МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования   
**«Национальный исследовательский   
Нижегородский государственный университет им. Н.И. Лобачевского»**

**(ННГУ)**

**Институт информационных технологий, математики и механики**

**Кафедра: Математического Обеспечения и Суперкомпьютерных Технологий**

Направление подготовки: «Фундаментальная информатика и информационные технологии»

Магистерская программа: «Компьютерная графика»

**Отчет**

**Тема:**

**«Применение Генетических алгоритмов для решения задач представленных в виде сетевых моделей»**

Допущена к защите Выполнил:

Заведующий кафедрой: студент группы 381806-2м

Доронин Роман Олегович

ученая. степень, ученое звание, ф.и.о

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ подпись подпись

Научный руководитель:

к.ф.-м.н.

Шапошников Дмитрий Евгеньевич

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ученая степень, ученое звание, ф.и.о.

сь

Нижний Новгород  
2019

1. ПОСТАНОВКА ЗАДАЧИ

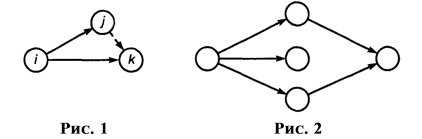
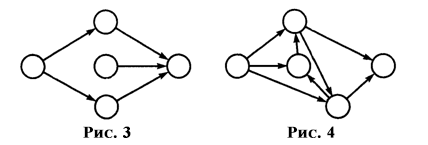
В современном мире большое количество практических задач, можно сформулировать и решить как сетевые модели. Исследования показывают, что не менее 70% реальных задач математического программирования можно представить в виде сетевых моделей. Приведем несколько конкретных примеров.

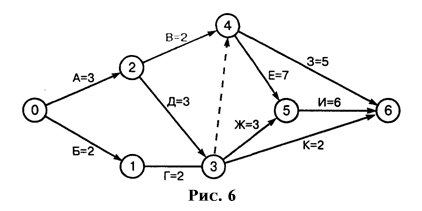
* + - Проектирование газопровода, соединяющего буровые скважины морского базирования с находящейся на берегу приемной станцией. Целевая функция соответствующей модели должна минимизировать стоимость строительства газопровода.
    - Поиск кратчайшего маршрута между двумя городами по существующей сети дорог.
    - Определение максимальной пропускной способности трубопровода для транспортировки угольной пульпы от угольных шахт к электростанциям.
    - Определение схемы транспортировки нефти от пунктов нефтедобычи к нефтеперерабатывающим заводам с минимальной стоимостью транспортировки.
    - Составление временного графика строительных работ (определение дат начала и завершения отдельных этапов работ).

Решение приведенных задач (как и многих аналогичных) требует применения различных сетевых оптимизационных алгоритмов. В нашем исследовании мы будем испытывать в действии Генетические алгоритмы.  
  
Нашей задачей будет:

1. Реализовать Генетический алгоритм, который должен иметь возможности гибкой настройки.

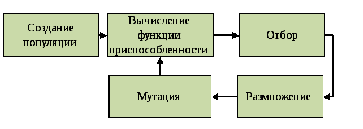
2. Оценить эффективность работы Генетического алгоритма для решения сетевых задач, по сравнению с обычными сетевыми оптимизационными алгоритмами.

1. ОБЗОР ИЗУЧАЕМОЙ ОБЛАСТИ  
   1. **Сетевые модели**  
      До появления сетевых методов планирование работ, проектов осуществлялось в небольшом объеме. Наиболее известным средством такого планирования был ленточный график Ганта, недостаток которого состоит в том, что он не позволяет установить зависимости между различными операциями.   
        
      Современное сетевое планирование начинается с разбиения программы работ на операции. Определяются оценки продолжительности операций, и строится сетевая модель (график). Построение сетевой модели позволяет проанализировать все операции и внести улучшения в структуру модели до начала ее реализации. Строится календарный график, определяющий начало и окончание каждой операции, а также взаимосвязи с другими операциями графика. Календарный график выявляет критические операции, которым надо уделять особое внимание, чтобы закончить все работы в директивный срок. Что касается некритических операций, то календарный план позволяет определить резервы времени, которые можно выгодно использовать при задержке выполнения работ или эффективном применении как трудовых, так и финансовых ресурсов.   
        
