Right)	(3;Move)	(4;Move)	(5;Move)	(6;Right)	(7;Left)
0	0	6	0	0	0	2	5	0
1	5	6	3	3	5	4	1	0
2	2	7	0	7	3	5	0	7
3	4	3	4	5	7	3	2	0
4	7	2	7	5	2	7	0	4
5	5	0	1	0	2	6	1	1
6	2	2	7	2	4	6	2	0
7	0	4	6	6	4	0	1	7
8	5	7	4	3	4	3	1	4
9	0	2	6	2	4	3	2	4
10	6	6	6	7	6	1	0	6
11	3	3	2	5	7	0	6	2
12	6	1	3	2	1	4	5	5
13	2	5	0	7	5	7	3	3
14	6	4	5	1	1	1	7	0
15	4	2	7	7	5	7	2	7
16	1	0	6	6	7	1	4	5
17	5	2	4	3	5	7	5	6
18	1	4	0	2	6	6	3	3
19	5	5	1	1	1	7	7	6
20	6	4	7	2	6	0	5	2
21	3	0	4	2	4	5	5	5
22	6	1	0	0	0	2	7	0
23	5	4	6	2	2	2	3	3
24	2	7	4	0	0	2	4	1
25	1	3	7	2	4	3	6	2
26	5	0	4	7	1	3	0	1
27	3	3	7	3	5	6	1	6
28	7	7	0	6	6	4	5	6
29	0	2	3	1	6	4	4	5
30	3	1	5	1	0	6	1	4
31	7	1	2	0	2	7	0	7

На десяти полях наилучший автомат позволил муравью съесть 48 яблок (табл. 2). В автомате восемь состояний, начально состояние — второе, используется сокращенная таблица с четырьмя значимыми переменными.

Маски состояний:

0: 10010110
1: 11100100
2: 10011001
3: 11000110
4: 00111001
5: 00100111
6: 10100110
7: 11001001

Таблица 2 — Таблица переходов автомата для десяти полей

№	(0;Move)	(1;Move)	(2;Move)	(3;Left)	(4;Right)	(5;Move)	(6;Move)	(7;Move)
0	2	3	3	5	5	5	3	6
1	7	5	5	5	4	4	0	7
2	6	3	2	5	3	3	1	6
3	3	5	7	2	6	7	6	7
4	5	4	6	6	7	1	3	3
5	7	5	0	2	6	2	4	7
6	0	1	0	6	2	5	4	0
7	6	7	2	3	3	0	7	7
8	1	3	6	6	1	6	3	1
9	1	4	1	1	4	7	6	7
10	3	0	3	2	4	3	1	2
11	6	1	2	3	7	5	7	5
12	0	2	7	7	3	4	5	2
13	4	0	3	5	5	6	5	0
14	0	0	5	1	7	2	4	7
15	7	2	6	4	0	3	0	3

Заключение.

В данной работе рассмотрено использование метода имитации отжига для решения задачи «Умный муравей – 3». Эксперименты подтвердили предположение, что следует использовать сокращенные таблицы при решении задач с большим числом входных воздействий. Также показано, что для решаемой задачи оптимальным является использование сокращенных таблиц с четырьмя или пятью значимыми переменными из восьми.

Источники.

1. Тяхти А. С., Чебатуркин А. А. Описание виртуальной лаборатории на языке C# // НИУ ИТМО, кафедра компьютерных технологий, 2010. http://is.ifmo.ru/genalg/labs_2010-2011/GlOpt_instruction.pdf
2. Поликарпова Н. И., Шалыто А. А. Автоматное программирование // СПб.: Питер, 2009. http://is.ifmo.ru/books/_book.pdf
3. Данилов В. Р., Шалыто А. А. Метод представления функций переходов деревьями решений для генерации автоматов с помощью генетического программирования // Сборник научных трудов V международной научно-практической конференции «Интегрированные модели и мягкие вычисления в искусственном интеллекте». М.: Физматлит, 2009, с. 589 — 595. http://is.ifmo.ru/works/_2009_08_01_danilov.pdf
4. Поликарпова Н.И., Точилин В.Н., Шалыто А.А. Метод сокращенных таблиц для генерации автоматов с большим числом входных переменных на основе генетического программирования // Известия РАН. Теория и системы управления, 2011. №2, с.100 — 117. http://is.ifmo.ru/works/_polikarpova_samolet.pdf
5. Метод имитации отжига. Конспект лекций А. Лопатина. http://rain.ifmo.ru/~buzdalov/lab-2011/books/annealing.pdf
6. Luke Sean “Essentials of Metaheuristics” // Online Version 1.0, 2010. http://rain.ifmo.ru/~buzdalov/lab-2011/books/metaheuristics.pdf