

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное агентство по образованию
Государственное образовательное учреждение высшего профессионального
образования

**«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ»**

Факультет _____ **Информационных технологий и программирования** _____

Направление (специальность) _____ **Прикладная математика и информатика** _____

Квалификация (степень) _____ **бакалавр прикладной математики и информатики** _____

Специализация _____ ----- _____

Кафедра _____ компьютерных технологий _____ Группа _____ 4539 _____

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ

_____ Многоэтапное обучение агентов, управляемых автоматами, _____
_____ с помощью генетических алгоритмов _____

Автор квалификационной работы – С.И. Попов

Руководитель – А.А. Шалыто

Санкт-Петербург, 2010 г.

ОГЛАВЛЕНИЕ

Введение	6
Глава 1. Теоретические основы	8
1.1. Генетический алгоритм	8
1.2. Конечный автомат	12
1.3. Задача «Об умном муравье-3»	13
Выводы по главе 1	15
Глава 2. Описание использованных структур и операторов	16
2.1. Представление автоматов	16
2.2. Функция приспособленности	16
2.3. Операторы отбора автоматов в следующее поколение	17
2.3.1. Оператор отбора колеса рулетки	17
2.3.2. Турнирный оператор отбора	18
2.3.3. Ранговый оператор отбора	19
2.4. Операторы скрещивания	19
2.4.1. k -точечный оператор скрещивания	19
2.4.2. Оператор скрещивания « k -процентное смешивание»	20
2.5. Оператор мутации	20
Выводы по главе 2	20
Глава 3. Эксперименты	21
3.1. Инструментальное средство	21
3.1.1. Главное окно	21
3.1.2. Окно «Изменение конфигурации»	23
3.1.3. Окно визуализации лучшей особи	26
3.2. Результаты экспериментов	27
3.2.1. Жадная стратегия	27
3.2.2. Выбор лучшего многоточечного оператора скрещивания	28
3.2.3. Выбор лучшего оператора отбора	36
3.2.4. Многоэтапное генетическое программирование	37
3.2.4.1. Изменение оператора скрещивания.....	37

3.2.4.2. Изменение числа яблок на поле	38
3.2.4.3. Добавление «хороших» яблок на поле	47
Выводы по главе 3	52
Заключение	53
Источники	54

Введение

Идея применения эволюционного моделирования для прикладных математических и компьютерных задач появилась еще в начале 50-х годов с развитием генетики. В 1954 году была опубликована работа «Численные примеры процессов эволюции» («Esempi numerici di processi di evoluzione») норвежско-итальянского математика Нилса Алла Баричелли (Nils Aall Barricelli) [1]. Она не получила широкой известности. Начиная с 1957 года, австралийский генетик Алекс Фрейзер (Alex Fraser) опубликовал серию работ по моделированию искусственного отбора организмов [2]. Эти публикации уже содержали все основные элементы современных генетических алгоритмов. В 1960 году американский ученый Лоуренс Дж. Фогель (Lawrence J. Fogel) представил Конгрессу США доклад на тему эволюционного программирования (в это время **Фогель проводил эксперименты по выращиванию конечных автоматов**) [3]. В 1964 году немецким ученым Инго Рехенбергом (Ingo Rechenberg) была разработана идея эволюционных стратегий, а в 1965 году опубликована статья «Путь кибернетического решения экспериментальной проблемы» («Cybernetic Solution Path of an Experimental Problem») [4]. Его идеи были развиты Ханс-Полом Швэфелом (Hans-Paul Schwefel) [5]. Наконец, в 1975 году вышла книга американского ученого Джона Холланда (John Holland) «Адаптация в естественных и искусственных системах» («Adaptation in Natural and Artificial Systems») [6], в которой были даны теоретические основы генетических алгоритмов (первоначально использовались для оптимизации функций, представленных битовой строкой). Позже генетические алгоритмы стали применять для выращивания деревьев в функциональном программировании [7], автоматов и других структур, при помощи которых можно представить компьютерные программы, и данная область применения генетических алгоритмов получило название **генетическое программирование**.

В настоящей бакалаврской работе изучается один из возможных способов ускорения поиска оптимума в генетическом программировании – многоэтапное генетическое программирование, что весьма актуально так как любые эксперименты с использованием генетического программирования требуют больших вычислительных ресурсов. Генетическое программирование – это, как следует из краткой истории вопроса, изложенной выше, один из способов нахождения оптимального решения задачи при помощи использования элементов естественной эволюции, происходящей на Земле.

С точки зрения автора многоэтапное генетическое программирование представляет собой выращивание особи по частям; изменение условий решаемой задачи от более простых к более сложным, а также в смене в процессе эксперимента операторов отбора, скрещивания, их параметров.

В главе 1 изложены необходимые теоретические аспекты генетического программирования и автоматов; описана задача, на основе которой проводится многоэтапное обучение. Приведены результаты, полученные на кафедре «Компьютерные технологии» СПбГУ ИТМО до написания данной работы.

В главе 2 описаны теоретические основы использованных в работе структур и операторов.

В главе 3 приведено описание написанного автором приложения для исследования многоэтапного обучения, а также приведены условия и результаты экспериментов.

В заключении сделаны необходимые выводы.

Глава 1. Теоретические основы

Эволюционное моделирование – раздел искусственного интеллекта, изучающий искусственное моделирование естественной эволюции, проходящей на Земле.

Эволюционные алгоритмы – раздел эволюционного моделирования, объединяющий эвристические алгоритмы, основанные на принципах эволюции, такие как эволюционное программирование, эволюционные стратегии, генетические алгоритмы, генетическое программирование.

1.1. Генетический алгоритм

Генетический алгоритм – это эвристический итерационный алгоритм поиска, используемый в задачах оптимизации, а также эволюционного моделирования, использующий специальные процедуры отбора, скрещивания и мутации.

Введем основные понятия генетических алгоритмов [6, 8, 9].

Особь – специальным образом закодированное представление оптимизируемого объекта.

Популяция – конечное множество особей.

Хромосома – закодированное представление оптимизируемого объекта.

Ген – атомарный элемент хромосомы.

k -е поколение – популяция после k итераций генетического алгоритма.

Функция приспособленности – функция оценки качества (соответствия цели задачи) особи.

Оператор отбора – процедура отбора лучших особей, исходя из их функций приспособленности.

Оператор скрещивания – процедура рекомбинации (перемешивания) хромосом особей. Чаще всего используется два родителя, после скрещивания которых, получаются два потомка. Скрещивание происходит

с определенной вероятностью.

Оператор мутации – вероятностная процедура изменения генов хромосомы.

Оператор инвертирования – оператор изменения порядка генов на обратный у части хромосомы (используется редко) [8].

Элита – часть особей с лучшими результатами функции приспособленности, переходящая в следующее поколение без изменения хромосом.

Условие останова – при достижении какого значения функции приспособленности следует остановиться.

Классический генетический алгоритм выполняет процедуру оптимизации в следующем порядке:

1. Формируется начальное поколение особей, чаще всего путем их случайного генерирования.
2. Подсчитываются результаты функций приспособленности особей.
3. Происходит отбор особей в промежуточное поколение.
4. Особи из промежуточного поколения для формирования следующего поколения подвергаются изменению операторами скрещивания и мутации. При этом после оператора скрещивания потомки заменяют своих родителей в следующем поколении.
5. Если не выполняется условие останова, происходит переход к пункту 2.

Теорема схем

Введем понятие схемы. Пусть хромосома представлена в виде цепочки из нулей и единиц. Тогда схема будет представлять собой множество хромосом с нулями и единицами на некоторых определенных позициях. Те места в схеме, где может стоять как 0, так и 1, обычно обозначают символом *. Например, схеме 01*00 принадлежат две

хромосомы: 01000 и 01100. Следовательно, если в схеме m символов *, то ей принадлежит 2^m хромосом.

Пусть $P(0)$ – начальная популяция особей, $P(k)$ – текущая популяция на k -й итерации алгоритма, $M(k)$ – те хромосомы, которые попали в промежуточное поколение в результате применения оператора отбора. В итоге формируется следующее поколение $P(k + 1)$.

Рассмотрим влияние оператора отбора на сохранность схемы. Обозначим рассматриваемую схему символом S , число хромосом из $P(k)$, соответствующих этой схеме, $c(S, k)$.

Пусть в результате отбора в промежуточное поколение на k -й итерации каждая i -я особь ind_i попадает из предыдущего поколения с вероятностью $p(ind_i, k)$, пропорциональной отношению ее значения функции приспособленности $F(ind_i, k)$ к сумме значений функций приспособленности всех особей из поколения, то есть

$$p(ind_i, k) = \frac{F(ind_i, k)}{\sum_{i=1}^N F(ind_i, k)}, \text{ где } N \text{ – число особей в поколении.}$$

Пусть $F(S, k)$ – среднее значение функций приспособленности, принадлежащих схеме S в $P(k)$:

$$F(S, k) = \frac{\sum_{i=1}^{c(S, k)} F(ind_i, k)}{C(S, k)}.$$

Величина $F(S, k)$ называется средней приспособленностью схемы S на k -й итерации.

Пусть $\Phi(k)$ – среднее значение функций приспособленности на k -й итерации: $\Phi(k) = \frac{\sum_{i=1}^N F(ind_i)}{N}$. Пусть $b(S, k)$ – ожидаемое число хромосом, попавших в промежуточное поколение.

Тогда математическое ожидание хромосом, принадлежащих схеме S , можно найти по формуле: $E[b(S, k)] = c(S, k) \frac{F(S, k)}{\Phi(k)}$. Из формулы видно, что если среднее значение функции приспособленности хромосом,

принадлежащих схеме S , больше среднего значения функции приспособленности всех хромосом в поколении, то в промежуточном поколении будет больше представителей схемы S .

Для анализа операторов скрещивания и отбора введем еще несколько определений. Пусть L – длина хромосом, принадлежащих схеме S . Порядком (order) схемы называется число $o(S)$ определенных позиций с нулями и единицами. Например, $o(0^*110^*) = 4$. Охватом (definition length) $d(S)$ схемы S называется расстояние между первым и последним постоянными символами схемы (не $*$). Например, $d(*1**0*0^*) = 7 - 2 = 5$.

Как было сказано в начале главы, при скрещивании хромосомы обмениваются своими частями для получения потомков. Например, если при скрещивании хромосомы обмениваются своими половинами (у них одна точка скрещивания), то получится следующее:

$$\begin{array}{ccc} 0101 & \longrightarrow & 0100 \\ 1000 & & 1001 \end{array}$$

При анализе влияния оператора скрещивания следует отметить, что больше шансов на то, чтобы остаться целой после такого обмена, у схемы с компактно расположенными постоянными символами:

$$\begin{array}{ccc} 1**0 & \longrightarrow & 1*** \\ 01** & & 01*0 \end{array}$$

Первая схема была разрушена, а вторая осталась.

