

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ  
ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ  
ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И ПРОГРАММИРОВАНИЯ  
КАФЕДРА «КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ»

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

---

Вариант № 53

А.В.Халиуллин

Санкт-Петербург  
2010

## Оглавление

1.	Введение .....	3
2.	Постановка задачи.....	4
2.1.	Автомат Мили .....	4
2.2.	Формулировка задачи «Умный муравей — 3» .....	4
3.	Реализация.....	5
3.1.	Представление автомата в виде полных таблиц .....	5
3.2.	Описание метода скрещивания.....	6
3.3.	Описание метода мутации .....	7
3.4.	Описание метода формирования нового поколения.....	7
3.4.1.	Клеточный генетический алгоритм.....	7
3.4.2.	Большая мутация .....	7
3.5.	Вычисление функции приспособленности.....	8
4.	Результат работы алгоритма .....	8
4.1.	Метод представления автомата.....	8
4.2.	Клеточный алгоритм .....	11
4.3.	Полученная особь.....	11
	Заключение .....	17
	Источники .....	18

## 1. Введение

Цель лабораторной работы — изучение генетических алгоритмов для построения конечных автоматов. В частности, изучается возможность построения конечного управляющего автомата Мили, наиболее эффективно решающего задачу «Умный муравей-3».

Требуется написать несколько плагинов для виртуальной лаборатории, которые решают задачу нахождения такого автомата в соответствии с рядом требований.

## 2. Постановка задачи

При помощи генетического алгоритма построить конечный автомат Мили, наиболее эффективно решающий задачу «Умный муравей-3».

Использовать клеточный генетический алгоритм, представление автомата в виде полных таблиц.

### 2.1. Автомат Мили

Конечный автомат Мили — автомат, в котором выходное действие генерируется в зависимости от текущего состояния и входного воздействия.

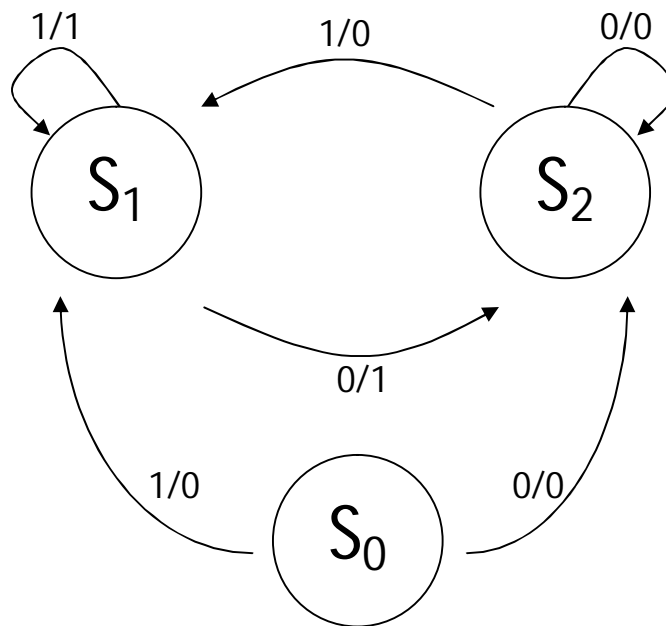


Рис. 1. Автомат Мили

На рис. 1 изображен пример диаграммы переходов для автомата Мили из трех состояний. Стрелка соответствует переходу между состояниями, первое число рядом со стрелкой – входное воздействие, второе – выходное действие.

### 2.2. Формулировка задачи «Умный муравей-3»

В задаче об «Умном муравье-3» рассматривается поле, располагающееся на поверхности тора и имеющее размер 32 на 32 клетки. Каждая клетка поля с некоторой, заранее определенной, вероятностью содержит яблоко.

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

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

- ничего не делать.

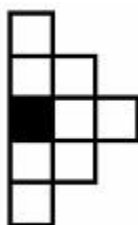


Рис. 2. Область видимости муравья.

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

### 3. Реализация

В виртуальной лаборатории (Java) уже реализовано представление особи для задачи «Умный муравей-3» с помощью полных таблиц. Реализованы были: плагин `ga.cellular2` — клеточный генетический алгоритм, плагин `ext.ant.full.table.crossover2` — одноточечное скрещивание, а так же были внесены изменения в ряд плагинов для расширения их функциональности.

#### 3.1. Представление автомата в виде полных таблиц

Автомат для данной задачи описывается множеством состояний, множеством переходов, а так же стартовым состоянием. Каждый из переходов задается исходным состоянием, конечным состоянием, входным воздействием и выходным воздействием.

Муравей видит восемь клеток — в каждой из них еда либо есть, либо нет. Итого существует 256 различных входных воздействий. Таблица, представляющая каждый автомат, имеет следующую структуру: каждому состоянию соответствует один столбец таблицы, каждому входному воздействию – строка. Если из состояния  $A$  в состояние  $B$  по входному воздействию  $\alpha$  ведет переход с выходным действием  $\beta$ , то в ячейке, лежащей в строке  $\alpha$  и в столбце  $A$ , будет записана пара  $(B, \beta)$ .