      Сетевая модель — графическое изображение плана выполнения комплекса работ, состоящего из нитей (работ) и узлов (событий), которые отражают логическую взаимосвязь всех операций. В основе сетевого моделирования лежит изображение планируемого комплекса работ в виде графа. Граф — схема, состоящая из заданных точек (вершин), соединенных системой линий. Отрезки, соединяющие вершины, называются ребрами (дугами) графа. Ориентированным называется такой граф, на котором стрелкой указаны направления всех его ребер (дуг), что позволяет определить, какая из двух его граничных вершин является начальной, а какая — конечной. Исследование таких сетей проводится методами теории графов.   
        
      Теория графов оперирует понятием пути, объединяющим последовательность взаимосвязанных ребер. Контур означает такой путь, у которого начальная вершина совпадает с конечной. Сетевой график — это ориентированный граф без контуров. В сетевом моделировании имеются два основных элемента — работа и событие.   
        
      Работа — это активный процесс, требующий затрат ресурсов, либо пассивный (ожидание), приводящий к достижению намеченного результата.   
        
      Фиктивная работа — это связь между результатами работ (событиями), не требующая затрат времени и ресурсов.   
        
      Событие — это результат (промежуточный или конечный) выполнения одной или нескольких предшествующих работ.   
        
      Путь — это любая непрерывная последовательность (цепь) работ и событий.   
        
      Критический путь — это путь, не имеющий резервов и включающий самые напряженные работы комплекса. Работы, расположенные на критическом пути, называют критическими. Все остальные работы являются некритическими (ненапряженными) и обладают резервами времени, которые позволяют передвигать сроки их выполнения, не влияя на общую продолжительность выполнения всего комплекса работ.   
        
      При построении сетевых моделей необходимо соблюдать следующие правила.   
        
      1. Сеть изображается слева направо, и каждое событие с большим порядковым номером изображается правее предыдущего. Общее направление стрелок, изображающих работы, также в основном должно быть расположено слева направо, при этом каждая работа должна выходить из события с меньшим номером и входить в событие с большим номером.  
        
        
      2. Два соседних события могут объединяться лишь одной работой. Для изображения параллельных работ вводятся промежуточное событие и фиктивная работа (рис. 1).   
        
      3. В сети не должно быть тупиков, т. е. промежуточных событий, из которых не выходит ни одна работа (рис. 2).  
        
        
        
      4. В сети не должно быть промежуточных событий, которым не предшествует хотя бы одна работа (рис. 3).   
        
      5. В сети не должно быть замкнутых контуров, состоящих из взаимосвязанных работ, создающих замкнутую цепь (рис. 4). Для правильной нумерации событий поступают следующим образом: нумерация событий начинается с исходного события, которому дается номер 1. Из исходного события 1 вычеркивают все исходящие из него работы, на оставшейся сети вновь находят событие, в которое не входит ни одна работа. Этому событию дается номер 2. Затем вычеркивают работы, выходящие из события 2, и вновь находят на оставшейся части сети событие, в которое не входит ни одна работа, ему присваивается номер 3, и так продолжается до завершающего события. Пример нумерации сетевого графика показан на рис. 5.

Продолжительность выполнения работ устанавливается на основании действующих нормативов или по экспертным оценкам специалистов. В первом случае временные оценки являются детерминированными (однозначными), во втором — стохастическими (вероятностными).   
  
Рассмотрим в качестве примера программу создания нового бытового прибора, пользующегося спросом у населения. Необходимые данные приведены в табл. 1.   
  
На основании данных таблицы построим сетевой график создания прибора с учетом вышеизложенных рекомендаций (рис. 6).