Вероятность того, что точка скрещивания окажется между первым и последним постоянными символами схемы равна $\frac{d(S)}{L-1}$. Тогда вероятность того, что схема выживет после скрещивания или не будет отобрана для скрещивания, и попадет в следующее поколение без него, равна: $1 - p_c \frac{d(S)}{L-1}$, где p_c – вероятность скрещивания.

Рассмотрим влияние оператора мутации. Вероятность того, что схема после мутации не изменится, равна $(1 - p_m)^{o(S)}$, где p_m – вероятность мутации.

Таким образом, математическое ожидание того, что схема после применения операторов отбора, скрещивания и мутации не изменится, вычисляется по формуле:

$$E[b(S, k)] = c(S, k) \frac{F(S, k)}{\Phi(k)} \left(1 - p_c \frac{d(S)}{L-1} \right)^{p_m \cdot d(S)}.$$

Эта формула является результатом теоремы схем. Существуют различные упрощения этой формулы в случае маленького значения вероятностей и большой длины хромосом, которые здесь приводиться не будут.

На основе теоремы сформулирована гипотеза.

Гипотеза 1. Генетический алгоритм стремится достичь близкого к оптимальному результату за счет комбинирования схем с приспособленностью выше среднего малого порядка и малого охвата. Такие схемы называются кирпичиками.

В книге [9] сформулированы еще одна интересная гипотеза, ставящаяся под сомнение в разд. 3.2.2 главы 3 данной работы:

Гипотеза 2. Использование меньшего числа точек разрыва в генетических операторах приводит к повышению быстродействия генетического поиска.

1.2. Конечный автомат

Конечным автоматом (finite-state machine) в теории алгоритмов принято называть математическую абстракцию, позволяющую описывать пути изменения состояния объекта в зависимости от его текущего состояния и входных данных, при условии, что общее возможное число состояний конечно [10, 11].

Конечным детерминированным автоматом типа Мили называется совокупность пяти объектов [12]:

$A = (S, X, Y, \delta, \lambda)$, где S (множество состояний автомата), X (множество входных воздействий) и Y (множество выходов) – конечные и непустые, а δ и λ – отображения вида:

$$\delta: S \times X \rightarrow S,$$

$$\lambda: S \times X \rightarrow Y$$

со связью элементов множеств S , X и Y в абстрактном времени $T = \{0, 1, 2, \dots\}$ уравнениями:

$$s(t + 1) = \delta(s(t), x(t)),$$

$$y(t) = \lambda(s(t), x(t)),$$

где t из T .

Отображение δ называется функцией переходов, отображение λ – функцией выходов автомата A .

Таким образом, на каждое входное воздействие автомат реагирует переходом в новое состояние и выходным воздействием.

Конечный автомат типа Мили не сложно представить в виде однонаправленного графа, где множество состояний S соответствует множеству вершин, отображение δ задает множество ребер, а значения функции λ и множество входных воздействий X записываются над ребрами графа. Такое представление называется диаграммой состояний или графом переходов.

1.3. Задача «Об умном муравье-3»

В задаче «Об умном муравье-3» (рис. 1) задан двумерный тор с натянутой на него декартовой координатной сеткой размером 32 на 32 [13]. Тор на заданное число процентов площади заполнен случайно расположенными «яблоками» – пищей для «муравья». Муравей – это объект, управляемый конечным автоматом.

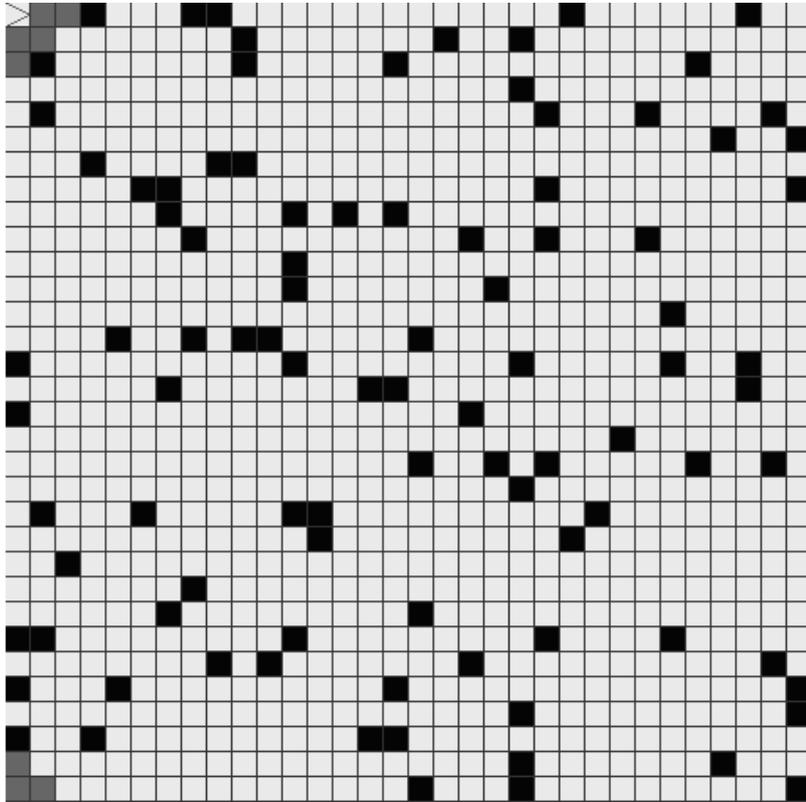


Рис. 1. Тор с муравьем и яблоками. Стрелка в левом верхнем углу изображает муравья, серые клетки – это те клетки, которые он видит. Черные квадраты обозначают места расположения яблок

На вход автомата подается воздействие, полученное в результате анализа восьми клеток, которые муравей видит около себя: две спереди, две слева, две справа и по одной спереди по диагонали (рис. 2).

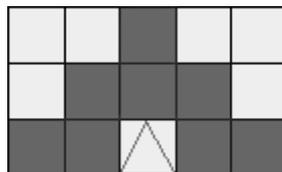


Рис. 2. Муравей и клетки, которые он видит

Выходным воздействием являются три возможных действия (хода) муравья:

- повернуться налево;
- повернуться направо;
- перейти на следующую клетку перед собой.

У муравья ограничено число ходов – 200.

Цель: за заданное число ходов съесть как можно больше яблок.

На кафедре «Компьютерные технологии» СПбГУ ИТМО ранее

проводилось исследование задачи «Об умном муравье-3». В работе [13] утверждается, что при заполненности поля яблоками больше, чем на 10%, оптимальным является автомат с одним состоянием. При небольшом числе яблок на поле, для того, чтобы автомат иногда поворачивал для поиска яблок, требуется более одного состояния. Результаты тех экспериментов показали, что при заполненности поля на 10% особь, управляемая автоматом с одним состоянием съедает в среднем на 100 полях 46,58 яблок, а особь, управляемая автоматом с 16 состояниями уже всего 40,3 яблок. При заполненности поля на 1%, когда на поле всего 10 яблок на 1024 клетки, муравей с одним состоянием в автомате съедает всего 2,68 яблок, а муравей с 16 состояниями уже 5,164.

В данной работе рассматривается немного модифицированная задача «Об умном муравье-3»: размеры поля произвольны и есть возможность добавить на поле «хорошие» яблоки с ценой в 10 раз большей, чем у обычных яблок.

Выводы по главе 1

В этой главе рассмотрены основные понятия генетических алгоритмов и выведена теорема схем. Далее было рассказано о конечных автоматах и основной задаче, на которой проводятся исследования в работе – задаче «Об умном муравье-3».

Глава 2. Многоэтапное генетическое программирование

Как было сказано во введении, бакалаврская работа посвящена многоэтапному генетическому программированию: выращиванию особи по частям; изменению условий решаемой задачи от более простых к более сложным; смене в процессе эксперимента операторов отбора, скрещивания, их параметров.

2.1. Представление автоматов

В работе в задаче «Об умном муравье-3» особью-муравьем управляет конечный автомат типа Мили.

Автомат представлен в виде массива байтов, в котором первый элемент S содержит номер начального состояния автомата. За ним следует n секторов (n – число состояний автомата), каждый из которых содержит m пар чисел (m – число входных воздействий), каждая из которых состоит из следующего состояния δ и выходного воздействия λ (рис. 3).

S	$\delta(1, 1)$	$\lambda(1, 1)$...	$\delta(1, m)$	$\lambda(1, m)$...	$\delta(n, m)$	$\lambda(n, m)$
-----	----------------	-----------------	-----	----------------	-----------------	-----	----------------	-----------------

Рис. 3. Представление автомата типа Мили

Муравей видит восемь клеток на поле, на каждой из которых может находиться обычное или хорошее яблоко, поэтому всего у управляющего им автомата возможны $3^8 = 6561$ входное воздействие.

2.2. Функция приспособленности

В работе для оценки «выращенных» генетическим программированием автоматов используется следующая функция приспособленности.

Муравью дается определенное число попыток на различных случайно сгенерированных полях (в смысле расположения яблок). На

каждой попытке подсчитывается число яблок, которые муравей успел съесть за 200 отведенных ему ходов (этот параметр можно варьировать, подробнее в главе 3). При этом если съедено хотя бы одно яблоко, к сумме съеденных на одной попытке яблок k прибавляется дробь, в числителе которой стоит число k , а в знаменателе – номер хода, на котором было съедено последнее яблоко. Итоговое значение функции приспособленности считается как среднее арифметическое значений функции за отведенное муравью число попыток.

К числу съеденных яблок прибавляется дробь для того, чтобы учитывать случай, когда две особи в поколении съедят за 200 ходов одинаковое число яблок. При этом более успешной считается та особь, которая успела съесть яблоки за меньшее число ходов. Это имеет значение при малом числе попыток на случайно сгенерированных полях.

2.3. Операторы отбора автоматов в следующее поколение

Для многоэтапного генетического программирования в работе используется несколько операторов отбора. Опишем каждый из них.

2.3.1. Оператор отбора колеса рулетки

Оператор отбора назван так из-за сходства механизма отбора с известной игрой.

Можно принять сумму значений функций приспособленности всех особей из поколения за единицу. Тогда на отрезке $[0, 1]$ каждой особи будет отведен интервал длиной L , соответствующий целой части отношению значения ее функции $f(i)$ приспособленности (где i – номер особи) к сумме всех функций приспособленности: $L = \left\lfloor \frac{f}{M} \right\rfloor$, $\sum_{i=1}^N f(i) = 1$, где N – число особей в поколении. Отрезок $[0, 1]$ можно представить в виде колеса рулетки, как показано на рис. 4.

