

Лабораторная работа 4. Генетические алгоритмы: задача коммивояжёра

Задача о коммивояжёре является классической NP-полной задачей о поиске гамильтонова цикла или пути в графе. В неформальном виде задача трактуется следующим образом: коммивояжёру необходимо посетить N городов, побывав в каждом не более 1 раза, с минимальной стоимостью путешествия.

Представим города в виде N связанных вершин графа, причем ребра этого графа взвешены стоимостью. Тогда необходимо найти путь через все вершины с наименьшей стоимостью. Поскольку коммивояжёр в каждом из городов встает перед выбором следующего города из тех, что он ещё не посетил, существует $(N-1)!$ различных маршрутов. Таким образом, размер пространства поиска зависит экспоненциально от количества городов.

Все эффективные (сокращающие полный перебор) методы решения задачи коммивояжёра – методы эвристические. В большинстве эвристических методов находится не самый эффективный маршрут, а приближённое решение.

Нахождение точного решения возможно только для малой размерности задачи (это можно сделать, например, полным перебором). С увеличением количества и задач время нахождения оптимального пути растет очень быстро.

Генетический алгоритм (ГА) – это эвристический алгоритм поиска, используемый для решения задач оптимизации и моделирования путём случайного подбора, комбинирования и вариации искомым параметров с использованием механизмов, напоминающих биологическую эволюцию. Основные механизмы – кроссинговер (скрещивание двух особей и получение новых решений, содержащих признаки обоих) и мутация (случайная смена гена в закодированном решении). Кроме того, для каждой популяции осуществляется оценивание для выявления лучших особей.

Общая процедура поиска ГА состоит в следующем:

1) Генерация популяции: популяция – набор особей, каждая из которых представляет возможное решение (путь коммивояжёра). Начальная популяция может быть получена случайным образом.

2) Оценка особи: каждая особь может быть оценена целевой функцией (стоимость полного пути коммивояжёра).

3) Операции кроссинговера и мутации: кроссинговер выбирает случайную пару особей и выбирает случайную точку в хромосоме первой особи. Хромосомы обеих особей обмениваются частями, полученными после деления по этой точке. Новые особи оцениваются, в популяции остаются лучшие. Мутация в произвольном порядке выбирает хромосому и меняет её, либо выбирает пару хромосом и переставляет их.

4) Хромосомы измененной популяции оцениваются снова. Это завершает одну итерацию ГА. ГА останавливается, когда предопределенное число развития достигнуто, или все хромосомы становятся одинаковыми.

Критерии останова:

- 1) нет улучшения предыдущей оценки;
- 2) все хромосомы сходятся к одному решению;
- 3) достигнута необходимая оценка.

Задание 1. Реализуйте чтение условий задачи из файла *varX.txt*, где X – номер варианта, согласно следующему формату: сначала идет число N – количество городов, затем матрица A размером $N \times N$, где A_{ij} – стоимость пути из города i в город j .

Для упрощения задачи, существует путь из каждого города в каждый.

Пример:

```
7
0 1 2 7 6 3 1
1 0 1 3 6 5 3
2 1 0 1 3 5 6
7 3 1 0 1 3 6
6 6 3 1 0 1 4
3 5 5 3 1 0 1
1 3 6 6 4 1 0
```

Задание 2. Реализуйте целевую функцию F – стоимость заданного пути. Коммивояжёру нужно вернуться в город, с которого он начал. Кодирование пути производится в виде списка городов, в порядке посещения.

Для приведенного выше примера $F(<1\ 4\ 2\ 3\ 5\ 7\ 6>) = 22$.

Задание 3. Реализуйте функцию мутации. Метод мутации реализуется следующим образом: выбирается случайный город, а затем он вставляется в случайное место последовательности.

Так, после мутации последовательность $<1\ 4\ 2\ 3\ 5\ 7\ 6>$ может преобразоваться, например в $<1\ 4\ 3\ 2\ 5\ 7\ 6>$.

Задание 4. Реализуйте функцию кроссинговера.