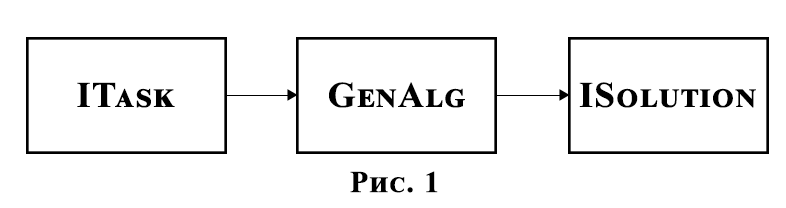
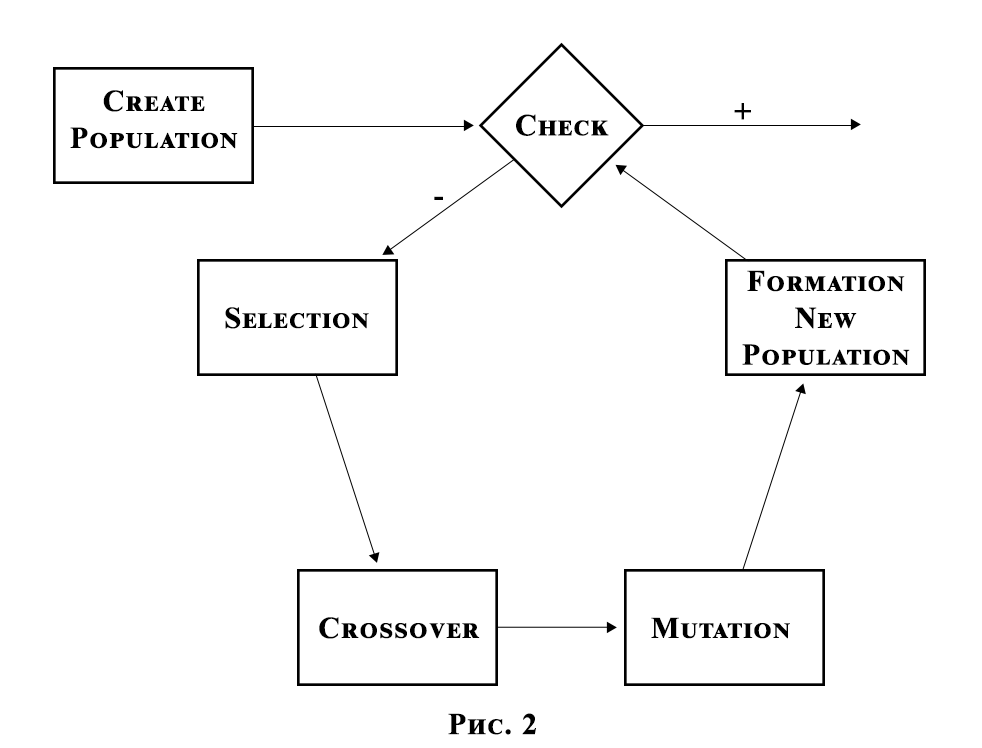
* 1. **Генетические алгоритмы**

Приведем ряд определений основных понятий, которые используются для описания генетических алгоритмов.  
  
Поколение – это совокупность особей, обособленная от других групп, с представителями которых потенциально возможен генетический обмен.   
  
Индивид (особь) – единичный представитель популяции.   
  
Хромосома – это упорядоченные последовательности генов, то есть структуры, содержащие генетическую информацию об индивиде.  
  
Ген – это атомарный элемент генотипа, в частности, хромосомы.   
  
Генотип – это набор хромосом данной особи. Следовательно, особями популяции могут быть генотипы либо единичные хромосомы (в довольно распространенном случае, когда генотип состоит из одной хромосомы).   
  
Фенотип – это набор значений, соответствующих данному генотипу, то есть декодированная структура или множество параметров задачи.   
Аллель – это значение конкретного гена.   
  
Локус – позиция размещения данного гена в хромосоме (цепочке). 

**Описание основной разновидности генетического алгоритма.**  
   
Генетический алгоритм — это эвристический алгоритм поиска. Наиболее часто он используется для решения задач оптимизации и моделирования путём случайного подбора, комбинирования и вариации искомых параметров с использованием механизмов. Называется алгоритм генетическим, так как напоминает биологическую эволюцию и является разновидностью эволюционных вычислений.   
  
Задача генетического алгоритма – максимизация или минимизация функции приспособленности. Эта задача решается постепенно – по мере эволюции. Эволюция состоит из последовательности поколений – популяций.   
  
Отличительной особенностью генетического алгоритма от эволюционной стратегии является акцент на использование оператора «скрещивания», который производит операцию рекомбинации решений-кандидатов.   
  
Для каждого поколения отбираются особи с наилучшими значениями приспособленности. Хромосомы отобранных особей с некоторой вероятностью подвергаются операции скрещивания, затем, с другой вероятностью, подвергаются мутациям, а вслед за этим переносятся в следующее поколение. Обычно вероятность мутации намного меньше вероятности скрещивания. Отбор, скрещивание и мутация являются генетическими операциями, их реализация зависит от конкретной задачи. Размер популяции и функция приспособленности не изменяются в процессе эволюции от поколения к поколению.   
  
Кратко сформулировать принцип генетического алгоритма можно следующим образом: наименее приспособленные особи вымирают, а наиболее приспособленные выживают и дают потомство. Предполагается, что каждое последующее поколение будет содержать “более приспособленных” особей, нежели предыдущее.   
  