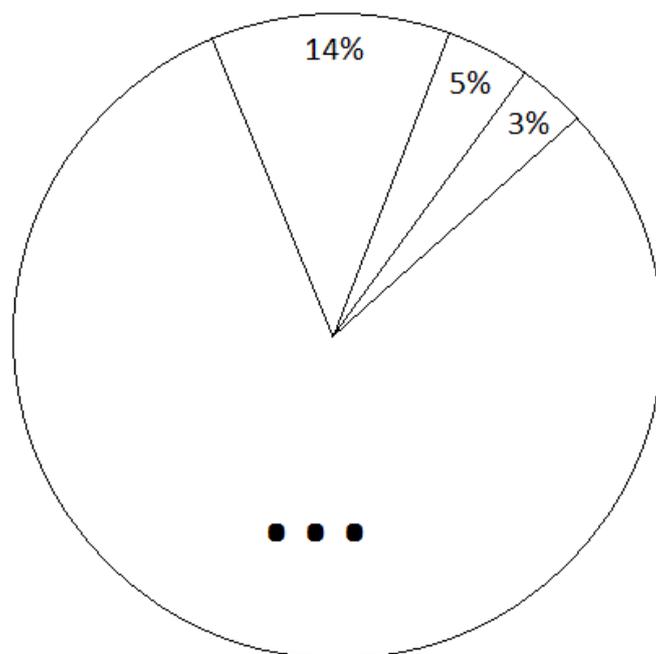


Рис. 4. Иллюстрация к методу колеса рулетки

Далее генерируется псевдослучайное число. В промежуточное поколение для скрещивания и мутации попадает та особь, в чей интервал попало это число. Псевдослучайные числа генерируются необходимое для заполнения промежуточного поколения число раз. Таким образом, получается, что в промежуточное поколение попадет несколько копий особей обладающих хорошими значениями функции приспособленности и, возможно, ни одной особи с плохим значением функции приспособленности. На примере этого классического оператора отбора доказывалась теорема схем в главе 1.

2.3.2. Турнирный оператор отбора

Данный оператор отбора состоит в том, что N (в общем случае не обязательно) раз, где N – число особей в поколении, случайным образом отбирается пара особей (в общем случае произвольное число особей). Затем сравниваются значения их функций приспособленности (происходит «турнир»). В промежуточное поколение для скрещивания и мутации попадает особь (в общем случае несколько особей) с большим значением функции приспособленности.

2.3.3. Ранговый оператор отбора

В ранговом операторе отбора заранее задана функция, отображающая значение функции приспособленности особи в число копий этой особи, попадающих в промежуточное поколение для скрещивания и мутации. Данная функция может быть как линейной, так и не быть.

В работе предлагается получить число k , равное частному размера поколения и числа 50. Затем в промежуточное поколение необходимо отобрать k копий лучшей, первой, особи, $k - 1$ копию второй особи, $k - i$ копий $i - 1$ особи. Если $k - i$ меньше единицы, то необходимо перенести в промежуточное поколение одну копию $k - i$ -й особи. Так действовать, пока не будет отобрано необходимое число особей (в данном случае в промежуточном поколении столько же особей, сколько и в предыдущем поколении). Худшие особи в промежуточное поколение не попадут за счет того, что лучшие были отобраны в нескольких экземплярах.

2.4. Операторы скрещивания

Для многоэтапного генетического программирования в работе используется несколько операторов скрещивания. Опишем каждый из них.

2.4.1. k -точечный оператор скрещивания

Это обобщенный вариант классического одноточечного оператора скрещивания, в котором две особи обмениваются половинами своих хромосом.

В k -точечном операторе скрещивания случайным образом в двух хромосомах выбирается k точек скрещивания. Первый кусок хромосомы (от начала, до первой из k точек скрещивания) остается у особей, вторым куском хромосомы (от первой до второй точки скрещивания) особи обмениваются, третий – остается, и т. д. В случае k точек скрещивания особи обмениваются k из $2k$ кусками своих хромосом.

2.4.2. Оператор скрещивания « k -процентное перемешивание»

В данном операторе скрещиваемые две особи обмениваются кусками своих хромосом длиной по k процентов от общей длины хромосомы. Такой обмен может происходить большое число раз. Например, столько раз, сколько процентов из 100 выбрано в параметре оператора для определения длины участков хромосом для обмена.

2.5. Оператор мутации

В работе используется оператор мутации, в котором вначале особь начинает мутировать с заданной вероятностью. Затем, если особь таким образом выбрана, с заданной вероятностью на случайную заменяется номер начального состояния автомата. Затем алгоритм проходит по всем состояниям и в каждом состоянии по всем входным воздействиям. При этом с заданной вероятностью на случайные заменяются выходное воздействие и номер следующего состояния, которые являются ответом на входное воздействие.

Выводы по главе 2

В этой главе была рассмотрена использованная в работе структура для представления конечного автомата типа Мили. Показано, как в наших исследованиях вычислялась функция приспособленности, какие операторы отбора, скрещивания и мутация использовались в работе.

Глава 3. Эксперименты

Опишем вначале инструментальное средство, написанное для изучения многоэтапного генетического программирования, затем приведем основные результаты, полученные с его помощью.

3.1. Инструментальное средство

В инструментальном средстве реализовано генетическое программирование, визуализация значения функции приспособленности лучшей особи в поколении, визуализация на поле задачи «Об умном муравье-3» поведения лучшей полученной особи и многое другое.

3.1.1. Главное окно

В главном окне программы, озаглавленной «Многоэтапное генетическое программирование» (рис. 5), располагается график значения функции приспособленности лучшей особи в поколении, в зависимости от поколения, строящийся в режиме реального времени.

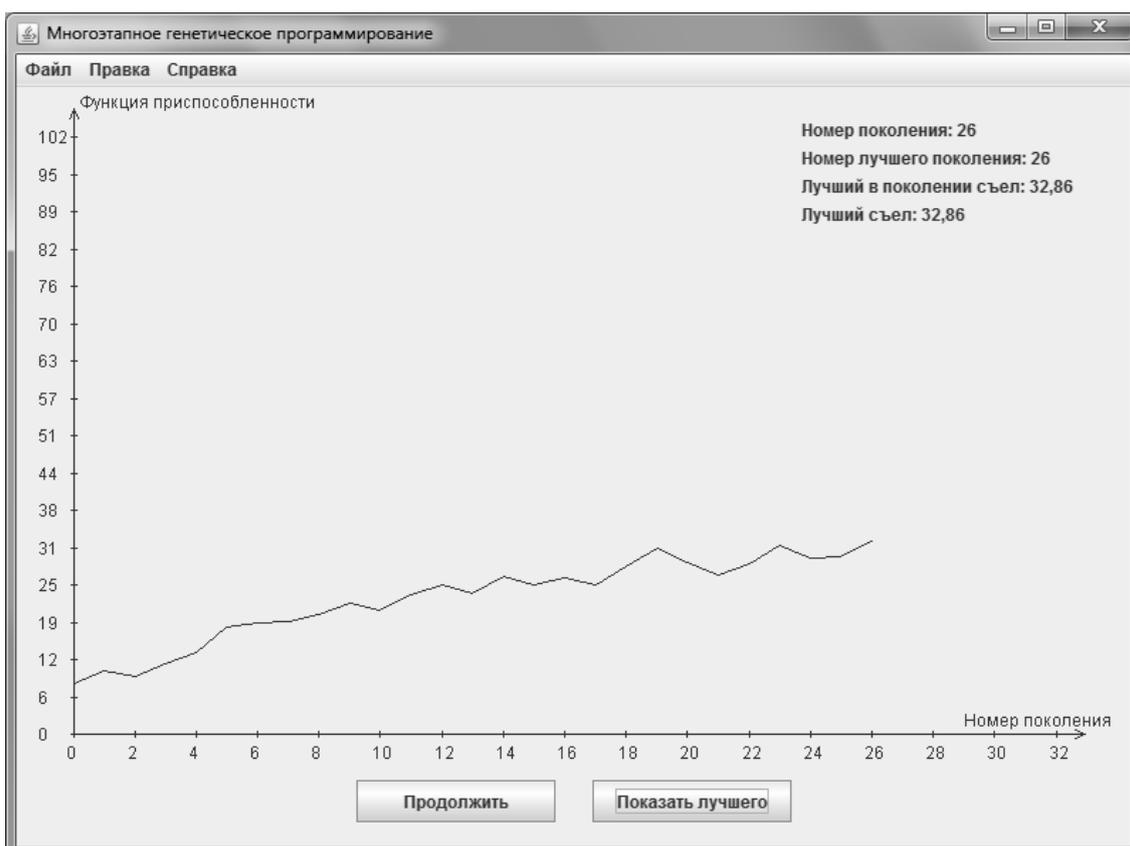


Рис. 5. Главное окно программы

В правом верхнем углу окна расположено четыре строки. В первой пишется номер текущего поколения, во второй – номер поколения, на котором получен лучший к данному моменту автомат, в третьей пишется, сколько яблок съела лучшая особь в текущем поколении, и, наконец, в четвертой строке отмечается, сколько яблок съела лучшая на данный момент особь.

Внизу окна расположены две кнопки. Левая – для запуска, приостановки генетического алгоритма, выращивающего лучший автомат, а также для запуска алгоритма после приостановки. Соответственно, в зависимости от того приостановлен алгоритм или нет, изменяются надписи на этой кнопке (Начать, Приостановить, Продолжить). Правая кнопка Показать лучшего предназначена для открытия нового окна, в котором визуализируется на поле из задачи «Об умном муравье-3» поведение лучшей полученной на текущий момент особи.

В верхней части окна расположено многофункциональное меню. В пункте меню Файл можно при остановленном алгоритме сохранить текущее поколение, загрузить ранее сохраненное поколение, выйти из приложения.

Также в пункте меню Файл можно сохранить лучший полученный на данный момент в эксперименте автомат в понятном человеку виде (Листинг 1). В первой строке файла записывается номер начального состояния автомата муравья. Затем идет четыре столбца. В первом столбце пишется номер состояния (от нуля и далее), во втором столбце номер входного воздействия, в третьем – выходное воздействие (L – повернуть налево, R – повернуть направо, F – пойти прямо), в четвертом столбце – номер состояния автомата, в которое надо перейти. Первые несколько строк файла приведены ниже.

Листинг 1

```
initialState: 0
stateNumber inputNumber action nextStateNumber
0
  0 F 1
  1 F 2
  2 F 1
  3 R 0
  4 F 2
  5 F 1
  6 F 1
  7 R 2
  8 F 1
  9 F 0
  ...
```

В следующем пункте меню – Правка, можно поменять, выбрав пункт Изменить конфигурацию..., все основные параметры эксперимента в новом всплывающем окне (разд. 3.1.2.). Также можно изменить язык всего интерфейса (русский или английский).

В следующем пункте меню – Справка, можно прочитать про создателя инструментального средства.

3.1.2. Окно «Изменение конфигурации»

При выборе пункта Изменить конфигурацию... в пункте меню Правка во всплывающем окне можно изменить параметры эксперимента (табл. 1).