Использование стандартных генетических операторов в алгоритмах решения задачи коммивояжёра приводит к большому числу нереальных

решений. Так при кроссинговере двух хромосом

$\langle 1\ 4\ 2\ 3\ 5\ 7\ 6 \rangle$,

$\langle 6\ 4\ 2\ 3\ 1\ 5\ 7 \rangle$

можно получить двух потомков

$\langle 1\ 4\ 2\ |\ 3\ 1\ 5\ 7 \rangle$,

$\langle 6\ 4\ 2\ |\ 3\ 5\ 7\ 6 \rangle$

которые, как видно, не являются решением.

Для задачи коммивояжёра существуют модифицированные операторы кроссинговера, например, «Жадная» стратегия, которая делает на каждом шаге «локально-оптимальный» выбор.

«Жадный» алгоритм кроссинговера:

1. Случайным образом выбрать стартовую вершину в первом родителе. Текущий родитель – первый родитель.

2. Выбрать какая из соседних вершин (из вершин, находящихся в хромосоме слева и справа от рассматриваемой) ближе к текущей.

3. Ближайшая из них, не вошедшая в путь, становится новой текущей вершиной.

4. Если только одна из них ещё не вошла в путь, то её и выбираем текущей.

5. Если обе соседние вершины являются посещенными, то выбрать ближайшую из числа не рассмотренных.

6. В качестве текущего родителя выбрать другого родителя.

7. Повторять цикл оператора, пока все вершины не будут посещены.

Пример работы алгоритма для двух хромосом:

$x = \langle 1\ 4\ 2\ 3\ 5\ 7\ 6 \rangle$, $F(x)=22$;

$y = \langle 6\ 3\ 5\ 4\ 2\ 1\ 7 \rangle$, $F(y)=15$.

Пусть на 1 этапе вершина 5 выбрана стартовой:

$x = \langle 1\ 4\ 2\ 3\ \mathbf{5}\ 7\ 6 \rangle$;

$y = \langle 6\ 3\ \mathbf{5}\ 4\ 2\ 1\ 7 \rangle$.

В первом родителе путь 5-3 короче, чем 5-7, поэтому текущий путь будет **5-3**, и текущая вершина 3:

$x = \langle 1\ 4\ 2\ \mathbf{3}\ 5\ 7\ 6 \rangle$;

$y = \langle 6\ \mathbf{3}\ 5\ 4\ 2\ 1\ 7 \rangle$.

На следующем шаге текущий родитель второй. В нем вершина 5 уже занята, выбираем вершину 6, путь 5-3-6:

$x = \langle 1\ 4\ 2\ 3\ 5\ 7\ \mathbf{6} \rangle$;

$y = \langle \mathbf{6}\ 3\ 5\ 4\ 2\ 1\ 7 \rangle$.

Текущий родитель первый, путь 6-7 короче, чем 6-1, тогда выбираем путь 5-3-6-7:

$$x = \langle 1 \ 4 \ 2 \ 3 \ 5 \ 7 \ 6 \rangle;$$

$$y = \langle 6 \ 3 \ 5 \ 4 \ 2 \ 1 \ 7 \rangle.$$

Текущий родитель второй, единственная незанятая вершина 1, тогда путь 5-3-6-7-1:

$$x = \langle 1 \ 4 \ 2 \ 3 \ 5 \ 7 \ 6 \rangle;$$

$$y = \langle 6 \ 3 \ 5 \ 4 \ 2 \ 1 \ 7 \rangle.$$

В первом родителе незанятая вершина 4, путь 5-3-6-7-1-4:

$$x = \langle 1 \ 4 \ 2 \ 3 \ 5 \ 7 \ 6 \rangle;$$

$$y = \langle 6 \ 3 \ 5 \ 4 \ 2 \ 1 \ 7 \rangle.$$

Оставшаяся вершина 2 включается в путь
 $z = 5-3-6-7-1-4-2.$

$F(z_1) = 26$, хуже, чем у родителей, поэтому в популяцию не включается.

Задание 4. Сгенерируйте начальную популяцию. Число особей выбирается самостоятельно. Особей начальной популяции можно получить случайным выбором путей.

Задание 5. Используя написанные ранее функции, реализуйте поиск решения с помощью генетического алгоритма. Самостоятельно подберите оптимальные параметры ГА: количество особей, вероятность мутации, количество скрещиваний и т.д.

В качестве завершающего критерия выбрано ограничение по количеству итераций.