Таким образом, основные этапы эталонного генетического алгоритма:   
  
задание функций приспособленности для особей популяции для решаемой задачи (для каждой задачи подбирается своя функция приспособленности);   
  
создание начальной популяции;   
  
этап генерации решения;   
  
применение генетических операций;   
  
вычисление значение целевой функции для всех особей в поколении;   
  
формирование нового поколения (селекция);   
  
выбор “лучшей” хромосомы;   
  
проверка условий останова;   
  
На предложенном ниже рисунке схематически представлена общая схема работы генетического алгоритма:   
  
  
  
Рис. 2. Общая схема работы генетического алгоритма   
  
  
По сравнению с традиционными методами оптимизации генетические алгоритмы имеют следующие преимущества:   
  
генетические алгоритмы легко модифицируются для параллельных вычислений;   
  
легко приспосабливаются для решения широкого спектра задач;   
  
хорошо подходят для оптимизации не дифференцируемых функций;   
  
Основными недостатками генетических алгоритмов являются:   
  
высокая трудоемкость;   
  
сложность контроля генерации решения;   
  
сложность оценки степени пригодности конкретных генетических алгоритмов для решения конкретной задачи.   
  
**Генетическое программирование**   
  
Генетическое программирование – разновидность генетических алгоритмов, в которой вместо низкоуровневых представлений объектов в виде битовых строк используются деревья разбора, диаграммы переходов конечного или клеточного автомата и т.д. С помощью генетического программирования наиболее эффективно решаются задачи автоматического построения некоторых классов задач, конечных автоматов и клеточных автоматов.   
  
Стагнация генетического алгоритма – такое состояние алгоритма, при котором на протяжении большого числа поколений не было изменения лучшего значения функции приспособленности у популяции, но текущее решение сильно отличается от глобального минимума.   
  
Схождением называется такое состояние популяции, когда все хромосомы популяции имеют много схожих частей и находятся в области некоторого экстремума. В такой ситуации операции скрещивания практически никак не могут изменить популяцию. А вышедшие из этой области за счет мутации особи склонны вымирать, так как чаще имеют значение функции приспособленности, которое хуже, чем в остальной популяции. Если данный экстремум является глобальным минимумом, то это означает, что найдено лучшее или близкое к нему решение.   
  
Вырождением популяции называется такая разновидность схождения, когда текущий экстремум сильно отличается от глобального минимума, то есть почти вся популяция забилась решениями, которые удовлетворяют какому-либо локальному минимуму. Вырождение популяции является одной из основных проблем при использовании генетических алгоритмов.   
  
**Стандартные операции скрещивания**   
  
Общая часть для всех алгоритмов следующая: особи промежуточной популяции случайным образом разбиваются на пары. Каждая из них с вероятностью pc скрещивается, т. е. к ней применяется оператор кроссовера, в результате чего получаются два потомка. Они записываются в новое поколение. Если же паре не выпало скрещиваться, в новое поколение записываются сами особи этой пары.   
  
Точечное скрещивание: выбираются пары хромосом из родительской популяции. Далее для каждой пары отобранных таким образом родителей разыгрывается позиция гена (локус) в хромосоме, определяющая так называемую точку скрещивания – lk. Если хромосома каждого из родителей состоит из генов L, то очевидно, что точка скрещивания lk представляет собой натуральное число, меньшее L. Поэтому фиксация точки скрещивания сводится к случайному выбору числа из интервала [1, L-1]. В результате скрещивания пары родительских хромосом получается следующая пара потомков:   
  
потомок, хромосома которого на позициях от 1 до lk состоит из генов первого родителя, а на позициях от lk + 1 до L – из генов второго родителя;   
  
потомок, хромосома которого на позициях от 1 до lk состоит из генов второго родителя, а на позициях от lk + 1 до L – из генов первого родителя.   
  
Двухточечное скрещивание: отличается от точечного скрещивания тем, что родительские хромосомы обмениваются участком генетического кода, который находится между двумя случайно выбранными точками скрещивания.   
  
Многоточечное скрещивание представляет собой обобщение предыдущих операций и характеризуется соответственно большим количеством точек скрещивания.   
  
**Стратегии отбора нового поколения**   
  
Выделяют два типа формирования нового поколения после получения множества детей в результате генетических операций:   
  
новые особи замещают родителей;   
новое поколение составляется из совокупности и новых особей, и их родителей, например, выбором определенного числа лучших.   
  
Ниже рассмотрим основные используемые стратегии отбора нового поколения.   
  
