

На правах рукописи

Лобанов Павел Геннадьевич

**Использование генетических алгоритмов для генерации  
конечных автоматов**

Специальность 05.13.11. Математическое и программное обеспечение вычислительных  
машин, комплексов и компьютерных сетей

АВТОРЕФЕРАТ

диссертации на соискание ученой степени  
кандидата технических наук

Санкт-Петербург

2008

Работа выполнена в Санкт-Петербургском государственном университете информационных технологий, механики и оптики

Научный руководитель: доктор технических наук, профессор  
Шалыто Анатолий Абрамович

Официальные оппоненты: доктор технических наук, профессор  
Тимофеев Адиль Васильевич

кандидат технических наук  
Шопырин Данил Геннадьевич

Ведущая организация: Санкт-Петербургский государственный  
университет аэрокосмического  
приборостроения

Защита диссертации состоится «22» октября 2008 года в 16 часов на заседании диссертационного совета Д.212.227.06 в Санкт-Петербургском государственном университете информационных технологий, механики и оптики, 197101, Санкт-Петербург, Кронверкский проспект, д. 49, СПбГУ ИТМО.

С диссертацией можно ознакомиться в библиотеке СПбГУ ИТМО.

Автореферат разослан 19 сентября 2008 г.

Ученый секретарь  
совета Д.212.227.06,  
доктор технических наук,  
профессор

Тарлыков Владимир Алексеевич

### **Общая характеристика работы**

**Актуальность проблемы.** В последнее время все шире применяется автоматное программирование, в рамках которого поведение программ описывается с помощью конечных детерминированных автоматов (в дальнейшем автоматов).

Для многих задач автоматы удается строить эвристически, однако существуют задачи, для которых такое построение затруднительно или невозможно. Известны задачи (например, задача о флибах и задача об умном муравье), в которых применение генетических алгоритмов (направленного перебора) позволяет генерировать автоматы. Это повышает уровень автоматизации *автоматных программ* и является одним из первых шагов к *автоматическому построению программ*.

Генетические алгоритмы применяются при решении широкого круга задач. Однако при использовании классических генетических алгоритмов для генерации автоматов часто требуются большие вычислительные ресурсы, для того чтобы получить решение с необходимым уровнем качества.

Для повышения эффективности генетических алгоритмов требуется их модифицировать с учетом специфики задач, решаемых в диссертации.

Поэтому исследования, направленные на разработку более эффективных генетических алгоритмов для построения автоматов, весьма актуальны.

**Цель диссертационной работы** – разработка методов оптимизации генетических алгоритмов для построения автоматов.

**Основные задачи исследования** состоят в следующем:

1. Разработка новых операторов мутации, позволяющих повысить эффективность генетических алгоритмов, используемых для построения автоматов, представленных в виде графов переходов.
2. Разработка функций приспособленности для повышения быстродействия генетических алгоритмов при построении автоматов.
3. Разработка модификаций генетических алгоритмов при построении автоматов для ряда задач.
4. Исследование эффективности генетических алгоритмов при построении двух разнотипных автоматов для одной из рассмотренных задач.

**Методы исследования.** В работе использовались методы теории автоматов, программирования и генетические алгоритмы.

**Научная новизна.** В работе получены новые научные результаты, которые выносятся на защиту:

1. Разработаны два оператора мутации для автоматов, представленных в виде графов переходов: восстановление связей между состояниями и сортировка состояний в порядке использования.
2. Предложена функция приспособленности, учитывающая число используемых состояний в автомате, которая позволяет для рассмотренного класса задач повысить эффективность генетических алгоритмов для генерации автоматов.

3. Разработаны модификации генетических алгоритмов для задач о флибах и умном муравье и для построения автопилота для упрощенной модели вертолета.
4. При решении задачи о флибах выполнено сравнение эффективности генетических алгоритмов, которые позволяют генерировать автоматы Мили и автоматы Мили с флагами. При этом показано, что точность предсказания флибом, который моделируется автоматом с флагами, выше.

Результаты диссертации получены в ходе работ по государственному контракту «Технология генетического программирования для генерации автоматов управления системами со сложным поведением» (шифр «2007-4-14-18-01-033»), проводимых СПбГУ ИТМО в 2007 – 2008 гг. в рамках Федеральной целевой программы «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России» на 2007 – 2012 гг.

**Достоверность** научных положений, выводов и практических рекомендаций, полученных в диссертации, подтверждается корректным обоснованием постановок задач, точной формулировкой критериев, а также результатами экспериментов по использованию предложенных в диссертации методов.