В табл. 1 отображены переходы автомата, представленного на рис. 1.

Таблица 1. Таблица переходов автомата

Состояние	$S_0$	$S_1$	$S_2$
Входное воздействие			
0	$(S_2, 0)$	$(S_2, 1)$	$(S_2, 0)$
1	$(S_1, 0)$	$(S_1, 1)$	$(S_1, 0)$

В реализации для индексации столбцов используется номер состояния, а для индексации строк – двоичное число, получаемое при записи подряд входных значений для каждой из восьми клеток (ноль – еда в клетке есть, единица – еды нет). Например, ситуации, когда муравей не видит еды, соответствует нулевой набор воздействий (00000000), а если во всех просматриваемых клетках есть еда — то 255-й (11111111).

### 3.2. Описание метода скрещивания

В работе рассмотрено два метода скрещивания: многоточечное скрещивание, которое уже реализовано в лаборатории, и одноточечное. В процессе изучения выяснилось, что оба метода дают одинаковые результаты в данном случае. Рассмотрим эти методы применительно к используемому представлению автомата.

На вход оператору скрещивания подается две особи. Оператор формирует две новые особи. Деление «генов» между потомками происходит взаимоисключающим образом – то есть, если какой-то из параметров первый потомок унаследовал от родителя *A*, то второй потомок унаследует его от родителя *B*. Таким образом, потомки сразу получают свои стартовые состояния. Далее наследуются таблицы:

- Для одноточечного скрещивания:

Каждый столбец таблицы разбивается на две части в случайном месте, каждый потомок первую часть столбца наследует от одного случайно выбранного родителя, а вторую – от другого. Процесс проиллюстрирован на рис. 3.

Родитель №1	Родитель №2	Потомок №1	Потомок №2
A	H	A	H
B	I	B	I
C	J	J	C
D	K	K	D
E	L	L	E
F	M	M	F
G	N	N	G

Рис. 3. Одноточечный кроссовер

- Для многоточечного скрещивания процесс аналогичен, но каждый столбец разбивается не на две части, а на несколько, как показано на рис. 4.

Родитель №1	Родитель №2	Потомок №1	Потомок №2
A	H	A	H
B	I	B	I
C	J	J	C
D	K	K	D
E	L	E	L
F	M	F	M
G	N	G	N

Рис. 4. Многоточечный кроссовер

### 3.3. Описание метода мутации

Использовался уже реализованный в виртуальной лаборатории метод mutation1 (Weak mutation). Он действует следующим образом — с вероятностью 50% начальное состояние изменяется на случайно выбранное. Далее случайным образом выбирается состояние, которое мутирует по следующей схеме: для каждого перехода из этого состояния случайным образом выбирается, будет ли изменяться номер состояния, в которое ведет переход, или выходное воздействие для этого перехода. Если было выбрано выходное воздействие, то оно заменяется на случайное. Если же было выбрано состояние, в которое ведет переход, то оно заменяется на случайное, номер которого не превосходит максимального номера среди всех состояний, в которые вели переходы из мутирующего.

### 3.4. Описание метода формирования нового поколения

#### 3.4.1. Клеточный генетический алгоритм

Рассмотрим поле, разбитое на  $m \times n$  клеток и лежащее на поверхности тора. В каждой клетке поля находится одна особь. Переход к следующему поколению осуществляется следующим образом: для каждой клетки среди четырех ее соседей выбирается лучшая особь, которая мутирует с вероятностью 15% и скрещивается с особью в текущей клетке. Из получившихся потомков выбирается лучший. Если его значение функции приспособленности больше, чем для особи в текущей клетке, то особь в клетке замещается этим потомком. Если значение функции приспособленности для особи в текущей клетке больше, чем для потомка, то она замещается потомком с вероятностью, вычисляемой по формуле:

$$p = 3^{\frac{a-b}{3 \cdot b}}$$

где  $a$  – значение функции приспособленности потомка,  $b$  – родителя.

### 3.4.2. Большая мутация

В алгоритме была применена идея большой мутации. Если в  $T$  последних поколениях не менялось лучшее значение функции приспособленности, то лучшие 10% особей остаются на своих местах, 45% особей мутируют и еще 45% особей заменяются на новых.  $T$  является настраиваемым параметром.

## 3.5. Вычисление функции приспособленности

Перед запуском алгоритма генерируется  $k$  случайных карт. Пользователем задаются значения  $p_{\min}$  и  $p_{\max}$ . Для  $i$ -ой карты вероятность появления в любой конкретной клетке яблока равна  $\frac{i \cdot (p_{\max} - p_{\min})}{k} + p_{\min}$ . Поведение оцениваемой особи моделируется на каждой карте из набора, и значение функции приспособленности для особи считается по формуле:

$$f = \sum_{i=1}^k \frac{a_i}{b_i \cdot k}$$

где  $a_i$  – число яблок, съеденных муравьем на  $i$ -ой карте,  $b_i$  – число яблок на  $i$ -ой карте.