Стратегия рулетки. При использовании такого метода вероятность выбора хромосомы определяется ее приспособленностью, то есть чем лучше значение функции приспособленности у хромосомы, тем выше у нее шанс попасть в новое поколение   
  
Турнирный отбор. Его суть заключается в том, что случайно выбирается несколько особей из популяции и победителем выбирается особь с наибольшей приспособленностью, она и попадает в новое поколение.  
   
Принцип элитизма. В новое поколение обязательно включается заданное количество лучших особей предыдущего поколения (часто только одна лучшая особь).   
  
У каждого и описанных выше методов есть свои преимущества и недостатки, поэтому чаше всего используется их комбинация. Так использование стратегии элитизма часто оказывается весьма полезным для эффективности, так как не допускается потеря лучших решений, то есть полученное хорошее решение будет оставаться в популяции до тех пор, пока не будет найдено решение, которое лучше находящегося в популяции. Недостаток использования стратегии элитизма в том, что повышается вероятность попадания алгоритма в локальный минимум.   
 **Модели генетических алгоритмов**   
  
Очевидно, что использование только специализированной стратегии отбора не всегда помогает избежать локальных экстремумов в процессе генерации решения. Поэтому зачастую вместо стандартного генетического алгоритма используют разнообразные модели, которые в некоторых случаях улучшают сходимость процесса. Ниже рассмотрены наиболее используемые модели генетических алгоритмов, описаны их основные преимущества и недостатки.   
  
Параллельные ГА   
  
Очевидно, что в генетическом алгоритме присутствуют множество независимых вычислений, следовательно, можно организовать несколько параллельно выполняющихся процессов для увеличения производительности на многопроцессорных устройствах. Ниже приведем несколько возможных стратегий распараллеливания генетического алгоритма:   
  
Параллельное выполнение генетических операций и алгоритмов отбора нового поколения. Например: параллельный турнирный отбор: заведем N/2 процессов, каждый из них будет выбирать случайно из популяции 4 особи, проводить 2 турнира, и победителей скрещивать. Полученные дети будут записываться в новое поколение. Таким образом, за один цикл работы одного процесса будет сменяться целое поколение.   
  
Параллельное вычисление значений фитнесс функций у особей. Заводятся N процессов. Особи, у которых необходимо выполнить вычисления кладутся в специальную очередь. Каждый процесс вытаскивает особь из очереди и производит необходимые вычисления.   
  
**Островная модель**   
  
Эта модель является одной из разновидностей параллельных генетических алгоритмов, однако кроме распределения вычислительной нагрузки она часто применяется для решения задач, для которых характерна многоэкстремальность функции приспособленности.   
  
Принцип работы стандартной островной модели заключается в том, что вместо одной большой популяции используется набор популяций меньшего размера. Эти популяции развиваются независимо, и только изредка происходит обмен представителями между популяциями.   
  
Так как чаще всего число особей, содержащихся в островных популяциях невелико, то такие подпопуляции склонны к преждевременной сходимости. Поэтому важно правильно установить частоту миграции. Чересчур частая миграция или миграция слишком большого числа особей приведет к смешению всех подпопуляций, тогда островная модель будет несильно отличаться от обычного ГА. Если же миграция будет слишком редкой, то она не сможет предотвратить преждевременного схождения подпопуляций. Также заметим, что иногда при миграции производится мутация или какая-либо иная модификация перемещаемых особей.   
  
Генетические алгоритмы стохастичны, поэтому при разных его запусках популяция может сходиться к разным решениям. Островная модель позволяет запустить алгоритм сразу несколько раз и пытаться совмещать «достижения» разных островов для получения в одной из подпопуляций наилучшего решения. Использование островной генетической модели позволяет во многих случаях избежать локальных минимумов, что положительно сказывается на сходимости к решению. Заметим, что при этом зачастую время работы генератора несколько больше, чем у стандартной модели, в силу того, происходит дупликация генерации решений.   
  
**Клеточная модель генетических алгоритмов**   
  
Эта модель представляет собой вариацию островной модели. На этапе инициализации строится двумерная сетка. В каждой ячейке сетки запускается содержится особь, при этом определяется процесс взаимодействия между ячейками: каждая ячейка может взаимодействовать только с четырьмя своими соседями (сверху, снизу, слева, справа), при этом для нее выбирается лучшую особь среди своих соседей, скрещивается с ней особь из своей ячейки и полученный ребенок помещается в свою ячейку вместо родителя. Заметим, что в ячейке может находиться не одна ячейка, а подпопуляция, тогда описанная выше модель представляет собой вариацию островной модели, в которой накладываются ограничения на взаимосвязи между островами.   
  