**Практическое значение** работы заключается в том, что полученные результаты могут быть использованы для решения практических задач. Предложенные модификации генетических алгоритмов для построения автоматов, позволяют повысить эффективность работы алгоритмов, что подтверждается экспериментальными данными. Практическая ценность подтверждается внедрением результатов работы.

**Внедрение результатов работы.** Результаты, полученные в диссертации, используются на практике в компании *Транзас* (Санкт-Петербург) при разработке тренажера вертолета, а также в учебном процессе на кафедре «Компьютерные технологии» СПбГУ ИТМО по курсу «Теория автоматов в программировании».

**Апробация диссертации.** Основные положения диссертационной работы докладывались на 4-й Всероссийской научной конференции «Управление и информационные технологии» СПбГЭТУ «ЛЭТИ» (2006), XXXVI научной и учебно-методической конференции профессорско-преподавательского и научного состава СПбГУ ИТМО (2007), IV межвузовской конференции молодых ученых СПбГУ ИТМО (2007), XIV Всероссийской научно-методической конференции «Телематика-2007» (Санкт-Петербург), XXXVII научной и учебно-методической конференции СПбГУ ИТМО (2008), V Всероссийской межвузовской конференции молодых ученых СПбГУ ИТМО (2008), XV международной научно-методической конференции «Высокие интеллектуальные технологии и инновации в образовании и науке» (Санкт-Петербургский государственный политехнический университет, 2008).

**Публикации.** По теме диссертации опубликовано шесть печатных работ, в том числе три в изданиях из перечня ВАК («Известия РАН. Теория и системы управления» и «Научно-технический вестник СПбГУ ИТМО»).

**Структура диссертации.** Диссертация изложена на 114 страницах и состоит из введения, четырех глав и заключения. Список литературы содержит 32 наименования. Работа содержит 42 рисунка и 15 таблиц.

### Содержание работы

Во введении обосновывается актуальность темы диссертационной работы, описывается предмет исследования, ставятся цель и задачи исследования. Выполнена оценка новизны полученных результатов и формируются положения, выносимые на защиту.

**Первая глава** содержит обзор задач, в которых автоматы строятся с помощью генетических алгоритмов. Рассматриваются такие задачи, как:

- построение простейшего предсказателя (эксперименты Л. Фогеля);
- оптимизация функций;
- автоматические переговоры;
- распознавание нерегулярных языков;
- проектирование логических схем для автоматов Мили.

*Генетические алгоритмы* – это оптимизационный метод, базирующийся на принципах естественной эволюции популяции особей (индивидов). Задача оптимизации состоит в максимизации функции приспособленности (фитнес-функции). В процессе эволюции (в результате отбора, скрещивания и мутаций особей, а также других операторов генетических алгоритмов) происходит поиск особей с высокой приспособленностью.

Методы генерации автоматов можно разделить на несколько групп, в зависимости от того, какой класс генетических алгоритмов используется:

- *Эволюционные алгоритмы* (по Л. Фогелю), в которых новые особи появляются только за счет операторов мутации. Алгоритмы данного класса не требуют много памяти, однако работают они довольно медленно и легко могут «застрять» в локальном экстремуме.
- *Генетические алгоритмы* используют не только операторы мутации, но и операторы скрещивания. Такие алгоритмы сходятся существенно быстрее, чем эволюционные, однако для их работы требуется популяции большого размера.
- *Генетическое программирование* является подклассом генетических алгоритмов, в котором особи кодируются с помощью деревьев решений.

В данной работе для генерации автоматов используются генетические алгоритмы.

Важную роль при разработке генетического алгоритма для конкретной задачи играет выбор способа кодирования особи. В тех случаях, когда решением задачи является автомат, наиболее часто используется один из следующих способов: битовые строки, граф переходов или дерево решений. В данной работе автоматы задаются графами переходов. Способ кодирования определяет, какие именно операторы скрещивания и мутации требуется использовать.

Выбор функции приспособленности также играет важную роль. Примером является функция, получающая на вход битовую строку, которая задает некоторый автомат. Эта функция возвращает число, соответствующее эффективности объекта, управляемого автоматом.

Как отмечалось выше, часто построение вручную автоматов, которые могут быть сгенерированы с помощью генетических алгоритмов, является крайне сложной, а, возможно, и невыполнимой задачей. Имеющие место в прошлом ограничения вычислительной техники существенно ограничивали число особей в поколении и не позволяли решать задачи со сложными функциями приспособленности.

Для эффективной работы генетических алгоритмов требуются операторы скрещивания и мутации, учитывающие особенности задачи. В случае, когда решением является автомат, стандартных операторов оказывается недостаточно. В данной работе предлагается несколько модификаций генетических алгоритмов, позволяющих повысить эффективность генерации автоматов.