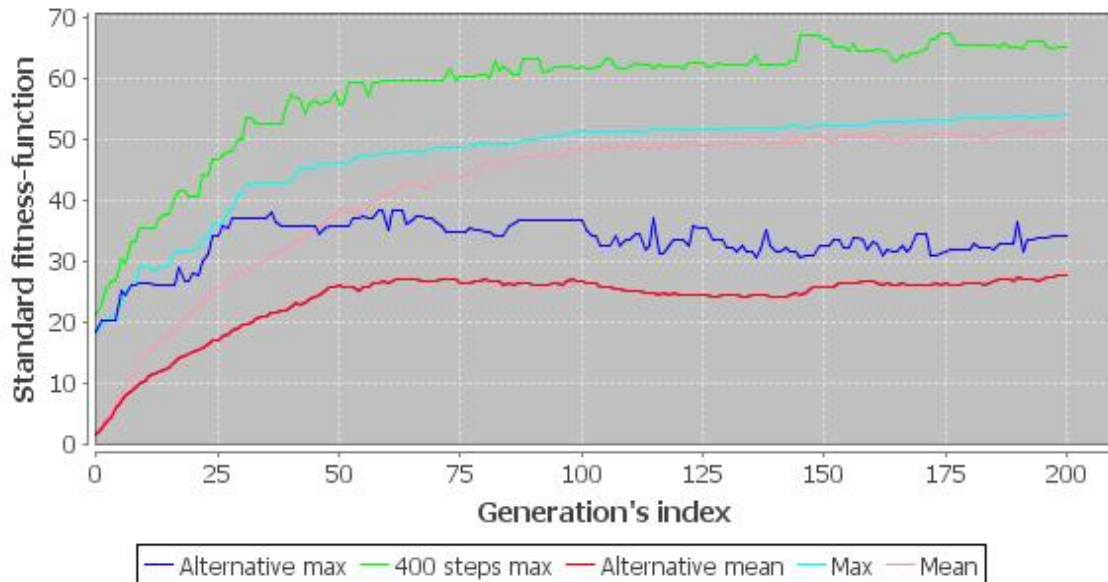
Кроме основного набора карт, создается добавочный набор, предназначенный для определения значения функции приспособленности на независимом наборе. Следует отметить, что значение функции приспособленности на независимом наборе более показательнее, так как не отображает степень «привыкания» особи к набору карт.

## 4. Результат работы алгоритма

### 4.1. Метод представления автомата

В ходе работы было обнаружено, что график значений функции приспособленности на дополнительном наборе значительно отстает от основного графика (рис. 5). Было сделано предположение, что это происходит по причине использования полных таблиц для представления особи.





**Рис. 5. График работы алгоритма для полных таблиц.**  
**Голубым цветом изображен график максимума функции приспособленности на основном наборе, синим – на дополнительном.**

Алгоритмом довольно быстро генерируются особи, которые умеют правильно поступать, когда в поле зрения находится ровно одно яблоко. Таких ситуаций ровно восемь – поэтому быстро подбирается правильное действие для каждой из них. Но остается еще 247 различных входных воздействий, когда муравей видит более одного яблока. В процессе работы алгоритма муравей может не увидеть многие из этих воздействий, и, следовательно, выходные действия для соответствующих переходов будут подбираться случайным образом. Так, например, если такому муравью встретится ситуация, когда во всех восьми клетках он видит яблоки, то нет гарантии, что он сможет собрать хотя бы одно из них.

В то же время, при использовании сокращенных таблиц или деревьев решений муравей будет ориентироваться не по всему набору входных воздействий, а по нескольким, которые в процессе эволюции были выбраны важными. Так, скорее всего, важным будет наличие яблока в клетке непосредственно перед муравьем, и вне зависимости от того, видит ли он другие яблоки, он съест яблоко, стоящее перед ним. То есть, при представлении особи сокращенными таблицами или деревьями решений муравей может правильно поступать в ситуациях, которые он не встречал в процессе обучения.