Таблица 1. Изменение конфигурации

Название поля в окне	Возможные значения
Ширина поля	от 10 до 100
Высота поля	от 10 до 100
Заполненность поля яблоками	от 0 до 1
Заполненность поля хорошими яблоками	от 0 до 1
Координаты X и Y начальной позиции муравья	в пределах размера поля

Название поля в окне	Возможные значения
Начальное направление движения муравья	Запад, Север, Восток, Юг
Число шагов в эксперименте	от 1 до 999
Размер поколения	от 10 до 999
Число состояний в автомате	от 1 до 50
Вероятность мутации	от 0 до 1
Доля элиты в поколении	от 0 до 1
Число попыток в эксперименте	от 1 до 999

Внизу окна расположены три кнопки (рис. 6). Первая (Сохранить) предназначена для установления в эксперименте новых заданных пользователем параметров. При этом пользователю во всплывающем окне предлагается либо продолжить эксперимент над уже полученным к текущему моменту поколением либо начать заново.

Изменение конфигурации

Укажите новую конфигурацию эксперимента:

Ширина поля (от 10 до 100):

Высота поля (от 10 до 100):

Заполненность поля яблоками (от 0 до 1):

Заполненность поля хорошими яблоками (от 0 до 1):

Координаты начальной позиции муравья:

X

Y

Начальное направление движения муравья:

Число шагов в эксперименте (от 1 до 999):

Размер поколения (от 10 до 999):

Число состояний в автомате (от 1 до 50):

Вероятность мутации (от 0 до 1):

Доля элиты в поколении (от 0 до 1):

Число попыток в эксперименте (от 1 до 999):

Рис. 6. Окно «Изменение конфигурации»

Вторая кнопка (Отмена) закрывает окно без изменения параметров. Третья кнопка (Сбросить) устанавливает стандартные параметры,

приведенные в табл. 2.

Таблица 2. Стандартные параметры эксперимента

Название поля в окне	Значения
Ширина поля	32
Высота поля	32
Заполненность поля яблоками	0,3
Заполненность поля хорошими яблоками	0,3
Координаты X и Y начальной позиции муравья	16, 16
Начальное направление движения муравья	Восток
Число шагов в эксперименте	200
Размер поколения	200
Число состояний в автомате	5
Вероятность мутации	0,1
Доля элиты в поколении	0,1
Число попыток в эксперименте	5

Здесь следует указать, как реализовано изменение таких параметров как число состояний, заполненность поля хорошими яблоками, размер поколения.

При изменении числа состояний на большее число и желании пользователя продолжить эксперимент над ранее полученным поколением, у автомата генерируется недостающее число состояний со случайными значениями функции переходов и функции выходов автомата. При изменении числа состояний на меньшее число, из автомата удаляются состояния с большими номерами и, если начальное состояние было среди удаленных, то номер начального состояния заменяется случайным номером среди оставшихся состояний.

Если на поле расположены только обычные яблоки, то автомат имеет $2^8 = 256$ входных воздействий (в каждой из восьми видимых муравьем клеток, либо есть яблоко, либо его нет). Если же на поле имеются и обычные и хорошие яблоки, то число входных воздействий возрастает до $3^8 = 6561$. Таким образом, если пользователь хочет в ходе эксперимента

добавить хорошие яблоки на поле, то автомат расширяется до требуемого размера. При этом выходные воздействия, соответствующие случаю, когда муравей видит только обычные яблоки, остаются теми же. Аналогично, если на поле сначала присутствовали и обычные и хорошие яблоки, то после того, как пользователь уберет хорошие яблоки в окне Изменение конфигурации, автомат каждой особи в поколении уменьшится в размерах. В нем останутся выходные воздействия для ситуации, когда особь видит только обычные яблоки.

При изменении размера поколения из него либо удаляются худшие, ставшие лишними, особи, либо, при его увеличении, добавляются случайно сгенерированные новые особи.

3.1.3. Окно визуализации лучшей особи

В окне визуализации лучшей особи рисуется поле с яблоками и муравьями (рис. 7).

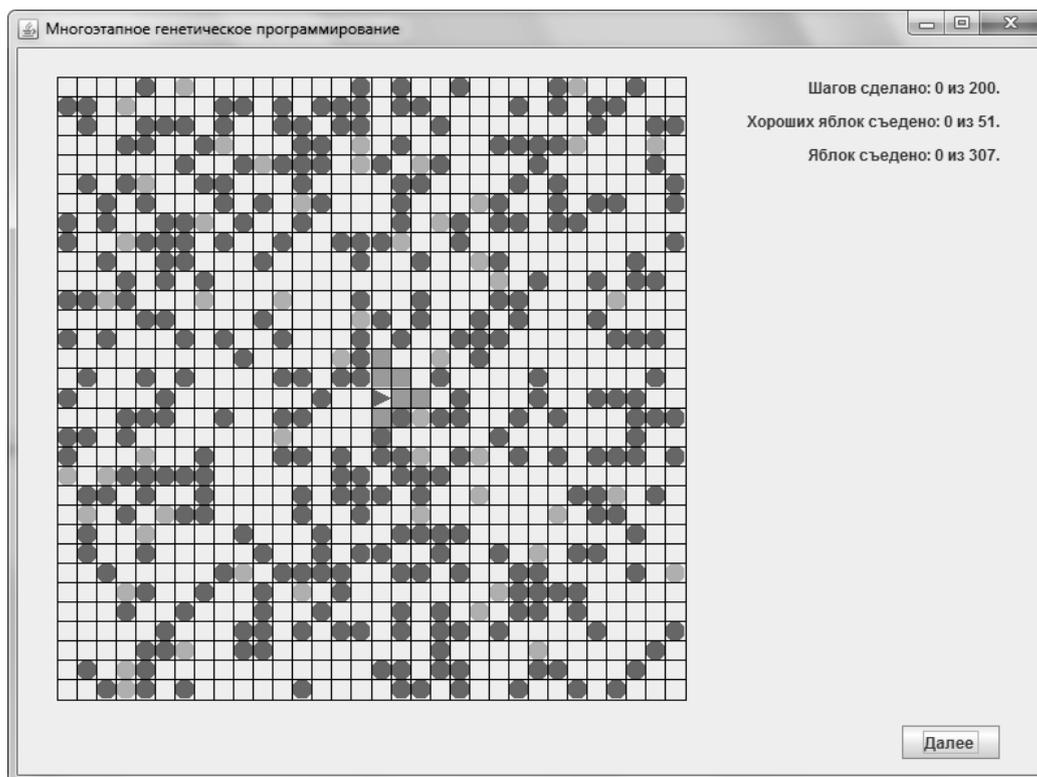


Рис. 7. Визуализация лучшей особи. Черный треугольник изображает муравья, темно-серые клетки – это те клетки, которые он видит. Темно-серые кружочки обозначают места расположения обычных яблок. Светло-серые кружочки обозначают места расположения хороших яблок

В нижнем правом углу окна расположена кнопка Далее, при

нажатию на которую визуализируемая особь делает ход.

В верхнем углу отображаются три строки. В первой пишется, сколько шагов сделал муравей, во второй – сколько съедено хороших яблок, в третьей – сколько обычных яблок съел муравей.

3.2. Результаты экспериментов

В работе была проведена серия экспериментов по изучению многоэтапного генетического программирования. Прежде всего, необходимо было выяснить, как хорошо генетическое программирование решает модифицированную задачу «Об умном муравье-3» (с хорошими яблоками и без). Затем предстояло узнать, насколько полезным является многоэтапное генетическое программирование.

3.2.1. Жадная стратегия

Модифицированную задачу «Об умном муравье-3» автор сначала попытался решить при помощи жадного алгоритма. Напомним, что в задаче муравей видит восемь клеток на поле: две спереди, две слева, две сзади и по одной спереди по диагонали (см. рис. 2).

Для того чтобы за 200 отведенных ходов съесть как можно больше яблок на поле размером 32 на 32, муравью необходимо как можно меньше поворачиваться. Поэтому жадный алгоритм построен таким образом:

1. Если цена яблок спереди больше или такая же, как слева или справа, то пойти вперед.
2. Иначе если цена яблок слева больше или такая же, как справа, то повернуть налево.
3. Иначе если слева первое яблоко с большей ценой ближе или на таком же расстоянии, как первое яблоко с такой же ценой справа, то повернуть налево.
4. Иначе если справа есть хоть одно яблоко, повернуть направо.
5. Иначе пойти вперед.

Муравей, управляемый этим алгоритмом, съедает в среднем на 100

случайных полях 45,22 – 48,33 обычных яблок при заполненности поля 0,1 (на поле нет хороших яблок). При увеличении числа полей до 1 000 000 (миллиона) это число приближается к 46,56. Если полей меньше 100, то среднее число в зависимости от запуска варьируется от 39,6 до 51,5.

Если поле заполнено обычными яблоками на 0,3, то, в среднем, на 50 полях особь съедает 89,8 – 95,2 яблок. При увеличении числа полей до 1 000 000 (миллиона) это число приближается к 92,79.

Если поле на 0,05 заполнено хорошими яблоками и на 0,3 обычными, то на 50 полях особь съедает 197 – 228 яблок, При увеличении числа полей до 1 000 000 (миллиона) это число приближается к 213,00.

На основе этих результатов было решено выращивать особи генетическим программированием до 500-го поколения, так как при большем числе поколений в большинстве из последующих экспериментов выращенные автоматы имеют значения функции приспособленности такие же или выше, чем у жадного, и генетический алгоритм уже сходится к оптимуму. Различия между разными подходами к выращиванию не будут сильно заметны, а время на эксперименты возрастет.

3.2.2. Выбор лучшего многоточечного оператора скрещивания

Необходимо хотя бы примерно узнать какой оператор скрещивания является лучшим для задачи «Об умном муравье-3». Выращивать особи будем до 100-го поколения. Исследуем k -точечный оператор скрещивания с числом точек скрещивания до 100. Пусть настроены следующие параметры эксперимента (табл. 3).

Таблица 3. Параметры первого эксперимента для выбора лучшего оператора скрещивания

Название поля в окне	Значения
Ширина поля	32
Высота поля	32
Заполненность поля яблоками	0,1
Заполненность поля хорошими яблоками	0

Название поля в окне	Значения
Координаты X и Y начальной позиции муравья	16, 16
Начальное направление движения муравья	Восток
Число шагов в эксперименте	200
Размер поколения	200
Число состояний в автомате	1
Вероятность мутации	0,1
Доля элиты в поколении	0,1
Число попыток в эксперименте	100

Число точек в многоточечном операторе скрещивания обозначим N . Значение лучшей к 100-му поколению функции приспособленности среди особей обозначим F . Номер поколения, на котором было получено F , обозначим k . Будем использовать оператор отбора колеса рулетки.

В табл. 4 приведены средние результаты за три опыта (пока цель получить примерные результаты – поэтому число экспериментов мало).