**Вторая глава** содержит описание методов генерации автоматов для задачи о флибах, которая появилась в ходе первых работ по моделированию способностей живых существ к предсказанию.

Задача о флибах – это задача моделирования элементарного живого существа, которое способно предсказывать изменения простейшей окружающей среды, имеющие периодичность. Это существо моделируется автоматом (сначала автоматом Мили, а затем автоматом Мили с флагами), а генетические алгоритмы позволяют генерировать автоматы, которые предсказывают изменения среды с достаточно высокой точностью. Таким образом, при решении этой задачи требуется построить «устройство» (*предсказатель*), которое предсказывает изменения среды с наибольшей вероятностью.

В настоящей работе используется генетический алгоритм, основанный на турнирном отборе и принципе элитизма.

*Предлагаемый* алгоритм построен на основе классического генетического алгоритма и учитывает специфику решаемой задачи. Он имеет следующий вид.

1. Создается текущее поколение случайных флибов.
2. Подсчитывается сколько изменений окружающей среды правильно предскажет каждый из флибов.
3. Строится новое поколение флибов:
  - а. Создается пустое новое поколение, и в него добавляется лучший предсказатель из текущего поколения.

- b. Случайным образом из текущего поколения выбираются две пары флибов.
  - c. Из каждой пары флибов выбирается лучший предсказатель.
  - d. Лучшие предсказатели из указанных пар *скрещиваются*.
  - e. Случайным образом определяется необходимость применения *оператора мутации* к полученному флибу. Если это необходимо, то указанный оператор применяется к флибу.
  - f. К флибу применяется *новый оператор мутации* – «восстановление связей между состояниями автомата».
  - g. Флиб добавляется в новое поколение.
  - h. Переходим к пункту b, если размер нового поколения меньше размера текущего поколения.
4. Текущее поколение флибов заменяется новым.
  5. Если число поколений меньше заданного пользователем, то переходим к пункту 2.

В общем случае при генерации генетическим алгоритмом нового поколения решений и применении операторов скрещивания и мутации, переходы в автоматах изменяются случайным образом. При таком изменении переходов в автомате, как правило, возникают состояния, в которые невозможно попасть из начального состояния при любой последовательности значений входных переменных. Будем называть такие состояния *недоступными*. Состояния, в которые можно попасть из начального состояния при некоторой последовательности значений входных переменных, будем называть *доступными*. Предлагаемый оператор мутации восстановления связей между состояниями изменяет переходы в автомате таким образом, чтобы в нем не было недоступных состояний.

Опишем алгоритм реализации предлагаемого оператора мутации.

1. Формируется список доступных состояний. Для этого можно, например, использовать алгоритм рекурсивного обхода графа.
2. Выполняется цикл по всем состояниям. Если текущее состояние не входит в число доступных состояний, то для него выполняются следующие операции:
  - a. Случайным образом выбирается состояние из списка доступных состояний.
  - b. Выбирается случайным образом один переход из выбранного состояния.
  - c. В текущем состоянии заменяется один переход из этого состояния на переход, который ведет в то же состояние, что и переход, выбранный в пункте b.
  - d. Определенный в пункте b переход заменяется переходом, который ведет в текущее состояние.
  - e. Обновляется список доступных состояний. В него добавляется текущее состояние и все состояния, в которые можно попасть из него.

Предложенный оператор является универсальным и может применяться в тех случаях, когда автоматы заданы графами переходов.

Далее в работе приводятся результаты компьютерных экспериментов, которые показывают, что оператор восстановления связей между состояниями повышает эффективность работы генетического алгоритма как при различных битовых масках, задающих среду, так и при различном числе состояний флибов.

Функция переходов автомата зависит от его текущего состояния и входного воздействия. Функция выходов автомата Мили зависит от тех же параметров, что и функция переходов. В автомате Мили с флагами состояние зависит также и от предыстории входных воздействий, которая запоминается с помощью переменных, называемых флагами (задержанными переменными).

Применительно к задаче о флибах флаговые переменные используются для хранения предыстории окружающей среды. Во флаговой переменной с номером  $i$  будем хранить значение параметра, формируемого средой на предыдущем шаге, а во флаговой переменной с номером  $n$  – значение этого параметра на  $n$  шагов назад.

Пусть  $x$  и  $F = (f_1, \dots, f_n)$  – значение параметра, формируемого средой, и вектор флаговых переменных в текущий момент времени, соответственно. Тогда вектор флаговых переменных для следующего момента времени будет иметь вид  $F' = (f'_1 = x, f'_2 = f_1, \dots, f'_n = f_{n-1})$ . При программной реализации флаговые переменные можно хранить в массиве. В этом случае переход от  $F$  к  $F'$  осуществляется с помощью удаления последнего элемента массива и вставки в начало массива текущего значения  $x$ .