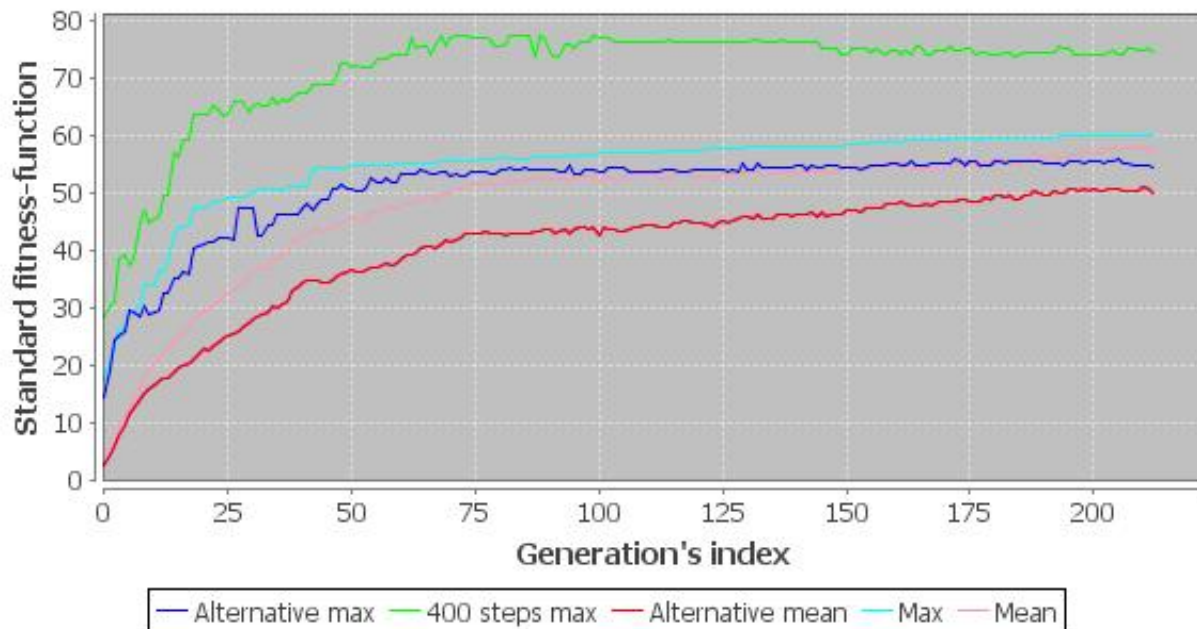


Рис. 6. График работы алгоритма для сокращенных таблиц.

Для построения графика, приведенного на рис. 6, использована особь, представленная сокращенными таблицами [3]. Сравнивая графики, можно также заметить, что сокращенные таблицы дают лучший рост максимума функции приспособленности.

Следует отметить, что при проверке этой гипотезы (о причине отставания значения функции приспособленности на дополнительном наборе карт) алгоритм несколько раз запускался для полных и сокращенных таблиц на картах с вероятностью появления яблока 5–7%, через фиксированное время бралась лучшая особь и запускалась на карте, представленной на рис. 7. Муравей, использующий полные таблицы, съедает 0–5 яблок, а муравей, использующий сокращенные таблицы – 55–100.

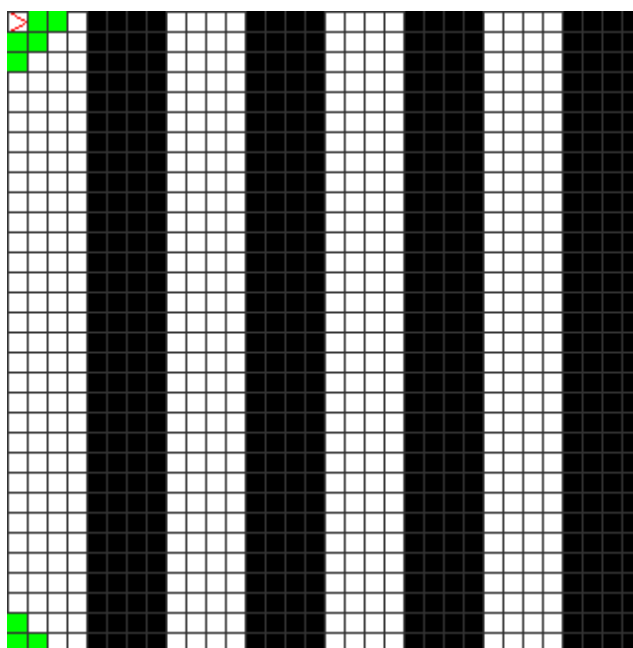


Рис. 7. Карта с вертикальными полосами яблок

## 4.2. Клеточный алгоритм

На графике (рис. 8) представлен результат работы классического генетического алгоритма при тех же параметрах, которые использовались для построения графиков выше (рис. 5, 6).