Таблица 4. Результаты первого эксперимента для выбора оператора скрещивания (N – число точек скрещивания, F – значение лучшей к 100-му поколению функции приспособленности среди особей, k – номер поколения, на котором было получено F)

N	0	1	2	3	4	5	10	15	20
K	98	86	95	100	98	99	88	89	100
F	38,0	41,1	40,2	40,2	39,7	39,3	43,5	41,3	42,9
N	25	30	35	40	45	50	55	60	65
K	87	80	98	96	85	91	94	88	83
F	42,5	41,6	44,2	43,4	44,1	42,8	43,7	44,4	44,3
N	70	75	80	85	90	95	100		
K	94	92	92	94	86	93	94		
F	43,7	44,7	43,9	45,1	44,8	43,9	44,2		

Из приведенных результатов видно, что для данной задачи наилучшим из рассмотренных, скорее всего, является оператор

85-точечного скрещивания (45,1), хотя результаты со значениями 43 и более находятся в пределах погрешности. Наихудший результат достигнут без использования оператора скрещивания, эволюция происходила только за счет 10% вероятности мутации. Результаты одно- и двухточечных операторов скрещивания также далеки от лучших.

Результаты опытов являются хоть и статистически слабым эмпирическим подтверждением гипотезы 1 из главы 1: охват лидирующих схем не более $\frac{1}{85}$ длины хромосомы. Также результаты данных опытов ставят под сомнение гипотезу 2 из главы 1, высказанную в работе [9]. Статистически более надежные опыты, ставящие под сомнение гипотезу 2, следуют далее.

При 100 попытках на поле муравей с выращенным к 500-му поколению автоматом в среднем съедает 48,34 яблок. Таким образом, эристика типа жадного алгоритма решает задачу «Об умном муравье-3» не лучше, чем генетическое программирование.

Теперь перейдем к эксперименту с большим числом опытов для каждого оператора скрещивания. Рассмотрим следующую конфигурацию (табл. 5).

Таблица 5. Конфигурация второго эксперимента для выбора лучшего оператора скрещивания

Название поля в окне	Значения
Ширина поля	32
Высота поля	32
Заполненность поля яблоками	0,3
Заполненность поля хорошими яблоками	0
Координаты X и Y начальной позиции муравья	16, 16
Начальное направление движения муравья	Восток
Число шагов в эксперименте	200
Размер поколения	200
Число состояний в автомате	3

Название поля в окне	Значения
Вероятность мутации	0,1
Доля элиты в поколении	0,1
Число попыток в эксперименте	50

Здесь и во всех последующих экспериментах этого раздела число точек в многоточечном операторе скрещивания обозначим N , число процентов в операторе « k -процентное смешивание» обозначим Mix . Значение лучшей к 500-му поколению функции приспособленности среди особей F , номер поколения, на котором достигнут результат – k . Будем использовать оператор отбора колеса рулетки. В табл. 6 приведены результаты 10 опытов.

Таблица 6. Результаты опытов для выбора лучшего оператора скрещивания (N – число точек в многоточечном операторе скрещивания, Mix – число процентов в операторе « k -процентное смешивание», F – значение лучшей к 500-му поколению функции приспособленности среди особей, k – номер поколения, на котором достигнут результат)

N	0				
Попытка	1	2	3	4	5
k	451	480	489	485	415
F	30,05	27,33	34,06	29,54	26,02
Попытка	6	7	8	9	10
k	492	394	373	500	495
F	37,96	37,90	35,15	25,98	29,86
N	1				
Попытка	1	2	3	4	5
k	431	472	489	490	487
F	58,03	61,20	57,76	52,62	51,67
Попытка	6	7	8	9	10
k	371	496	489	423	437
F	58,99	66,93	58,79	61,78	62,20

<i>N</i>	2				
Попытка	1	2	3	4	5
<i>k</i>	469	483	431	498	446
<i>F</i>	59,53	64,05	59,78	66,83	61,06
Попытка	6	7	8	9	10
<i>k</i>	435	457	436	433	486
<i>F</i>	58,20	68,29	54,95	55,81	62,73
<i>N</i>	10				
Попытка	1	2	3	4	5
<i>k</i>	490	498	479	492	476
<i>F</i>	83,22	85,63	80,08	83,33	80,92
Попытка	6	7	8	9	10
<i>k</i>	487	499	499	452	461
<i>F</i>	80,96	83,49	86,65	82,47	83,01

<i>N</i>	30				
----------	----	--	--	--	--

Попытка	1	2	3	4	5
<i>k</i>	475	452	430	426	491
<i>F</i>	83,11	85,01	84,25	83,91	84,17
Попытка	6	7	8	9	10
<i>k</i>	490	480	446	496	492
<i>F</i>	83,79	85,82	85,40	83,95	81,82

<i>N</i>	50				
Попытка	1	2	3	4	5
<i>k</i>	486	488	482	497	489
<i>F</i>	85,12	85,98	84,52	83,87	87,23
Попытка	6	7	8	9	10
<i>k</i>	493	486	467	483	491

<i>F</i>	85,72	85,70	85,68	85,82	82,43
<i>N</i>			70		
Попытка	1	2	3	4	5
<i>k</i>	498	431	490	495	486
<i>F</i>	85,22	84,94	86,83	86,93	83,39
Попытка	6	7	8	9	10
<i>k</i>	495	493	498	498	484
<i>F</i>	85,70	86,36	86,79	84,88	88,50
<i>N</i>			100		
Попытка	1	2	3	4	5
<i>k</i>	462	495	492	480	495
<i>F</i>	84,19	84,44	87,45	85,46	88,33
Попытка	6	7	8	9	10
<i>k</i>	478	483	463	449	437
<i>F</i>	87,29	86,63	88,23	87,49	86,49
<i>N</i>			1000		
Попытка	1	2	3	4	5
<i>k</i>	404	455	500	444	499
<i>F</i>	87,73	86,08	85,82	87,23	86,53
Попытка	6	7	8	9	10
<i>k</i>	423	391	482	500	473
<i>F</i>	87,23	88,56	87,27	87,09	87,09
<i>N</i>			1516		
Попытка	1	2	3	4	5
<i>k</i>	489	497	475	499	490
<i>F</i>	87,83	85,84	86,63	87,21	85,18
Попытка	6	7	8	9	10
<i>k</i>	491	458	498	479	467

<i>F</i>	86,59	87,13	87,15	87,33	85,66
<i>Mix</i>			1		
Попытка	1	2	3	4	5
<i>k</i>	493	488	486	487	441
<i>F</i>	85,33	84,55	84,11	84,46	86,77
Попытка	6	7	8	9	10
<i>k</i>	486	461	500	496	500
<i>F</i>	81,86	85,70	86,36	83,91	83,59

Из табл. 6 видно, что, чем лучше оператор скрещивания, тем особи с большим значением функции приспособленности выращиваются и тем меньший разброс в значениях в попытках опыта. Так, если в случае, когда оператор скрещивания не применяется, то не успевают получиться хороший автомат и разброс достигает 30%, а в случае с 1000-точечным оператором скрещивания довольно хороший автомат получен уже к 500-му поколению и разброс в попытках не превышает 5%. Ясно, что при большем числе поколений оба числа уменьшатся. В табл. 7 и на рис. 8 приведены усредненные за 10 попыток результаты.

Таблица 7. Усредненные за 10 попыток результаты второго эксперимента для выбора лучшего оператора скрещивания (*N* – число точек в многоточечном операторе скрещивания, *Mix* – число процентов в операторе «*k*-процентное смешивание», *F* – усредненное за 10 попыток значение лучшей к 500-му поколению функции приспособленности среди особей)

<i>N</i>	0	1	2	10	30	50	70	100	1000	1516
<i>F</i>	31,4	59,0	61,1	83,0	84,1	85,2	86,0	86,6	87,1	86,7
<i>Mix</i>	1									
<i>F</i>	84,7									

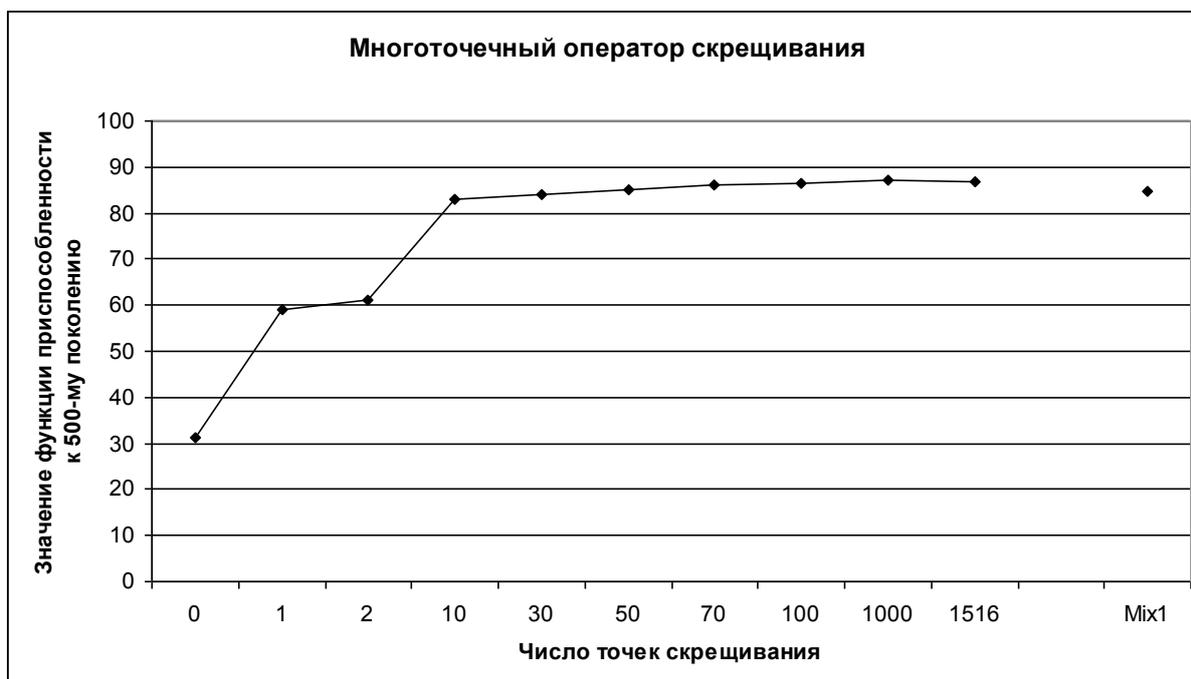


Рис. 8. Усредненные за 10 попыток результаты второго эксперимента для выбора лучшего оператора скрещивания

По результатам эксперимента получено несколько операторов скрещивания, обеспечивающих хороший результат. Выберем 1000-ти точечный оператор скрещивания. Он будет использоваться в дальнейших опытах. Эксперимент довольно хорошо исключает влияние вероятности на результат, так как в каждом опыте в каждом поколении значение функции приспособленности считалось на 50 случайных полях, а каждый опыт для каждого оператора скрещивания проводился 10 раз.