Введение каждой флаговой переменной увеличивает число переходов из каждого состояния флиба вдвое. При  $n$  флаговых переменных число переходов из каждого состояния флиба будет равно  $2^{n+1}$ . Номер перехода, который должен использоваться в текущий момент времени, можно вычислить по формуле  $x + 2^1 f_1 + 2^2 f_2 + \dots + 2^n f_n$ .

В рассматриваемой главе приводятся результаты компьютерных экспериментов, показывающие, что использование автоматов с флагами является более эффективным методом для улучшения точности работы генетического алгоритма по сравнению с увеличением числа состояний автоматов.

**Третья глава** содержит описание методов генерации автоматов для задачи об умном муравье, которая стала классической в рассматриваемой области.

На тороидальном поле размером 32 на 32 вдоль определенной ломаной линии («тропы») расположено 89 ячеек с едой. Расположение ячеек с едой на «тропе» фиксировано. На рис. 1 приведено пример «тропы», называемый в литературе «тропой Джона Мьюра» (*John Muir Trail*), которая образована черными и серыми ячейками. При этом каждая черная ячейка содержит одну единицу еды.



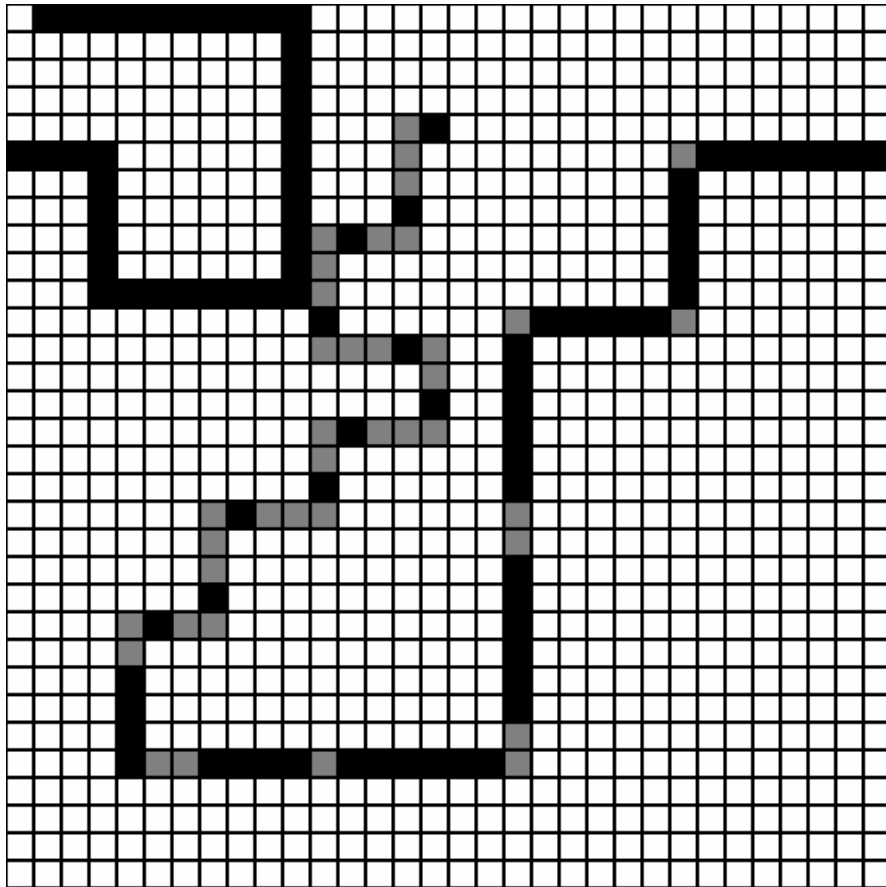


Рис. 1. Тропа Джона Мьюра

Перед первым шагом муравей находится в верхней левой ячейке и смотрит направо. Он может определить, есть ли еда в ячейке, находящейся непосредственно перед ним. За ход муравей может сделать одно из следующих трех действий – подвинуться вперед, повернуться налево или направо.

Для того чтобы съесть еду в ячейке, муравью необходимо попасть в нее. После этого ячейка считается пустой, и неотличима от изначально пустых ячеек. Муравей должен съесть всю еду не более чем за 200 ходов.

Эта задача может быть решена, если муравья моделировать автоматом. При этом требуется решить задачу с помощью алгоритма, который обладает большим быстродействием по сравнению с классическим.