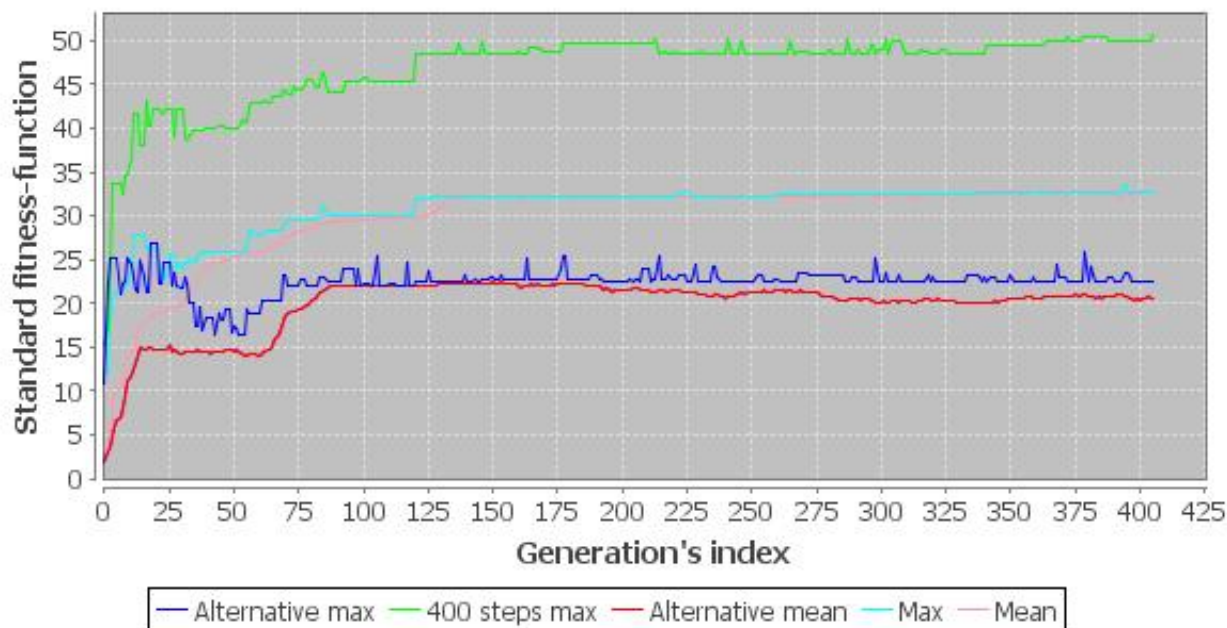


Рис. 8. График работы классического генетического алгоритма.

Видно, что график максимума функции приспособленности практически прекратил свой рост при значении примерно 33, в то время как клеточный генетический алгоритм быстро добирается до отметки 60. Очевидно преимущество клеточного генетического алгоритма перед классическим для решения данной задачи.

К минусам алгоритма можно отнести низкую скорость работы. Однако клеточный генетический алгоритм хорошо поддается распараллеливанию. Таким образом, при наличии нескольких процессоров или даже компьютеров можно добиться существенного прироста производительности.

## 4.3. Полученная особь

В результате работы алгоритма (рис. 9) была получена особь, съедающая в среднем 40% яблок на картах с вероятностью появления яблока в клетке, равной 5%.

В табл. 2 представлен автомат этой особи, стартовое состояние – нулевое, состояния нумеруются с нуля.

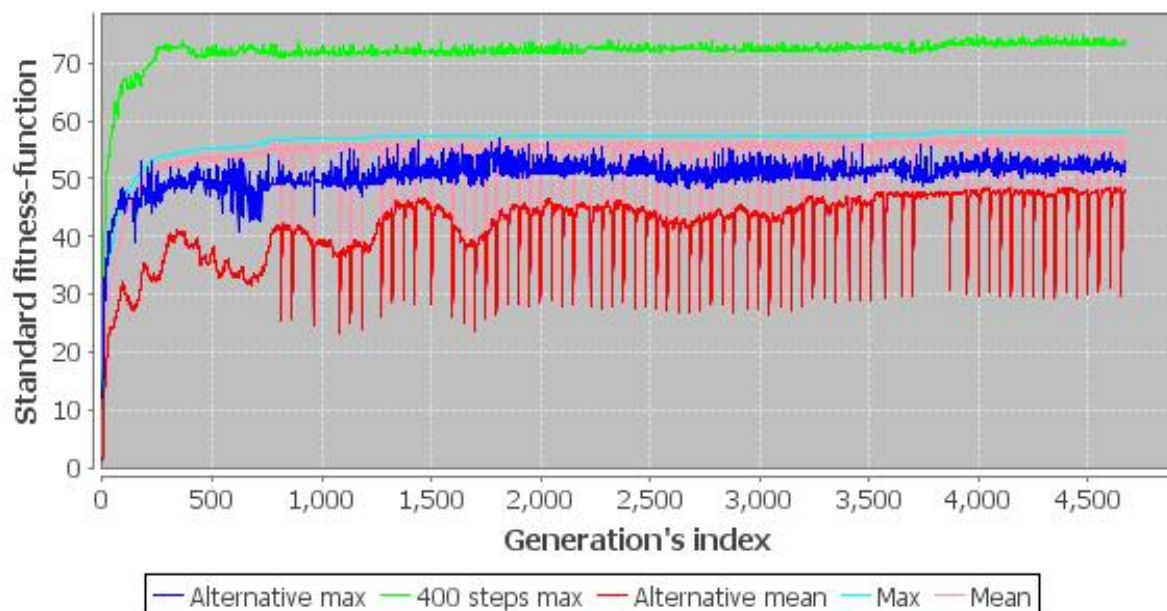


Рис. 9. Результат работы.