Результаты явно ставят под сомнение гипотезу 2 из главы 1 (одно- и двухточечные операторы скрещивания дали плохие результаты). Исходя из гипотезы 1, результат генетического алгоритма получается за счет комбинирования схем с приспособленностью выше среднего малого порядка и малого охвата. Поэтому оператор скрещивания с большим числом точек скрещивания дает большое разнообразие хромосом в поколении и тем самым ускоряет поиск оптимума, в то время как в генетическом алгоритме с малым числом точек скрещивания у оператора скрещивания такого разнообразия особей нет.

3.2.3. Выбор лучшего оператора отбора

Лучший оператор отбора выбирался среди трех операторов: колеса рулетки, турнирного (с выбором двух особей для турнира) и рангового, описанных в главе 2. Для скрещивания использовался 1000-точечный оператор из предыдущего раздела. Результаты представлены в табл. 8.

Таблица 8. Выбор лучшего оператора скрещивания (F – значение функции приспособленности, k – номер поколения, на котором был достигнут результат)

Оператор отбора	Колеса рулетки				
Попытка	1	2	3	4	5
K	404	455	500	444	499
F	87,73	86,08	85,82	87,23	86,53
Попытка	6	7	8	9	10
k	423	391	482	500	473
F	87,23	88,56	87,27	87,09	87,09
Оператор отбора	Турнирный				
Попытка	1	2	3	4	5
k	425	468	311	330	485
F	86,26	85,62	85,52	84,74	84,31
Попытка	6	7	8	9	10
k	476	387	450	491	452
F	83,35	84,11	82,08	81,56	82,49
Оператор отбора	Ранговый				
Попытка	1	2	3	4	5
k	491	490	471	493	475
F	82,06	81,14	84,33	83,49	84,19
Попытка	6	7	8	9	10
k	496	428	427	489	498
F	83,63	79,58	84,27	85,56	86,28

В табл.9 приведены средние результаты за 10 попыток (значения функции приспособленности).

Таблица 9. Результаты третьего эксперимента (F – среднее за 10 попыток значение

функции приспособленности)

Оператор отбора	Колеса рулетки	Турнирный	Ранговый
F	87,1	84,0	83,5

Функция в ранговом операторе оказалась не очень удачной. Операторы отбора колеса рулетки оказался лучше турнирного, скорее всего из-за того, что в операторе колеса рулетки остается чуть больше шансов у особей с плохим значением функции приспособленности попасть в промежуточное поколение и тем самым его разнообразить. Выберем для дальнейших экспериментов оператор колеса рулетки.

3.2.4. Многоэтапное генетическое программирование

Уже выбрав лучшие операторы скрещивания и отбора, попробуем ускорить получение результата.

3.2.4.1. Изменение оператора скрещивания

По теореме схем из главы 1 число хорошо приспособленных представителей одной схемы увеличивается экспоненциально. Поэтому имеет смысл при попадании генетического алгоритма в локальный оптимум на некоторое число поколений изменять оператор скрещивания, для того чтобы получить большее разнообразие особей.

В следующем опыте используется один из лучших полученных оператор скрещивания – 1000-точечный и оператор отбора колеса рулетки. В случае если в течение пяти поколений не будет получена особь с лучшим, чем у предыдущей, значением функции приспособленности, то на одно поколение оператор скрещивания будет замен на случайный из множества 0 – 1000 точечных и 0,1 – 10 (с шагом 0,1) процентных смешиваний операторов скрещивания. Результаты представлены в табл. 10.

Таблица 10. Опыты с изменением оператора скрещивания

Номер опыта	1	2	3	4	5
<i>K</i>	499	483	471	466	484
<i>F</i>	86,83	85,00	86,79	83,41	85,86
Номер опыта	6	7	8	9	10
<i>K</i>	419	479	469	499	476
<i>F</i>	83,75	87,77	84,86	89,30	86,85

Средний результат за 10 опытов – 86,0. Следовательно, в данной задаче предложенный метод ускорения получения результата ничего не улучшает.

3.2.4.2. Изменение числа яблок на поле

Будем выращивать автомат особи до 500-го поколения. При этом будем использовать 1000-точечный оператор скрещивания и оператор отбора колеса рулетки. Проведем несколько опытов:

1. Поле заполнено на 0,3 обычными яблоками.
2. Первые 250 поколений поле заполнено на 0,35 обычными яблоками, затем оставшиеся 250 поколений поле заполнено на 0,3 обычными яблоками.
3. Первые 250 поколений поле заполнено на 0,4 обычными яблоками, затем оставшиеся 250 поколений поле заполнено на 0,3 обычными яблоками.
4. Первые 250 поколений поле заполнено на 0,45 обычными яблоками, затем оставшиеся 250 поколений поле заполнено на 0,3 обычными яблоками.
5. Первые 250 поколений поле заполнено на 0,5 обычными яблоками, затем оставшиеся 250 поколений поле заполнено на 0,3 обычными яблоками.
6. Первые 250 поколений поле заполнено на 0,55 обычными яблоками, затем оставшиеся 250 поколений поле заполнено на 0,3 обычными яблоками.
7. Первые 250 поколений поле заполнено на 0,6 обычными

яблоками, затем оставшиеся 250 поколений поле заполнено на 0,3 обычными яблоками.

8. Первые 250 поколений поле заполнено на 0,7 обычными яблоками, затем оставшиеся 250 поколений поле заполнено на 0,3 обычными яблоками.

9. Первые 250 поколений поле заполнено на 0,8 обычными яблоками, затем оставшиеся 250 поколений поле заполнено на 0,3 обычными яблоками.

Затем сравним результаты к 500-му поколению. Обозначим номер поколения, на котором в среднем в данных опытах особь начинает съедать яблок больше, чем в одноэтапном эксперименте символом k . F – значение функции приспособленности. Результаты представлены в табл. 11.

Таблица 11. Многоэтапное генетическое программирование: эксперимент по увеличению числа яблок на поле (F – значение функции приспособленности, k – номер поколения, на котором в среднем в опытах особь начинает есть яблок больше, чем в одноэтапном эксперименте)

Номер опыта	1				
Попытка	1	2	3	4	5
k	404	455	500	444	499
F	87,73	86,08	85,82	87,23	86,53
Попытка	6	7	8	9	10
k	423	391	482	500	473
F	87,23	88,56	87,27	87,09	87,09
Номер опыта	2				
Попытка	1	2	3	4	5
k	382	–	454	393	304
F	87,49	86,91	87,71	88,31	88,60
Попытка	6	7	8	9	10
k	357	359	–	–	432
F	87,29	91,19	86,40	86,87	87,55

Номер опыта	3
-------------	---

Попытка	1	2	3	4	5
k	252	257	306	334	257
F	89,92	88,68	89,20	89,98	88,78
Попытка	6	7	8	9	10
k	342	251	328	390	411
F	87,97	88,98	87,23	87,23	86,41
Номер опыта	4				
Попытка	1	2	3	4	5
k	254	291	318	256	324
F	89,92	89,12	89,08	88,86	88,60
Попытка	6	7	8	9	10
k	259	355	325	382	337
F	89,94	89,90	87,51	88,31	87,97
Номер опыта	5				
Попытка	1	2	3	4	5
k	380	320	335	359	340
F	88,35	87,12	87,92	88,74	89,40
Попытка	6	7	8	9	10
k	311	272	332	309	315
F	90,20	89,60	87,11	90,59	89,30
Номер опыта	6				
Попытка	1	2	3	4	5
k	324	264	296	315	317
F	89,54	91,33	90,81	90,45	89,70
Попытка	6	7	8	9	10
k	376	323	271	299	326
F	89,32	89,54	88,60	89,52	88,86

Номер опыта	7
-------------	---

Попытка	1	2	3	4	5
k	325	334	339	304	321
F	88,21	90,18	89,02	89,76	89,30
Попытка	6	7	8	9	10
k	389	381	416	366	324
F	90,20	88,78	88,54	88,35	90,32
Номер опыта	8				
Попытка	1	2	3	4	5
k	363	–	464	432	–
F	89,36	86,85	87,77	87,41	86,36
Попытка	6	7	8	9	10
k	–	–	401	425	384
F	84,74	85,62	88,76	89,18	90,12
Номер опыта	9				
Попытка	1	2	3	4	5
k	–	–	–	–	–
F	86,95	86,49	86,53	87,83	85,24
Попытка	6	7	8	9	10
k	–	–	–	–	–
F	82,38	82,16	82,38	86,10	85,90

В табл. 12 приведены усредненные за 10 попыток результаты каждого опыта. Разброс значений в попытках не более чем в два раза больше, чем разница между соседними средними в табл. 12. Таким образом, выводы ниже можно считать достоверными.

Таблица 12. Результаты эксперимента по увеличению числа яблок на поле. (F – среднее значение функции приспособленности по результатам 10 попыток, k – усредненный

номер поколения, на котором был получен результат лучший или такой же, как у первого опыта)

Номер опыта	1	2	3	4	5
<i>K</i>	457	383	313	310	327
<i>F</i>	87,1	87,8	88,4	88,9	88,8
Номер опыта	6	7	8	9	
<i>K</i>	311	350	412	–	
<i>F</i>	89,8	89,3	87,6	85,2	

Результаты также представлены на рис. 9.

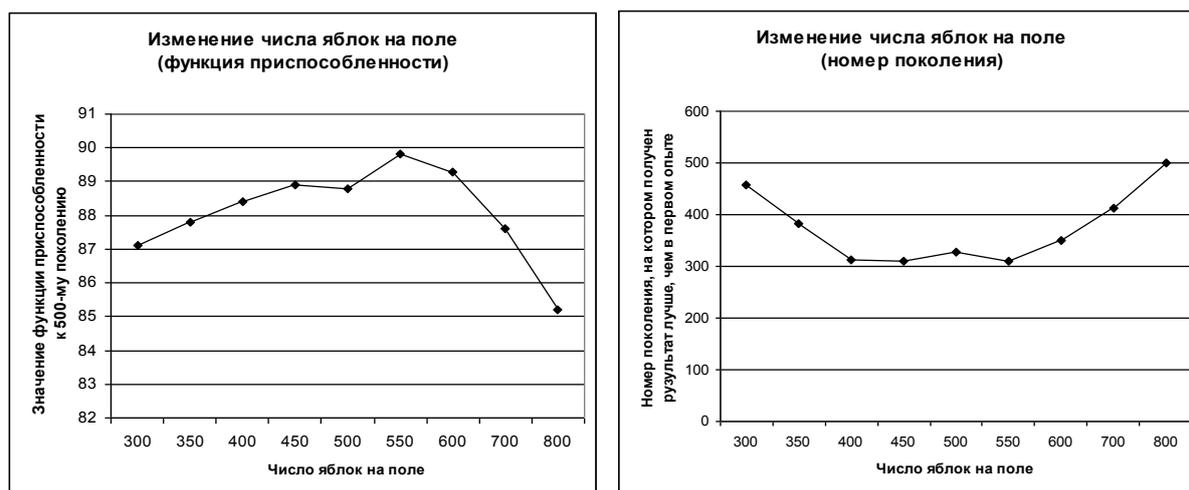


Рис. 9. Изменение числа яблок на поле. Слева – зависимость значения функции приспособленности к 500-му поколению от числа яблок на поле первые 250 поколений. Справа – зависимость усредненного номера поколения, на котором был получен результат лучший или такой же, как у первого опыта от числа яблок на поле первые 250 поколений

По результатам видно, что если необходимо вырастить автомат на поле с определенным числом объектов, то выгоднее при выращивании сначала в полтора-два раза увеличить число этих объектов, а затем уменьшить.

