МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
им. Н.Э. Баумана

Кафедра «Систем обработки информации и управления»

ОТЧЕТ

Лабораторная работа № 4 по курсу
Методы поддержки принятия решений

«Решение оптимизационных задач с помощью генетических алгоритмов»

Предметная область: определение нарисованной цифры

7
(количество листов)

ИСПОЛНИТЕЛЬ:

студенты группы ИУ5-74 _____
Повираева Марина
(Poviraeva@me.com) "__"_____2017 г.
Волобуев Василий

ПРЕПОДАВАТЕЛЬ:
Терехов В.И.,
к.т.н., доцент

Кафедра ИУ-5. _____

Москва - 2017

Оглавление

Цель работы

Целью лабораторной работы является углубление и закрепление теоретических знаний, полученных на лекциях, приобретение практических навыков самостоятельной работы при решении оптимизационных задач больших размерностей с помощью генетических алгоритмов.

В процессе выполнения лабораторной работы по теме «Решение оптимизационных задач с помощью генетических алгоритмов» на примере задачи поиска кратчайшего пути для информационного пакета (сообщения) в компьютерной сети студенты решают следующие задачи (задания):

- описывают предметную область;
- определяют исходные данные задачи;
- формулируют задачу и исходные данные в терминах генетических алгоритмов;
- определяют последовательность работы генетического алгоритма;
- разрабатывают компьютерную программу;
- исследуют работу генетического алгоритма и полученное решение.

Задание

Сформулировать задачу и описать исходные данные в терминах генетических алгоритмов.

Разработать программу, которая осуществляет поиск кратчайшего пути для информационного пакета (сообщения) в компьютерной сети с помощью генетического алгоритма.

При проведения серии экспериментов (не меньше 10) по исследованию работы генетического алгоритма программа должна позволять пользователю задавать топологию сети (пропускные способности каналов связи), содержащей не менее 10 компьютеров (серверов), а также указывать компьютер-отправитель и компьютер-получатель. Должны отображаться все решения (хромосомы) одного поколения до и после применения каждого оператора (скрещивания, селекции, редукции и мутации). Переход к следующему поколению должен осуществляться: в автоматическом режиме в соответствии с заданным критерием; в ручном режиме.

Описание предметной области и выбранной задачи

Генетические алгоритмы относятся к числу универсальных методов оптимизации, позволяющих решать задачи различных типов и различной степени сложности. При этом ГА характеризуются возможностью как однокритериального, так и многокритериального поиска в большом пространстве, ландшафт которого является негладким.

Эволюционные алгоритмы, моделирующие процессы естественной эволюции, были предложены уже в 60-х годах прошлого века. Их особенностью является то, что они опираются на естественную эволюцию в природе, используя основные ее механизмы:

- отбор или селекцию;
- скрещивание;
- мутацию.

Известны утверждения: "алгоритм является хорошим оптимизационным методом, потому что его принцип используется в природе", и наоборот: "алгоритм не может быть хорошим оптимизационным методом, потому что вы не находите его в природе".

Формулировка задачи, описание исходных данных в терминах генетических алгоритмов и блок-схема генетического алгоритма

Имеется полносвязный граф и задаваемая пользователем или случайным образом матрица весов путей между вершинами (NET_SIZE).

При генерировании матрицы весов случайным образом получается симметричная матрица из треугольной с нулями на главной диагонали.

Задаётся максимальное число поколений (NUM_OF_EXPS = 100), предел «простаивающих» поколений (PASS_LIMIT = 30 поколений без изменений), fitness-функция задаётся через длину пути, которую мы минимизируем. Также задаются входные и выходные вершины на графе.

В качестве хромосом выбраны маршруты по графу, где гены означают конкретную вершину. Популяция числом POP_SIZE состоит из множества хромосом.

Ниже представлена блок-схема реализованного генетического алгоритма с пояснениями.

1. Создание популяции:

   Создаём 2*<Размерность графа> особей для лучшего захвата ген.

2. Селекция:

   Отбор особей для скрещивания в надежде получить в следующем поколении лучший результат. Выбираем 25% самых успешных в популяции, скрещиваем их между собой, получаем 50% новых кандидатов, и добиваем сверху ещё 25% вполне удачных претендентов.