Таблица 2. Таблица переходов полученного автомата

Входное воздействие	Состояние №0	Состояние №1	Состояние №2	Состояние №3	Состояние №4
0	1 L	3 L	3 M	4 L	4 M
1	2 M	3 M	4 M	2 M	4 M
2	2 L	0 M	0 M	4 M	4 M
3	3 M	2 L	1 L	2 M	0 M
4	0 R	0 L	2 R	0 L	2 R
5	0 R	4 M	3 R	1 M	1 M
6	0 M	3 R	3 M	3 L	1 M
7	1 R	0 L	2 R	0 R	4 L
8	1 L	2 L	4 M	4 R	3 R
9	3 M	1 L	2 M	3 R	2 M
10	0 M	1 R	0 M	2 L	2 R
11	1 M	1 M	0 R	4 R	0 M
12	1 L	4 M	3 R	2 M	2 L
13	3 R	0 R	2 R	3 L	2 M
14	4 R	4 R	2 L	4 M	1 R
15	4 R	2 L	2 R	0 R	0 L
16	1 R	0 M	0 M	3 L	4 R
17	4 R	0 M	4 M	4 L	2 R
18	2 R	3 L	4 M	3 M	0 L
19	1 M	4 R	3 M	0 R	2 M
20	1 L	1 L	2 L	2 R	1 L
21	3 M	0 R	3 L	2 M	1 M
22	2 M	3 L	0 L	0 R	0 R
23	0 M	4 R	4 R	1 M	4 L
24	3 R	4 R	0 L	3 L	2 R
25	1 R	4 R	4 R	2 R	3 M
26	4 R	3 M	4 M	2 M	1 R

27	4	R	3	M	4	M	4	M	2	M
28	0	R	3	L	3	M	3	R	3	M
29	0	M	1	L	0	M	4	M	3	L
30	1	M	0	L	0	M	1	L	1	R
31	1	M	2	R	2	L	1	M	0	L
32	1	M	4	L	2	R	4	M	1	L
33	1	M	0	M	0	L	3	M	4	M
34	1	R	2	R	2	M	1	R	4	M
35	0	R	3	L	1	M	2	L	1	M
36	3	L	3	R	2	R	0	M	4	R
37	2	L	3	R	1	R	1	L	0	L
38	0	L	3	R	3	L	0	R	4	R
39	4	R	0	M	1	M	3	R	2	L
40	4	R	4	R	4	L	0	L	4	R
41	0	M	2	M	4	R	1	L	2	R
42	3	R	4	R	1	L	0	L	4	L
43	0	R	0	M	0	M	4	L	4	L
44	0	L	1	L	3	R	0	M	3	R
45	3	L	1	L	2	R	4	M	4	R
46	3	R	1	R	0	R	3	L	0	M
47	3	R	2	M	3	L	3	L	3	M
48	0	M	2	L	0	R	0	L	0	M
49	1	M	3	R	4	M	4	R	1	L
50	2	L	4	M	2	R	0	M	4	M
51	1	M	3	M	3	L	3	L	0	M
52	4	L	4	R	2	R	0	M	1	M
53	0	M	1	R	0	L	3	L	1	L
54	2	R	0	L	0	R	1	M	3	M
55	0	M	4	R	3	R	0	M	0	L
56	2	M	1	L	1	R	0	M	3	M
57	0	R	4	M	3	M	4	R	3	R
58	1	R	3	R	2	M	2	L	2	M
59	1	L	3	L	3	R	4	R	0	M
60	0	L	4	R	4	M	1	L	4	R
61	2	R	4	R	3	L	0	M	3	R
62	0	R	2	R	1	R	2	R	3	M
63	4	L	3	R	4	L	0	L	1	R
64	1	M	3	R	2	M	2	R	2	M
65	1	M	4	R	0	M	3	R	3	R
66	3	L	1	M	1	R	2	L	1	M
67	3	M	2	R	0	M	4	M	4	L
68	2	M	3	M	3	R	4	R	3	M
69	1	R	2	M	1	R	2	R	4	M
70	1	L	3	R	1	R	0	M	3	M
71	2	M	1	L	2	M	2	L	1	L
72	1	R	4	L	0	L	4	M	2	M
73	3	L	4	M	4	M	4	R	3	M
74	0	M	3	M	1	M	3	R	4	R
75	2	L	2	L	0	L	4	L	3	L