Муравей определяет значение входной битовой переменной (ноль – еды в ячейке перед муравьем нет, единица – еда в этой ячейке муравьем есть). После получения входного воздействия автомат генерирует значение выходной переменной (вперед, налево или направо) и переходит в новое состояние.

В диссертации при построении генетического алгоритма используется отбор отсечением в качестве стратегии формирования особей для следующего поколения. При использовании такого отбора родительские решения выбираются из группы лучших решений текущего поколения. Размер этой группы решений (порог отсечения)

составляет 20% от размера поколения. Все решения из группы лучших решений имеют одинаковые шансы быть выбранными в качестве родительских решений.

Приведем описание предлагаемого генетического алгоритма, использующего отбор отсечением:

1. Текущее поколение решений заполняется случайными решениями.
2. Из текущего поколения решений отбирается группа лучших решений.
3. Формируется новое поколение решений:
  - a. Создается пустое новое поколение решений.
  - b. Из группы лучших решений случайным образом выбирается пара решений.
  - c. Формируется новое решение с помощью применения  $n$ -точечного оператора скрещивания к двум выбранным решениям.
  - d. К новому решению применяются операторы мутации ( $n$ -точечный оператор мутации и *новый* оператор мутации).
  - e. Новое решение добавляется в новое поколение решений.
  - f. Если размер нового поколения меньше размера текущего поколения, то переходим к пункту b.
4. Новое поколение становится текущим.
5. Если число созданных поколений меньше заданного пользователем, переходим к пункту 2.

Для повышения эффективности стандартного генетического алгоритма в работе предлагаются еще один *новый оператор мутации* (сортировка состояний в порядке использования) и измененная функция приспособленности.

Во многих задачах в большинстве решений, которые перебирает генетический алгоритм, автоматы в процессе вычисления значения функции приспособленности не попадают в часть состояний. Состояния, из которых выполняется переход хотя бы на одном шаге моделирования, будем называть *используемыми*, а все остальные состояния – *неиспользуемыми*.

Если при вычислении функции приспособленности на вход автомата подаются все возможные последовательности входных значений, то неиспользуемые состояния и переходы, связанные с ними, не влияют на поведение автомата.

Опишем алгоритм реализации нового оператора мутации – сортировка состояний в порядке использования.

1. Создается пустой словарь пар номеров: [«старый номер состояния» – «новый номер состояния»].
2. Моделируется работа автомата. Перед каждым переходом выполняется следующее: если в словаре нет пары, в которой первый элемент равен текущему номеру состояния, то в него добавляется пара [текущий номер состояния – число пар в словаре].

3. Выполняется цикл по всем состояниям автомата. Для каждого из них: если в словаре нет пары, в которой первый элемент равен номеру состояния, то в него добавляется пара [номер состояния – число пар в словаре].
4. Согласно словарю, изменяется порядок состояний и номера состояний, в которые ведут переходы.

Состояния, для которых в словарь добавляются пары в пункте 2 – используемые состояния. Состояния, для которых в словарь добавляются пары в пункте 3 – неиспользуемые состояния.

В настоящей работе известная для данной задачи функция приспособленности (число съеденных единиц еды) изменена таким образом, что наиболее приспособленными оказываются те решения, в которых меньше используемых состояний. Для этого в функцию приспособленности добавляется соотношение:

$$\frac{1}{1 + \text{useful\_state\_count}}, \quad \text{где } \text{useful\_state\_count} \text{ – число}$$

используемых состояний.

Так как это соотношение всегда меньше единицы, то наиболее приспособленными будут те муравьи, которые съедают больше еды за 200 ходов.

Все эксперименты, о которых идет речь ниже, проводились при размере поколения 10000 и пороге отсечения 2000. Вероятность применения  $n$ -точечного оператора скрещивания к элементу автомата выбрана равной 0.04, а вероятность применения  $n$ -точечного оператора мутации – 0.02. Число состояний автомата было ограничено двадцатью. Эксперименты проводились на компьютере с процессором *AMD Athlon 3800+*.

В табл. 1 приведены результаты экспериментов, в которых использовалась предложенная в работе функция приспособленности. Число поколений в экспериментах, проводившихся с  $n$ -точечным и предложенным в работе операторами мутации, было ограничено числом 100. Число поколений в экспериментах, в которых использовался только  $n$ -точечный оператор мутации, не превышало 200. Лучшим результатом считался муравей, съедающий максимальное число единиц еды. Каждый эксперимент без нового операторов мутации и с известной функцией приспособленности длился примерно 320 с – время построения двухсот поколений. Продолжительность каждого эксперимента с новым оператором мутации и измененной функцией приспособленности составила около 290 с – время построения ста поколений.