3. Скрещивание:

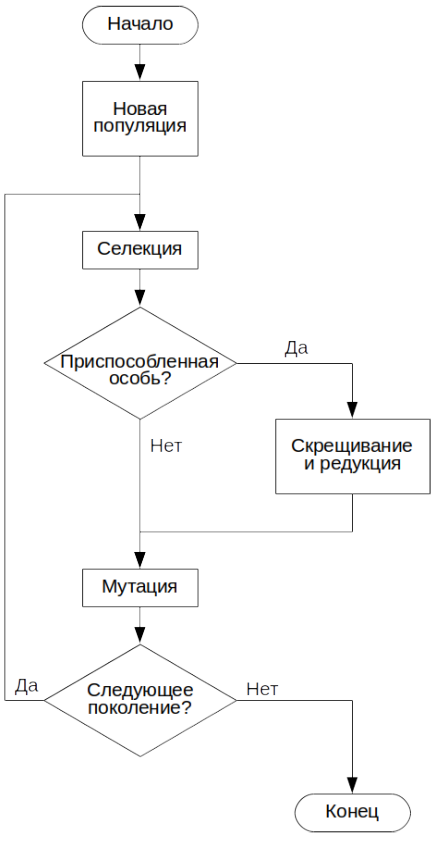
   Повторяет процесс кроссинговера в биологии – в нашем случае случайным образом в середине происходит разрыв и скрепление выбранной пары.

4. Редукция:

   Уменьшение числа хромосом вдвое – удаляем менее приспособленную.

5. Мутация:

   Все исходные и скрещенные хромосомы случайным образом меняют один из своих генов для внесения большего разнообразия.



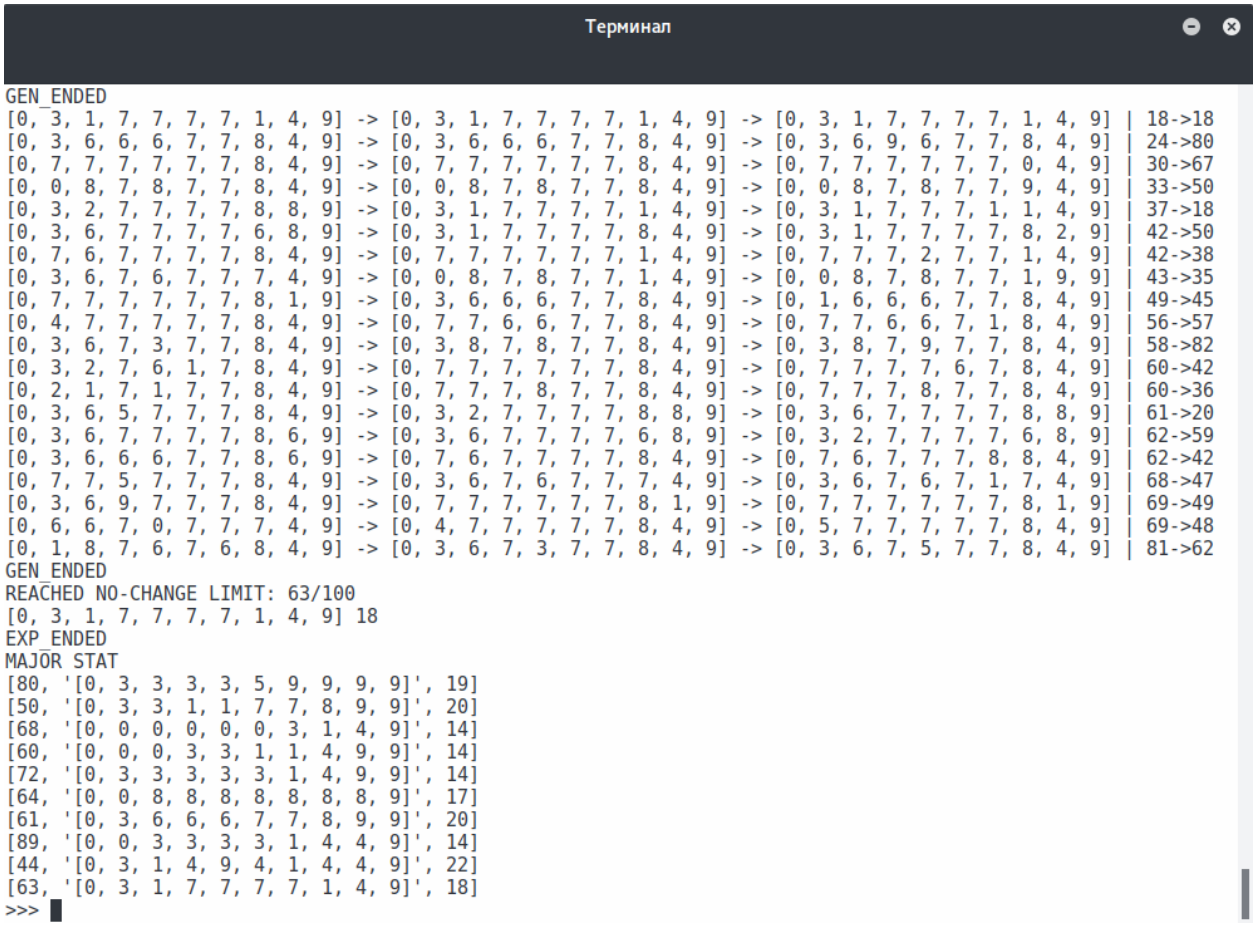
Описание программы, ее ключевые особенности и новшества

Внятный консольный интерфейс позволяет случайным образом задать весы графу весов, а переменные, вынесенные в начало __main__ помогут внести нужные конфигурации в работу программы.