76	4	M	4	M	1	R	4	M	1	R
77	4	M	0	L	3	R	3	R	2	R
78	3	M	0	L	4	R	4	M	3	L
79	4	R	2	M	4	M	0	R	0	R
80	2	M	1	R	0	L	2	M	4	L
81	0	M	2	R	1	R	2	L	1	L
82	0	R	3	L	4	R	1	R	3	R
83	4	L	3	L	4	L	0	L	1	M
84	2	R	1	L	3	R	0	L	2	L
85	0	R	1	L	0	R	0	M	3	R
86	4	L	2	R	2	M	0	R	0	M
87	3	M	4	R	2	M	0	L	0	M
88	4	R	1	R	1	L	3	R	2	M
89	1	M	4	M	4	R	3	M	0	L
90	1	L	2	L	3	M	2	M	1	L
91	1	R	1	M	4	L	2	M	2	L
92	2	M	0	L	2	M	3	R	2	M
93	4	R	1	M	1	R	4	L	1	R
94	3	L	2	M	3	R	1	R	4	L
95	1	R	1	L	3	M	3	L	2	L
96	0	R	2	L	0	M	3	M	3	L
97	0	R	2	R	3	R	4	L	0	M
98	0	L	2	M	3	L	3	L	3	M
99	2	L	3	L	0	L	2	L	4	R
100	2	L	0	R	4	R	3	M	4	M
101	4	M	4	R	2	M	4	R	4	M
102	1	M	2	L	2	L	3	R	3	L
103	1	R	3	R	2	L	3	L	4	R
104	0	L	0	L	1	M	2	L	2	R
105	4	R	4	M	0	R	2	L	3	R
106	4	M	4	L	4	M	1	L	2	L
107	2	L	1	R	1	R	4	M	0	L
108	3	L	0	R	0	L	0	R	4	R
109	3	R	1	L	0	M	1	L	1	M
110	2	M	2	L	1	L	4	R	2	R
111	2	M	3	M	3	M	2	M	3	L
112	4	M	2	R	2	M	2	L	4	L
113	2	R	1	L	1	R	4	R	3	L
114	2	R	4	M	4	M	2	M	0	R
115	1	L	2	M	4	M	2	M	3	R
116	0	M	0	R	1	M	0	M	4	M
117	1	R	2	R	2	M	2	M	3	M
118	2	M	0	R	0	R	2	M	0	L
119	2	R	4	M	1	R	4	M	0	L
120	2	R	2	M	3	M	3	L	4	R
121	2	M	3	R	4	R	3	M	1	R
122	4	R	0	L	0	R	2	M	0	L
123	3	L	2	M	4	L	2	R	1	L
124	3	M	4	M	0	M	3	M	3	R

125	1	L	4	R	3	R	1	L	3	M
126	0	R	1	L	3	M	0	M	1	L
127	2	M	3	R	4	M	1	L	3	R
128	3	L	3	M	4	L	3	M	3	M
129	4	M	0	M	2	M	3	M	3	M
130	4	M	3	L	2	R	2	M	4	L
131	4	M	2	M	4	L	2	L	0	R
132	4	L	2	R	4	R	1	R	2	M
133	1	L	2	M	0	R	0	M	4	M
134	0	R	1	M	4	M	3	R	4	M
135	4	L	2	R	0	M	2	L	1	M
136	4	L	1	R	0	M	3	R	2	L
137	4	M	0	L	2	R	1	L	2	M
138	3	L	0	R	1	M	1	L	4	L
139	2	L	1	M	1	L	1	R	0	R
140	3	R	0	M	2	R	0	R	4	R
141	0	R	3	R	1	R	3	R	4	R
142	0	R	4	L	3	M	1	M	1	R
143	3	L	4	L	0	L	3	R	0	M
144	4	L	2	M	2	M	3	M	4	M
145	4	L	2	L	1	M	3	L	3	M
146	2	M	1	L	0	L	2	M	3	L
147	1	M	3	M	0	M	1	L	0	M
148	4	M	1	M	0	M	4	M	1	L
149	4	R	2	L	4	L	1	M	1	M
150	0	L	2	R	0	L	4	R	4	M
151	2	R	0	M	2	L	2	R	3	L
152	2	R	0	M	1	L	0	R	4	R
153	4	L	2	R	0	L	2	R	1	R
154	4	R	3	M	0	L	1	M	1	L
155	0	M	4	R	1	L	4	M	2	L
156	3	R	4	L	3	M	4	R	4	R
157	4	R	3	M	0	M	1	M	0	L
158	0	R	1	L	1	M	3	R	1	L
159	3	M	2	R	4	R	3	M	3	R
160	3	L	3	M	0	R	3	R	4	L
161	4	L	2	M	4	M	1	M	3	M
162	0	R	4	M	1	R	4	M	4	M
163	1	R	2	L	1	M	3	M	1	M
164	4	L	1	M	3	L	0	L	4	R
165	0	L	2	R	0	L	3	R	0	R
166	1	R	2	R	2	L	0	M	2	R
167	2	R	4	L	1	R	2	M	4	R
168	0	R	0	R	0	M	1	L	4	L
169	0	L	0	R	0	L	1	L	4	M
170	0	M	4	L	1	R	2	R	0	L
171	1	M	3	M	2	R	2	R	1	R
172	4	M	4	M	1	R	1	M	4	R
173	1	L	0	R	4	L	0	L	2	M

