**СРАВНИТЕЛЕН АНАЛИЗ**

**ЗАДАЧАТА ЗА ТЪРГОВСКИЯ ПЪТНИК**

Един от най-известните и не напълно решени проблеми е този за задачата за търговския пътник. Той се състои в това да се намери път, който обикаля списък от градове, като минава през всеки град точно по веднъж и завършва в града, от който е започнало пътуването. Този проблем и днес е много актуален. Целта на проблема е този път да бъде с минимална дължина, т.е търсим най-краткия такъв път.

Проблемът намира широко приложение и днес, например в логистиката, транспорта, изкуствения интелект, машинното инженерство, математиката и софтуерното инженерство. Той е много ясно дефиниран, но е труден за решение, тъй като единственият абсолютно точен начин да се намери този път с минимална дължина изисква много ресурси, тоест той не е практически приложим. Този алгоритъм намира всички възможни пътица от град А, които преминават през всички останали градове точно по веднъж и приключват отново в град А. Но при N града всички възможни пътища са (N-1)! , което за малко на брой градове (напр. 3,4,..7) е добро решение, но когато градовете станат 50,100 това решение не е оптимално, тъй като не оправдава времето и паметта, необходими за неговото намиране.

Поради тази причина се използват различни евристични алгоритми за решение на проблема, които, разбира се, намаляват многократно времето за намиране на пътя, но се губи качеството на решението. Това е и въпросът, на който се търси най-добър отговор – добро съотношение на качеството на полученото решение и времето за получаването му.

Най-известните и точни алгоритми за решаване на задачата за търговския пътник са:

* Алгоритъм на най-близкия съсед (Nearest Neighbor)
* Genetic algorithm
* Greedy Heuristic Algorithm
* Christofides Algorithm
* Lin-Kernighan Algorithm

Алгоритмите, които съм избрал за реализацията на този проект са :

* Алгоритъм на най-близкия съсед (Nearest Neighbor)
* Намиране на минимално покриващо дърво (Minimal spanning tree) , така наречения ( Greedy Heuristic Algorithm).
* Например Christofides algorithm и Lin-Kernighan Algorithm дават много по-точно решение на проблема, но времето за тяхното изпълнение е повече в сравнение с това на избраните алгоритми. Докато алгоритмите Nearest Neighbor, Greedy Heuristic Algorithm и Genetic algorithm дават много по-бързо решение, но за сметка на това се получат много по-големи разминавания с най-късия път (например алгоритъмът за най-близкия съсед не намира винаги правилното решение и това може да се провери и види с просто око).

Например в следния граф след приложение на избраните алгоритми се намира пътят:

A ->C -> D -> B ->E -> A с дължина 1200, но лесно може да видим, че най-късият път е A->B->D->C->E->A с дължина 900.

200

200

100 700 300

100

100 600

200 400

Алгоритмите Christofides algorithm и и Lin-Kernighan Algorithm биха дали точно решение в случая.

Проблемът за търговския пътник винаги ще бъде актуален и няма как да има абсолютно точно и оптимално решение, защото най-точното решение е чрез намиране на всички пътища, което както вече споменахме не е оптимално решение заради ресурсите (памет и време), които се използват при повече на брой градове.

С избраните два алгоритъма за моя проект (Nearest Neighbor и Greedy Heuristic) и тяхната реализация искам да представя задачата за търговския пътник и проблемът в нея за намиране на най-кратък път.