

Генетический алгоритм для задачи комивояжёра

Требуется найти в полносвязном графе маршрут с минимальной стоимостью, проходящий через все вершины ровно один раз. Стоимость определяется как сумма весов рёбер, через которые проходит маршрут.

Задание

Реализовать генетический алгоритм для решения задачи комивояжёра, используя pthreads.

Выбор операторов mutation, crossover, selection предоставляется вам.

В качестве критерия остановки использовать следующий критерий: "остановить алгоритм, если на протяжении последних S итераций не получилось улучшить решение".

Требования:

- Генерировать псевдослучайные числа в одном, специально выделенном, потоке (нужно соблюсти баланс между потребляемой памятью и быстродействием)
- Для вычисления значения фитнес-функции, оператора mutation, crossover реализовать master-slave с threadpool. Фитнес-функцию, mutation, crossover необходимо выполнять в slave-потоках.
- Для представления графа нужно использовать предоставленный модуль (https://gitlab.com/n-canter/openmp_dot_product/tree/master/graph).

Построить графики зависимости стоимости маршрута в популяции от номера итерации для графа с числом вершин равным 1000 и популяцией в 10000, 20000, 30000, ..., 100000 особей. На графике отобразить стоимость лучшего маршрута, худшего маршрута и среднюю стоимость маршрута в популяции на данной итерации.

Графики для разного числа особей строить отдельно.

Параметры запуска

Первым параметром указывается количество pthreads для slave-потоков (без учёта потока, который генерирует псевдослучайные числа и master-

потока). Т.е при $t = 2$ должно быть минимум 4 pthreads: master-поток, 2 slave потока, один поток для генерации псевдослучайных чисел.

Вторым параметром указывается число N – начальный размер популяции.

Третьим параметром указывается число S – параметр для критерия остановки.

У программы должно быть два режима работы: с автоматической генерацией графа (флаг `--generate`) и с чтением графа из файла (флаг `--file`).

После флага `--generate` следует целое число, количество вершин графа.

После флага `--file` следует путь к файлу, содержащему граф, в формате совместимом с графом описанным в репозитории.

В первой строчке файла `stats.txt` должны содержаться количество pthreads, N , S , количество вершин графа, количество итераций алгоритма, время работы программы и стоимость лучшего найденного маршрута.

Во второй строчке файла `stats.txt` должна содержаться последовательность вершин, соответствующая лучшему найденному маршруту.

Пример:

```
$ ./run 2 12 25 --generate 5
$ cat stats.txt
2 12 25 5 42 0.0001s 27
5 1 2 4 3

$ ./run 2 12 25 --file graph.txt
$ cat stats.txt
2 12 25 5 42 0.0001s 27
5 1 2 4 3
```

Примечание

В репозитории (readme.md) или в письме нужно указать, какие операторы crossover, selection, mutation были выбраны для решения задачи.

Программа при размере популяции 10^5 особей и графом с 10^3 вершин должна потреблять не более 2Гб памяти.