174	3	M	1	R	0	L	0	L	0	L
175	3	L	4	M	2	R	4	L	2	M
176	3	R	1	M	4	M	3	L	4	L
177	2	M	0	R	2	L	0	M	2	M
178	3	R	0	L	1	R	1	R	3	R
179	2	L	3	R	1	L	2	R	1	R
180	2	R	3	L	1	L	3	L	2	L
181	0	L	2	M	4	M	1	M	3	L
182	0	M	1	L	0	M	1	M	1	L
183	4	L	0	L	3	L	3	L	4	M
184	4	L	1	L	4	R	4	R	1	L
185	1	L	4	R	4	M	0	R	1	L
186	2	M	4	R	1	R	1	M	4	R
187	0	R	1	R	3	M	0	L	3	R
188	1	R	4	L	4	M	1	L	1	M
189	0	L	4	R	4	L	4	L	1	M
190	3	M	3	L	4	M	3	L	4	M
191	2	R	1	M	3	R	4	L	4	R
192	4	R	1	R	4	R	3	M	0	M
193	2	R	1	M	4	L	0	L	3	L
194	0	L	0	M	4	M	1	R	3	R
195	0	R	3	L	1	R	3	L	4	L
196	2	M	3	M	4	R	3	L	3	M
197	0	L	3	M	3	R	2	M	4	R
198	2	L	1	L	4	L	0	L	2	L
199	2	R	3	M	4	L	0	R	4	R
200	4	R	1	R	4	M	0	R	2	L
201	0	L	0	M	0	L	3	L	3	L
202	2	L	0	L	1	M	3	L	1	R
203	2	R	0	M	1	R	2	L	3	M
204	0	L	1	M	0	L	4	R	0	L
205	4	M	0	R	4	R	4	L	4	M
206	2	L	0	L	4	M	4	M	4	L
207	0	R	3	R	0	M	4	M	4	R
208	2	M	2	L	2	L	2	M	0	R
209	3	M	2	R	4	M	0	L	1	M
210	3	L	1	R	4	M	0	M	2	L
211	3	M	2	M	0	L	3	L	3	R
212	3	L	1	M	1	R	3	L	1	M
213	3	M	1	M	0	R	3	M	1	L
214	4	R	2	R	1	R	3	R	1	R
215	0	M	0	R	1	L	2	R	1	L
216	3	L	3	M	3	R	3	R	3	L
217	0	M	1	L	0	L	0	L	4	L
218	0	M	2	R	1	M	1	L	2	M
219	3	R	1	R	1	L	1	R	0	M
220	0	M	3	M	0	R	3	M	0	M
221	4	R	4	M	3	R	4	L	0	M
222	0	R	1	R	0	M	4	R	4	M



223	3	M	1	R	3	L	1	M	1	M
224	4	M	2	M	0	L	1	M	4	R
225	0	M	2	L	4	M	1	M	2	L
226	3	R	4	L	1	M	2	L	0	M
227	2	R	3	R	1	L	2	L	0	L
228	2	M	1	R	0	M	0	M	3	L
229	4	M	3	R	3	R	1	M	3	M
230	0	M	1	R	1	M	1	R	1	L
231	4	M	2	L	0	R	1	R	2	M
232	4	M	0	R	4	L	1	M	2	R
233	0	L	0	R	1	M	3	L	0	R
234	1	L	1	R	4	R	3	L	3	L
235	3	M	0	R	3	M	2	L	4	R
236	0	M	0	L	3	L	3	M	1	M
237	1	L	0	L	0	R	4	R	0	R
238	1	R	4	L	0	M	2	L	3	M
239	1	M	1	L	0	R	1	R	4	L
240	0	M	2	M	0	R	2	M	0	L
241	1	R	4	M	4	R	1	M	0	R
242	0	R	0	R	2	R	0	M	2	M
243	2	L	0	M	2	M	4	M	2	M
244	4	R	3	R	4	L	2	R	2	L
245	2	L	0	L	3	M	0	M	2	R
246	4	M	0	M	4	L	4	L	4	R
247	0	R	3	L	3	R	4	L	3	L
248	0	M	4	R	0	R	1	M	3	R
249	1	R	1	L	3	R	2	M	1	L
250	0	M	1	L	3	M	3	M	1	R
251	0	L	1	M	3	R	4	M	0	M
252	4	M	1	R	0	L	2	R	2	L
253	0	L	0	R	3	L	1	M	4	R
254	3	M	1	R	1	R	2	M	4	M
255	0	L	1	L	2	M	1	M	1	L

## Заключение

В заключение можно отметить, что рассмотренный клеточный генетический алгоритм лучше подходит для решения задачи «Умный муравей-3». Однако метод представления особи полными таблицами демонстрирует слабые результаты для этой задачи. Для получения более жизнеспособных особей следует использовать сокращенные таблицы или деревья решений. Следует отметить, что у полных таблиц существует одно преимущество – простота реализации как самой особи, так и операторов скрещивания и мутации.

## **Источники**

- 1) Генетические алгоритмы  
<http://rain.ifmo.ru/cat/view.php/theory/unordered/genetic-2005>
- 2) Калиниченко А.И. Отчет по лабораторной работе "Использование генетических алгоритмов для построения управляющих автоматов".  
Вариант №18 [http://is.ifmo.ru/genalg/labs/\\_kalinichenko.pdf](http://is.ifmo.ru/genalg/labs/_kalinichenko.pdf)
- 3) Кевер М.Е. Отчет по лабораторной работе «Применение генетических алгоритмов для решения задачи «Умный муравей-3». Вариант №63.  
[http://is.ifmo.ru/genalg/labs\\_2010-2011/Kever.pdf](http://is.ifmo.ru/genalg/labs_2010-2011/Kever.pdf)