Вручную задать стоимость перехода из одной вершины графа в другую представляется возможным с помощью стандартных средств python.

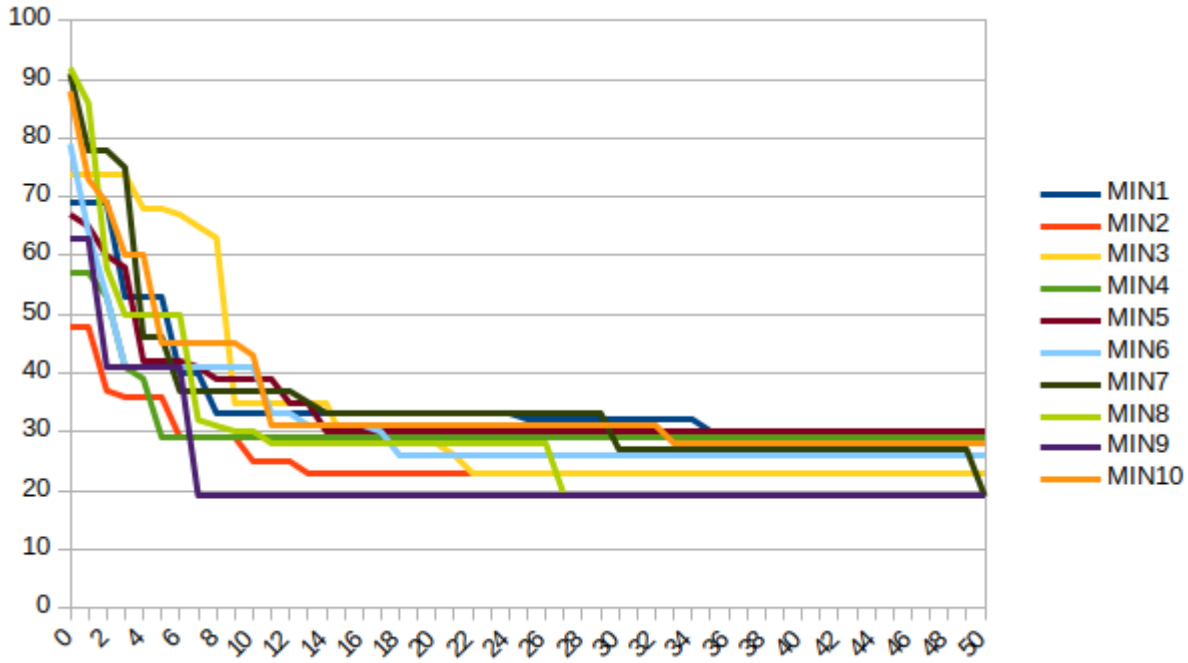
Также присутствует функция Populate для организации случайной популяции особей.

Вся необходимая информация (текущая выборка, скрещивания и мутации) выводятся в консоль пользователю сразу после выполнения команд с минимальной задержкой.



По выполнению основной программы выводится основная статистика по 10 проведённым экспериментам.

Протоколы проведенных экспериментов



Уже к 10-ому поколению результаты удаётся улучшить более чем в 2 раза, однако система быстро уходит в насыщение и выбранный способ скрещивания не позволяет добиться оптимального результата — почти наверняка исключается возможность обхода циклов, остаётся надеяться только на случайные мутации, которые также происходят только в одном гене.

Выводы

В процессе лабораторной работы была реализована программа, которая осуществляет поиск кратчайшего пути для информационного пакета (сообщения) в компьютерной сети с помощью генетического алгоритма. В качестве хромосом использовались пути между компьютерами, а в качестве ген – следующая вершина графа сети. Так как выбранный алгоритм скрещивания меняет только правую часть хромосомы, его эффективность невысока (т. к. часть генов не будет изменяться), а получение решения в большей степени зависит от начальной популяции, сформированной случайным образом.

Используемая литература

1. Терехов В.И. Лекции по курсу «Методы поддержки принятия решений», рукопись.