Таблица 1. Результаты экспериментов по построению автомата, моделирующего умного муравья

	Номер эксперимента	Номер поколения	Результат	Число используемых состояний	Время поиска
Без применения оператора «сортировка состояний в порядке использования» и новой функции приспособленности	1	137	87	15	225
	2	180	88	15	218
	3	73	88	12	137
	4	72	86	12	114
	5	68	89	12	104
С применением оператора «сортировка состояний в порядке использования» и новой функции приспособленности	1	61	89	10	161
	2	85	89	10	263
	3	70	89	10	227
	4	63	89	10	176
	5	96	89	9	269

Проведенные эксперименты показывают, что введение нового оператора мутации и новой функции приспособленности позволило улучшить эффективность генетического алгоритма.

**В четвертой главе**, используя новый оператор мутации «сортировка состояний в порядке использования», предложенный в предыдущей главе, разработан генетический алгоритм генерации автоматов для задачи построения автопилота для упрощенной модели вертолета (в дальнейшем вертолета).

Решаемая задача формулируется следующим образом. Требуется построить автопилот для вертолета, перемещающегося на плоскости. За один шаг вертолет может повернуть на некоторый фиксированный угол и изменить скорость движения (ускориться или замедлиться). Минимальная скорость, с которой может перемещаться вертолет  $V_{\min} = 10^{-4}$ , максимальная –  $V_{\max} = 2$  и ускорение  $a = 0.1$  (все величины в настоящей главе, в том числе и на осях графиков, приведены в условных единицах). Требуется построить автопилот, который за отведенное время полета, проведет вертолет в определенном порядке через максимальное число заранее заданных меток. Метки нумеруются, а вертолет их проходит в порядке возрастания номеров. При этом вертолет не может пропускать метки, но, так как время полета ограничено, то возможна ситуация, что он может не успеть пройти их все. Считается, что вертолет прошел метку, если он оказался от нее на расстоянии не более 0.5.

Входные данные (воздействия) (рис. 2) автопилота представляют собой единственную переменную (номер сектора обзора): положение текущей метки

относительно вертолета, заданное углом между направлением движения вертолета и направлением на метку (рис. 2, а). Сектора всегда неподвижны относительно вертолета, а вертолет летит не по границе двух секторов, а посередине одного из них (рис. 2, б).

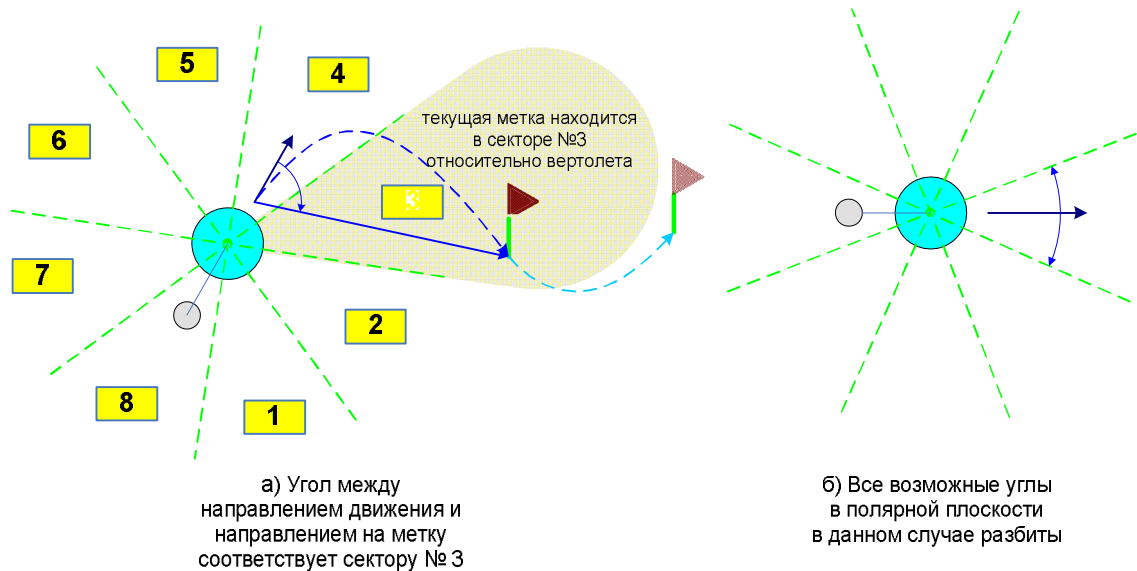


Рис. 2. Входные данные автопилота

При решении этой задачи в качестве стратегии отбора особей для построения следующего поколения используется отбор отсечением. При применении такой стратегии родительские решения выбираются из группы лучших решений текущего поколения. Размер этой группы (порог отсечения) выбран равным 30% от размера поколения. Все автоматы из группы лучших решений имеют одинаковые шансы быть выбранными в качестве родительских особей.