По мере работы такой модели возникают эффекты, похожие на островную модель. Сначала все особи имеют случайную приспособленность. Спустя несколько поколений образуются небольшие области похожих особей с близкой приспособленностью. По мере работы алгоритма происходит рост и конкуренция областей между собой.   
  
Основное преимущество заключается в том, что происходит генерация сразу нескольких решений в разных «областях». За счет этого использование клеточной генетической модели позволяет во многих случаях избежать локальных минимумов. Основной недостаток заключается в более низкой скорости сходимости по сравнению со стандартным генетическим алгоритмом. 

1. ОБЗОР ТЕХНОЛОГИЙ  
   1. **Готовые реализации генетического алгоритма**
      * **GAUL (The Genetic Algorithm Utility Library, C/C++)**Гибкая библиотека, предназначенная для помощи в разработке приложений, использующих генетические или эволюционные алгоритмы. Библиотека предоставляет структуры данных и функции для обработки и манипулирования ими. Для сравнения с генетическими алгоритмами предусмотрены стохастические алгоритмы.
      * **GAlib (Library of Genetic Algorithm Components, C++)**Библиотека имеет обширные возможности добавления и замены своими компонентами (Генетическими операторами).
      * **GALGO (С++)**Более современный продукт, использующий современный С++ и более лучшие методы распараллеливания. В основе используются шаблоны.
      * **OpenGA (С++)**Похожа на GALGO, в дополнении обладает более дружественным API и имеет встроенные возможности визуализации процесса работы и результатов.

В GALGO и OpenGA также есть возможность добавлять свои компоненты.

1. РЕАЛИЗАЦИЯ  
   1. **Текущие результаты**Была изучена теория по Генетическим алгоритмам. Параллельно с этим в рамках изучения, была написана программа на языке C#.  
        
      Программа представляет из себя большую структуру классов, в которой по-отдельности можно менять компоненты ГА, что дает большую свободу в настройках.  
        
      **Общая схема**  
        
        
        
      Имеется некоторый формат входа для ГА представленный интерфейсом ITask. Любая задача должна реализовывать его. Далее идет работа самого алгоритма и на выходе получаем некоторое тоже заранее определенное решение ISolution. (Рис. 1)
   2. **Схема работы Генетического алгоритма**  
        
        
        
      Реализованный алгоритм работает по самой известной схеме.  
      1. Создание популяции (Create Population) – эта задача возлагается на ITask, поскольку мы должны изначально уже знать какой будет формат ответа.  
      2. Проверка на решение (Check) – вариантов проверки или остановки алгоритма может быть множество. Обычно здесь используется количество поколений.  
      3. Выбор (Selection) – в программе реализованы Выбор лучший, Рулеточный выбор и Турнирный выбор.  
      4. Скрещивание (Crossover) – в программе реализованы Кроссинговер и Рекомбинация  
      5. Мутация (Mutation) – в программе реализованы Мутация аллели и Мутация гена  
      6. Формирование новой популяции (Formation New Population) – в программе реализованы отбор лучших и отбор рулеткой.  
        
      В качестве тестовой задачи текущий вариант реализации ГА решал NP-полную «задачу о рюкзаке» с 10 предметами средним весом 15 единиц и ограничениями 300 и 800 единиц веса. При должных подобранных настройках (операторах) ГА, алгоритм находил лучшее решение в среднем из 1000 раз за 0,15 и 0,47 секунды соответственно.
   3. **Дальнейшая работа**
      * Поиск оптимальных начальных данных для задачи о «худшем клиенте». При этом для проверки корректности работы алгоритма требуется знать ответ на поставленную задачу.
      * Поиск оптимального формата представления начальных данных во входной структуре, представленной в нашей реализации ГА.
      * Анализ трудоемкости решаемой задачи. При нехватке вычислительных ресурсов для решения задач необходимого объема, потребуется прибегнуть к использованию сторонних библиотек, обзор которых был в Пункте 3.
2. СПИСОК ЛИТЕРАТУРЫ

[1] Д. И. Батищев, Генетические алгоритмы решения экстремальных задач, 1995

[2] Д. Рутковская, М. Пилиньский, Л. Рутковский, Нейронные сети, генетические алгоритмы и нечеткие системы, 2006

[3] Хемди А. Таха, Введение в Исследование Операций, 2005