Это можно объяснить тем, что в случае многоэтапного выращивания автомата особь первые 250 поколений активно учится справляться с ситуациями, когда она видит довольно много яблок перед собой, затем же оставшиеся 250 поколений учится также в ситуации, когда она видит перед собой меньше яблок. В случае одноэтапного выращивания автомата, когда

все 500 поколений число яблок на поле не изменяется, особь с лучшим автоматом успевает лучше научиться действовать в ситуации, когда она видит немного яблок, но не успевает научиться хорошо реагировать на большое число яблок перед собой из-за довольно редкой встречаемости такой ситуации.

Один из лучших результатов показал шестой опыт, в котором первые 250 поколений поле заполнено на 0,55 обычными яблоками, затем оставшиеся 250 поколений поле заполнено на 0,3 обычными яблоками. Проведем еще несколько опытов для уточнения номера поколения, на котором следует изменять число яблок на поле (результаты представлены в табл. 13):

1. Первые 50 поколений поле заполнено на 0,55 обычными яблоками, затем оставшиеся 450 поколений поле заполнено на 0,3 обычными яблоками.
2. Первые 100 поколений поле заполнено на 0,55 обычными яблоками, затем оставшиеся 400 поколений поле заполнено на 0,3 обычными яблоками.
3. Первые 150 поколений поле заполнено на 0,55 обычными яблоками, затем оставшиеся 350 поколений поле заполнено на 0,3 обычными яблоками.
4. Первые 200 поколений поле заполнено на 0,55 обычными яблоками, затем оставшиеся 300 поколений поле заполнено на 0,3 обычными яблоками.
5. Первые 250 поколений поле заполнено на 0,55 обычными яблоками, затем оставшиеся 250 поколений поле заполнено на 0,3 обычными яблоками.
6. Первые 300 поколений поле заполнено на 0,55 обычными яблоками, затем оставшиеся 200 поколений поле заполнено на 0,3 обычными яблоками.
7. Первые 350 поколений поле заполнено на 0,55 обычными

яблоками, затем оставшиеся 150 поколений поле заполнено на 0,3 обычными яблоками.

8. Первые 400 поколений поле заполнено на 0,55 обычными яблоками, затем оставшиеся 100 поколений поле заполнено на 0,3 обычными яблоками.

9. Первые 450 поколений поле заполнено на 0,55 обычными яблоками, затем оставшиеся 50 поколений поле заполнено на 0,3 обычными яблоками.

Таблица 13. Многоэтапное генетическое программирование: эксперимент по уточнению номера поколения, на котором следует изменить число яблок на поле (F – значение функции приспособленности)

Номер опыта	1				
Попытка	1	2	3	4	5
F	88,17	86,55	88,44	89,10	89,28
Попытка	6	7	8	9	10
F	88,01	86,81	87,59	88,90	87,31
Номер опыта	2				
Попытка	1	2	3	4	5
F	89,88	89,12	88,66	88,78	88,31
Попытка	6	7	8	9	10
F	89,28	89,22	89,40	89,36	88,48
Номер опыта	3				
Попытка	1	2	3	4	5
F	90,67	89,72	89,92	88,62	89,88
Попытка	6	7	8	9	10
F	87,65	89,12	90,22	90,75	89,60
Номер опыта	4				
Попытка	1	2	3	4	5
F	88,13	89,02	89,88	89,16	89,02
Попытка	6	7	8	9	10
F	89,58	90,97	89,70	89,40	90,45
Номер опыта	5				

Попытка	1	2	3	4	5
<i>F</i>	89,54	91,33	90,81	90,45	89,70
Попытка	6	7	8	9	10
<i>F</i>	89,32	89,54	88,60	89,52	88,86
Номер опыта	6				
Попытка	1	2	3	4	5
<i>F</i>	89,02	88,31	89,76	90,24	89,86
Попытка	6	7	8	9	10
<i>F</i>	89,10	90,67	89,30	89,96	89,48
Номер опыта	7				
Попытка	1	2	3	4	5
<i>F</i>	90,04	88,33	88,15	90,02	90,30
Попытка	6	7	8	9	10
<i>F</i>	89,28	89,74	89,80	89,14	87,41
Номер опыта	8				
Попытка	1	2	3	4	5
<i>F</i>	89,72	89,54	89,74	85,42	88,35
Попытка	6	7	8	9	10
<i>F</i>	88,41	89,38	88,60	88,56	89,34
Номер опыта	9				
Попытка	1	2	3	4	5
<i>F</i>	86,87	89,34	89,38	86,67	89,22
Попытка	6	7	8	9	10
<i>F</i>	87,59	86,59	87,93	86,55	85,68

В табл. 14 и на рис. 10 приведены усредненные за 10 попыток результаты.

Таблица 14. Результаты эксперимента по уточнению номера поколения, на котором следует изменить число яблок на поле (*F* – усредненное за 10 попыток значение функции приспособленности)

Номер опыта	1	2	3	4	5
<i>F</i>	88,0	89,0	89,6	89,5	89,8
Номер опыта	6	7	8	9	

<i>F</i>	89,6	89,2	88,7	87,6	
----------	------	------	------	------	--



Рис. 10. Результаты эксперимента по уточнению номера поколения, на котором следует изменить число яблок на поле. На графике представлена зависимость значения функции приспособленности от номера поколения, на котором меняется число яблок на поле

По результатам эксперимента видно, что лучше всего изменять число яблок на 150 – 350 поколениях из 500 – по прошествии 30% – 70% процентов от запланированного числа поколений. Разброс значений в опытах три-четыре процента.

Попробуем выращивать лучший автомат, изменяя число яблок на поле более чем один раз. Проведем еще один опыт (табл. 15): пусть поле первые 100 поколений заполнено на 0,6 обычными яблоками (именно при такой заполненности поля был получен один из лучших результатов в предыдущих опытах, см. табл. 12), следующие 100 поколений на 0,5, затем 100 поколений на 0,4 и последний 200 поколений на 0,3.

Таблица 15. Троекратное изменение числа яблок на поле (*F* – значение функции приспособленности, *k* – номер поколения, на котором особь начинает есть яблок больше, чем в одноэтапном эксперименте)

Попытка	1	2	3	4	5
k	321	303	369	404	398
F	89,60	89,78	87,71	88,19	88,74
Попытка	6	7	8	9	10
k	301	303	304	312	347
F	88,82	89,38	89,74	89,72	88,56

Усредненный за 10 попыток результат – 89,0 – аналогичен увеличению числа яблок в полтора раза до 250 поколения (опыты 4, 5 в табл. 12). Не лучший результат получен, скорее всего, потому, что 100 поколений слишком маленькое число для того, чтобы успел быть выращен хороший автомат на поле с определенным числом яблок.

3.2.4.3. Добавление «хороших» яблок на поле

Введем в эксперименты хорошие яблоки. Пусть обычные яблоки, использованные в предыдущих опытах, будут иметь цену единица. Тогда хорошие яблоки имеют цену 10. Сравним на следующих опытах, что выгоднее в условиях, когда обычных яблок намного больше, чем хороших: обучать муравья на поле, на котором уже есть и обычные и хорошие яблоки, или сначала обучить особь есть только обычные яблоки, а затем добавить хорошие (опыты 1 – 3). И наоборот, если на поле намного больше хороших яблок, чем обычных, что выгоднее: обучать муравья в ситуации, когда на поле имеются оба типа яблок, или сначала обучить муравья на хороших яблоках, а затем добавить обычные (опыты 4 – 6). Напомним, что цена хороших яблок в 10 раз больше, чем обычных.

Проведем несколько опытов:

1. Поле заполнено на 0,3 обычными яблоками и на 0,05 хорошими яблоками.
2. Поле заполнено первые 250 поколений на 0,3 обычными яблоками, затем 250 поколений поле заполнено на 0,3 обычными яблоками и на 0,05 хорошими яблоками.
3. Поле заполнено первые 250 поколений на 0,4 обычными

яблоками, затем 250 поколений поле заполнено на 0,3 обычными яблоками и на 0,05 хорошими яблоками.

4. Поле заполнено на 0,3 хорошими яблоками и на 0,05 обычными яблоками.
5. Поле заполнено первые 250 поколений на 0,3 хорошими яблоками, затем 250 поколений поле заполнено на 0,3 хорошими яблоками и на 0,05 обычными яблоками.
6. Поле заполнено первые 250 поколений на 0,4 хорошими яблоками, затем 250 поколений поле заполнено на 0,3 хорошими яблоками и на 0,05 обычными яблоками.

Результаты опытов представлены в табл. 16, 17 и на рис. 11. Как и ранее F – значение функции приспособленности, k – номер поколения, на котором был получен результат лучше, чем в первом опыте для опытов два и три, и лучше чем в четвертом опыте для опытов пять и шесть.