Необходимо отметить, что в данном алгоритме к каждому новому решению последовательно применяются операторы мутации, в качестве которых выступают  $n$ -точечный оператор и предлагаемый автором работы оператор мутации – сортировка состояний в порядке использования.

Все эксперименты проводились при размере поколения 300. При этом порог отсечения равен 90. Максимальное число поколений – 200. Вероятность применения  $n$ -точечного оператора скрещивания – 0.04, а вероятность применения  $n$ -точечного оператора мутации – 0.02. Число состояний автомата не больше 15. Время полета – 200. На рис. 3 изображены двадцать меток, пронумерованных в том порядке, в котором через них должен пролететь вертолет. Точка с координатами (0, 0) является исходной. В начале движения вертолет ориентирован вправо и движется в этом направлении с минимальной скоростью.

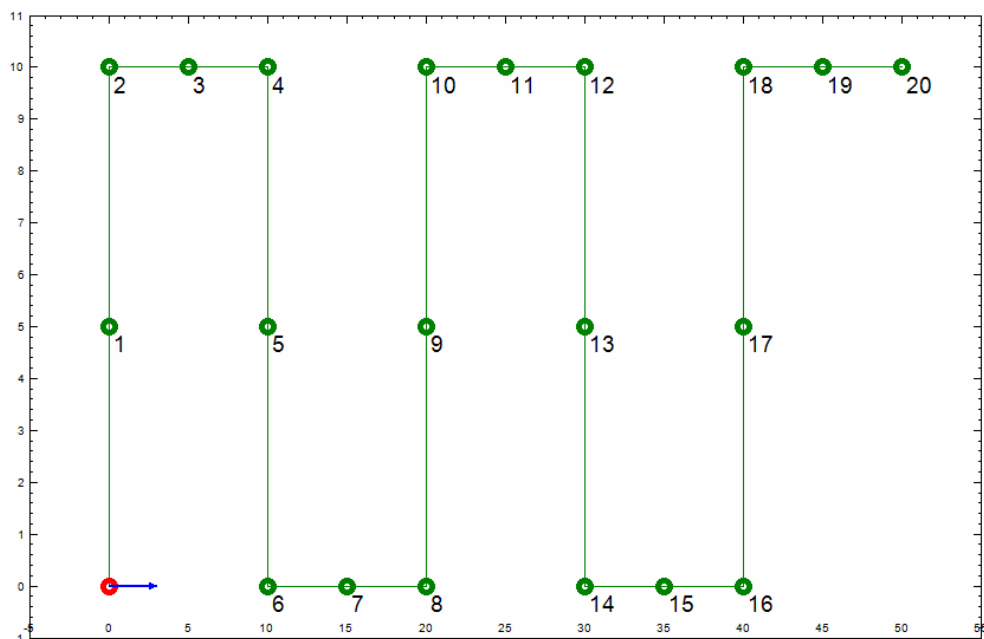


Рис. 3. Метки, проходимые вертолетом

При проведении экспериментов изменялось число секторов обзора вертолета (четыре или шесть), а также использовался или не использовался новый оператор мутации. Эксперименты с каждым набором этих параметров алгоритма проводились по 10 раз.

В табл. 2 приведены результаты экспериментов. Под результатом эксперимента понимается число меток, через которые пролетел вертолет с автопилотом, построенным с помощью предложенного генетического алгоритма. В столбце «худший» приводится самый плохой результат из десяти экспериментов, в столбце «лучший» – самый хороший результат, а в столбце «средний» – усредненный результат.

Таблица 2. Результаты экспериментов по построению автопилота для вертолета

Число секторов обзора	Новый оператор мутации	Результат		
		худший	средний	лучший
4	Нет	12	14.1	16
	Есть	12	14.7	17
6	Нет	11	15.6	18
	Есть	14	16.6	18

Из этой таблицы видно, что использование предлагаемого оператора мутации улучшает эффективность работы генетического алгоритма.

На рис. 4 изображена траектория полета вертолета, управляемого лучшим из построенных автопилотов.

