2 (маршрутная сеть гиф)

Нынешняя курсовая работа является логичным продолжение предыдущей, в которой я поставил цель разработать не нп-полный алгоритм решения задачи оптимизации маршрутной сети. Напомню, что маршрутная сеть - некий граф, расположенный в пространстве и имеющий понятие потока – транспорта, что по нему передвигается и собой нагружающий рёбраю

3 (слизевик гиф)

После исследования данной темы было решено использовать алгоритм физариум молд или же просто алгоритм слизевика. В заключении прошлой работы я оставил парочку идей для модернизации алгоритма

4 (фотки того, как всё было)

Среди них были указаны проблемы: все настройки сохранялись в просто в txt файле; результаты сохранялись в bmp файл 200 на 200; но главной проблемой было то, что итоговое решение об лучшей маршрутной сети было за человеком, который самостоятельно должен был просмотреть идущие друг за другом решения. Наличие пользователя в самом алгоритме явно лишнее

5 (файл с целями, задачами и актуальностью)

Так что были сформулированы следующие цели, чтобы решить эти проблемы алгоритма. Начнём

6 (сверху то, как он работает; снизу то, что получалось) 5 30

Так как результаты алгоритма слизевика – это множество точек в пространстве, нужно было перевести их в математический объект – граф. С этим мне помог алгоритм загрубления, уменьшаюший количество вершин графа, но сохраняющий его структурные свойства.

7 (Четыре типа аномалий; то, что получается в результате)

Однако в результате получалась маршрутная сеть с множеством различных аномалий, типы которых указаны слева и справа на слайде. Все они увеличивали количество вершин, но смысла не особо несли. Разработанный алгоритм минимизации решал эти проблемы, соединяя ненужные вершины. Конечно, вершины генераторов оставались нетронутыми

8 (Все три шага преобразования)

В итоге, из множества точек алгоритма слизевика через алгоритм загрубления и последующий алгоритм минимизации начала получаться нормальная такая маршрутная сеть. Теперь у нас есть математическое представление, а значит, можем перейти к цифрам. 6 10

9 (гифка с порталами)

Чтобы рассказать о метриках, должен напомнить главные потребности бизнеса в маршрутных сетях:

10 (картинки метрик и немного текста про каждую)

-они должны как можно меньше потратить на дороги  
-пути должны быть кратчайшими, чтобы всё было быстрее  
-сеть должна быть устойчива  
-сеть должна выдерживать как можно больший поток

Идеальное решение для первой задачи – минимальное оставное дерево, а для всех остальных – полный граф. Так что нам нужно искать оптимальное решение где-то по середине между этими тремя параметрами.  
Так что было решено создать вот такие метрики:

12 (картинки графов и их метрик)

И теперь имея математические значения нам не составит труда определить функцию сравнения на множестве решений, чтобы машина сама выполняла всю работу, предоставляя пользователю конечный результат – лучшую маршрутную сеть на данном этапе итерации. 7 10

13 (Картинки кода)

Супер, отлично, но теперь беда с интерфейсом. Код-то на С++. И все 1,5к строк алгоритмов крайне проблематично переносить на язык с удобным редактором интерфейса

14 (FLUTTER)

И тут на помощь пришёл FLUTTER - кроссплатформенный фреймворк для удобного создания приложений с красивым интерфейсом. Можно долго говорить про его инновационность, гибкость, довольство программистов, но самое главное, что я должен упомянуть – это наличие на нём библиотеки FFI, которая позволяет запускать синхронно, асинхронно или в другом потоке код на С++.

15 (Гифка готовой проги)

В итоге с лёгкость получилось обернуть весь имеющийся функционал приложения в красивый интерфейс, который легко масштабировать. Теперь, чтобы получить решение задачи построения маршрутной сети достаточно ввести точки на карту, настроить параметры и ждать в приятном глазу интерфейсе того, как программа выводит лучшее решение и его метрики.

16 (картинки результата слизевика, загрубления, минимизации и кусок программы, на фоне текста выполненных)

В итоге, за время курсовой работы были выполнены все поставленные цели при помощи математических алгоритмов на графах и Flutter. Но, конечно, должен сказать, что нет предела совершенству и даже сейчас есть много различных идей модернизации.

17 (конец)

Но пока что, на этом всё. Спасибо за внимание  
  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Пока что примерно 4 минуты, отлично