Таблица 16. Многоэтапное генетическое программирование: эксперимент с хорошими яблоками (F – значение функции приспособленности, k – номер поколения, на котором был получен результат лучше, чем в первом опыте для опытов два и три, и лучше чем в четвертом опыте для опытов пять и шесть)

Номер опыта	1				
Попытка	1	2	3	4	5
K	392	454	489	489	495

<i>F</i>	88,6	90,1	106,3	96,4	102,9
Попытка	6	7	8	9	10
<i>K</i>	472	488	420	442	383
<i>F</i>	100,2	113,2	110,7	110,5	115,4

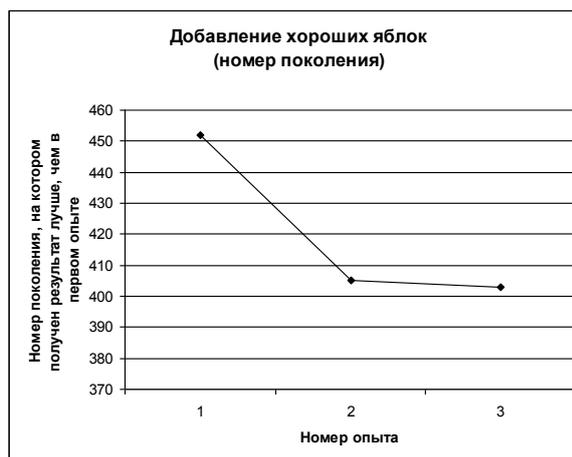
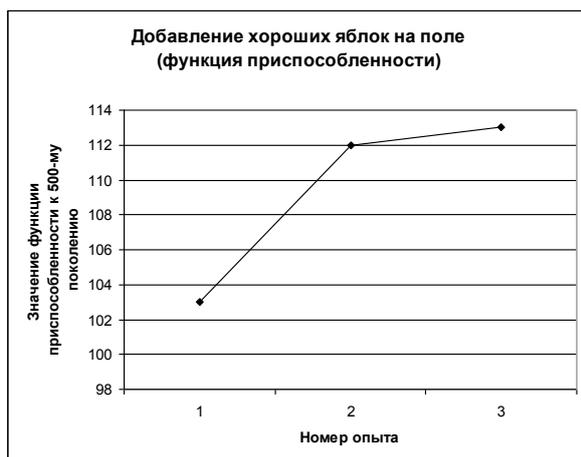
Номер опыта	2				
Попытка	1	2	3	4	5
<i>K</i>	321	335	394	366	341
<i>F</i>	102,5	103,0	111,0	125,1	124,5
Попытка	6	7	8	9	10
<i>k</i>	471	447	446	428	500
<i>F</i>	108,9	108,0	111,5	115,4	107,4
Номер опыта	3				
Попытка	1	2	3	4	5
<i>k</i>	311	476	366	446	345
<i>F</i>	98,5	120,4	117,2	108,2	118,4
Попытка	6	7	8	9	10
<i>k</i>	431	425	491	348	393
<i>F</i>	112,1	113,6	104,3	120,4	120,1
Номер опыта	4				
Попытка	1	2	3	4	5
<i>k</i>	473	475	492	482	465
<i>F</i>	424,4	382,4	408,9	437,5	459,5
Попытка	6	7	8	9	10
<i>k</i>	471	437	482	409	500
<i>F</i>	430,1	409,5	411,6	406,7	419,6
Номер опыта	5				
Попытка	1	2	3	4	5
<i>k</i>	347	358	328	430	407
<i>F</i>	493,6	487,8	487,2	437,0	495,6
Попытка	6	7	8	9	10
<i>k</i>	413	374	378	407	335
<i>F</i>	456,6	469,4	522,7	448,0	496,9

Номер опыта	6				
Попытка	1	2	3	4	5
k	418	399	352	322	394
F	457,3	490,2	481,1	517,5	488,5
Попытка	6	7	8	9	10
k	395	344	402	388	367
F	487,8	495,6	445,2	486,7	504,7

Разброс значений в попытках не более чем в два раза больше, чем разница между соседними средними в табл. 17. Таким образом, выводы ниже можно считать достоверными.

Таблица 17. Многоэтапное генетическое программирование: усредненные результаты эксперимента с хорошими яблоками. С хорошими яблоками (F – среднее значение функции приспособленности по результатам 10 попыток, k – усредненный номер поколения)

Номер опыта	1	2	3	4	5	6
k	452	405	403	432	378	378
F	103	112	113	435	480	486



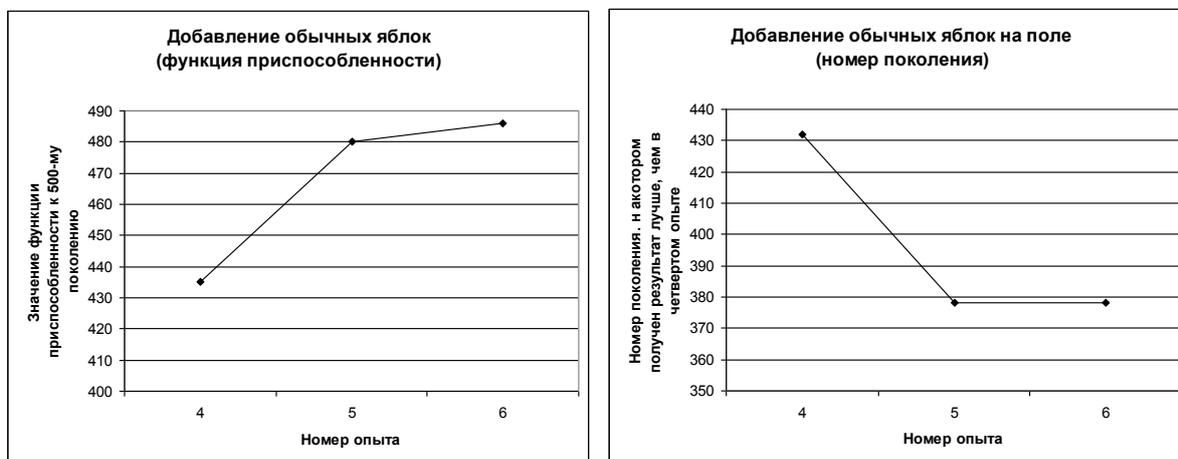


Рис. 11. Усредненные результаты эксперимента с хорошими яблоками. Наверху слева – график зависимости значения функции приспособленности от номера опыта. Наверху справа – график зависимости номера поколения, на котором получен результат лучше, чем в первом опыте от номера опыта. Внизу слева график зависимости значения функции приспособленности от номера опыта. Внизу справа – график зависимости номера поколения, на котором получен результат лучше, чем в четвертом опыте от номера опыта.

По результатам эксперимента видно, что в случае, когда на поле яблок одного типа намного больше, чем другого, намного выгоднее сначала вырастить автомат, реагирующий только на один тип яблок, а затем добавить другой тип. Это можно объяснить тем, что в случае, когда особь видит яблоки только одного типа, ее автомат имеет $2^8 = 256$ входных воздействий, и его представление в виде байтовой строки не очень длинное. В случае же, когда на поле имеются и обычные и хорошие яблоки автомат особи имеет $3^8 = 6561$ входное воздействие, и его представление в виде байтовой строки более чем в 10 раз длиннее, а чем длиннее изменяемая строка, тем дольше получается оптимальная строка. Для подтверждения этого в табл. 18, 19 приведены результаты 4 – 6 опытов для случая изначально большой строки представления автомата.

Таблица 18. Опыты 4 – 6 с изначально большой строкой представления автомата в

генетическом программировании (F – значение функции приспособленности, k – номер поколения на котором значение функции приспособленности стало лучше или таким же как в опыте 4)

Номер опыта	4				
Попытка	1	2	3	4	5
K	476	351	413	442	443
F	412,8	431,2	408,5	426,8	452,4
Попытка	6	7	8	9	10
K	485	339	490	407	473
F	440,1	439,8	479,3	432,6	428,9
Номер опыта	5				
Попытка	1	2	3	4	5
k	–	–	–	–	–
F	384,5	334,2	422,5	402,7	376,0
Попытка	6	7	8	9	10
k	–	–	–	–	–
F	375,0	339,2	401,1	387,7	369,4

Номер опыта	6				
Попытка	1	2	3	4	5
k	–	–	–	492	–
F	406,6	385,1	430,0	436,8	402,4
Попытка	6	7	8	9	10
k	–	–	–	–	–
F	359,9	377,9	392,9	339,8	338,3

Таблица 19. Средние значения F за 10 попыток для опытов 4 – 6 с изначально большой строкой представления автомата

Номер опыта	4	5	6
F	435	379	387

Выводы по главе 3

В этой главе было описано инструментальное средство, использовавшееся в экспериментах; проведенные опыты.

В главе рассматривалось два подхода к решению задачи «Об умном муравье-3»: жадная стратегия и генетическое программирование. Были отобраны лучшие операторы скрещивания и отбора. Была поставлена под сомнение гипотеза о числе точек скрещивания в операторе скрещивания.

В главе рассматривались различные подходы по ускорению получения результата генетическим программированием.

Заключение

В бакалаврской работе была изучена история эволюционных алгоритмов, рассмотрены теоретические основы генетических алгоритмов. Затем было реализовано приложение для изучения генетического программирования и многоэтапного выращивания автоматов. В приложении имеется возможность изменять любые параметры экспериментов.

Важным результатом является то, что лучшими операторами скрещивания для задачи «Об умном муравье-3» оказались операторы с большим числом точек скрещивания. Это ставит под сомнение гипотезу о том, что оператор с наименьшим числом точек скрещивания является лучшим для генетических алгоритмов.

Были рассмотрены несколько случаев увеличения скорости получения требуемого автомата при помощи многоэтапного генетического программирования. Была рассмотрена задача, в которой особь, управляемая автоматом, видит объекты на поле. Экспериментально получено, что если при обучении на некоторое время увеличить число объектов по сравнению с их числом, определенном в условии задачи, то это ускорит получение наиболее приспособленной особи.

При наличии нескольких объектов различного типа, которые могут быть видны особи, управляемой автоматом, и при превалировании по числу одних объектов над другими, выгодно выращивать автомат многоэтапно, постепенно добавляя новые типы объектов.

ИСТОЧНИКИ

1. *Barricelli, Nils Aall.* Esempi numerici di processi di evoluzione / *Methodos*. 1954, pp. 45 – 68.
2. *Fraser, Alex S.* Simulation of Genetic Systems by Automatic Digital Computers. I. Introduction / *Australian Journal of Biological Sciences* 10. 1957, pp. 484 – 491.
3. *Lawrence J. Fogel.* *Biotechnology: Concepts and Applications*. Prentice Hall. 1963.
4. *Ingo Rechenberg.* *Evolutionsstrategie – Optimierung technischer Systeme nach Prinzipien der biologischen Evolution* (PhD thesis). 1971.
5. *Hans-Paul Schwefel.* *Cybernetic Evolution as Strategy for Experimental Research in Fluid Mechanics* (in German). Diploma Thesis. Hermann Föttinger-Institute for Fluid Mechanics, Technical University of Berlin, March 1965.
6. *Holland, John H.* *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor. 1975.
7. *Koza, John.* *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press. 1992.
8. *Рутковская Д., Пулинский М., Рутковский Л.* *Нейронные сети, генетические алгоритмы и нечеткие системы*. М.: Горячая линия-Телеком. 2008.
9. *Емельянов В. В., Курейчик В. В., Курейчик В. М.* *Теория и практика эволюционного моделирования*. М.: ФИЗМАТЛИТ, 2003.
10. *Поликарпова Н. И., Шалыто А. А.* *Автоматное программирование*. СПб.: Питер, 2009. <http://is.ifmo.ru/books/book.pdf>
11. «Википедия» – интернет-энциклопедия / Конечный автомат. http://ru.wikipedia.org/wiki/Конечный_автомат
12. «Википедия» – интернет-энциклопедия / Классификация абстрактных автоматов. http://ru.wikipedia.org/wiki/Классификация_абстрактных_автоматов
13. *Бедный Ю. Д., Шалыто А. А.* *Применение генетических алгоритмов для построения автоматов в задаче «Умный муравей»*. СПбГУ ИТМО, 2007. <http://is.ifmo.ru/works/ant>