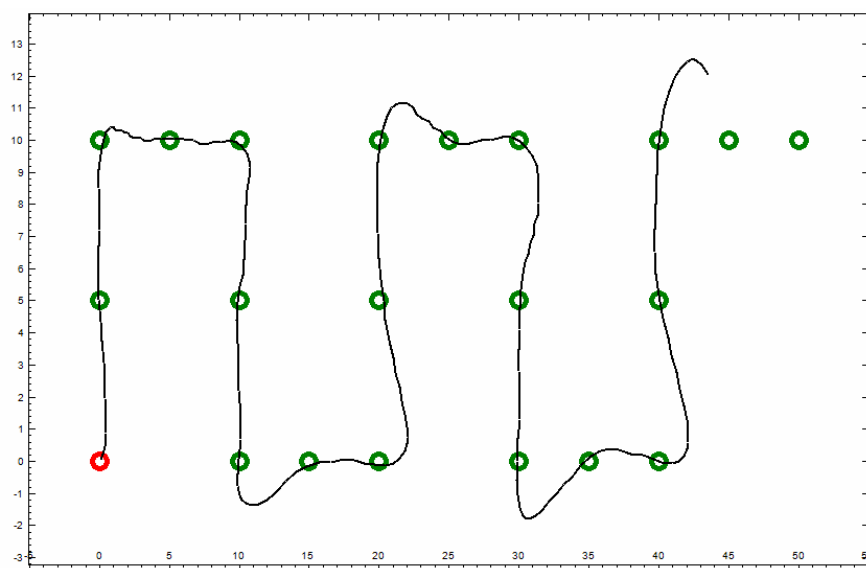


Рис. 4. Траектория полета лучшего вертолета

Также был выполнен эксперимент по использованию восьми секторов обзора вертолета. Однако, несмотря на увеличение времени работы генетического алгоритма, существенного улучшения работы автопилота не наблюдалось.

С помощью предложенного генетического алгоритма был построен автопилот с 12 состояниями, который проводит вертолет через первые 18 меток из 20 за отведенное время. Установлено, что при увеличении времени полета на 33 условных единицы, полученный автопилот обеспечит прохождение всех 20 меток.

Результаты этой главы использовались в компании *Транзас* при разработке тренажера вертолета, что подтверждается актом внедрения.

### Заключение

В диссертации получены следующие научные результаты:

1. Разработаны новые операторы мутации для конечных автоматов, представленных в виде графов переходов: восстановление связей между состояниями и сортировка состояний в порядке использования.
2. Предложена новая функция приспособленности, учитывающая число используемых состояний в автомате, которая позволяет повысить эффективность генетических алгоритмов для генерации автоматов.
3. Разработаны модификации генетических алгоритмов для задач о флибах и умном муравье и для построения автопилота для упрощенной модели вертолета.

4. При решении задачи о флибах выполнено сравнение эффективности генетических алгоритмов, которые позволяют генерировать автоматы Мили и автоматы Мили с флагами. Показано, что точность предсказания флибом, который моделируется автоматом с флагами, выше.

Результаты, полученные в диссертации, используются на практике в компании *Транзас* при разработке тренажера вертолета и в учебном процессе в СПбГУ ИТМО.

#### Список публикаций

1. **Лобанов П. Г., Шалыто А. А.** Использование генетических алгоритмов для автоматического построения конечных автоматов в задаче о флибах / Материалы 1-й Российской мультikonференции по проблемам управления. Сборник докладов 4-й Всероссийской научной конференции «Управление и информационные технологии». СПбГЭТУ «ЛЭТИ». 2006, с. 144-149.
2. **Лобанов П. Г.** Использование генетических алгоритмов для решения задачи об умном муравье // Научно-технический вестник СПбГУ ИТМО. Вып. 39. Исследования в области информационных технологий. Труды молодых ученых. 2007, с. 215-221.
3. **Лобанов П. Г., Шалыто А. А.** Использование генетических алгоритмов для автоматического построения конечных автоматов в задаче о флибах // Известия РАН. Теория и системы управления. 2007. № 5, с. 127-136.
4. **Лобанов П. Г., Шалыто А. А.** Использование автоматов с флагами для решения задачи о флибах // Научно-технический вестник СПбГУ ИТМО. Вып. 42. Фундаментальные и прикладные исследования информационных систем и технологий. 2007, с. 67-74.
5. **Лобанов П. Г., Шалыто А. А.** Использование генетических алгоритмов для решения задачи об умном муравье / XIV Всероссийская научно-методическая конференция «Телематика-2007». СПб. 2007. Т.2, с. 426, 427.
6. **Лобанов П. Г., Сытник С. А.** Использование генетических алгоритмов для построения автопилота для простейшего вертолета / Материалы XV международной научно-методической конференции «Высокие интеллектуальные технологии в образовании и науке». Санкт-Петербургский государственный политехнический университет. 2008. Т.1, с. 298.

Тиражирование и брошюровка выполнены в учреждении  
«Университетские телекоммуникации»  
197101, Санкт-Петербург, Кронверкский проспект, д. 49  
Тел. (812) 233 46 69  
Тираж 100 экз.