**Софийски университет**

**„Св. Климент Охридски”**

## Факултет по математика и информатика

## Проект по дисциплината

## “****Извличане на знания от данни****”



Проект на:

Мария Пламенова Паскова фн. 61559

Николай Станиславов Стойков фн. 61587

Съдържание

[Факултет по математика и информатика 1](#_Toc441847343)

[Проект по дисциплината 1](#_Toc441847344)

[“Извличане на знания от данни” 1](#_Toc441847345)

[1. Мотивация и задача на курсовата работа 3](#_Toc441847346)

[1.1. Основни цели и задачи 3](#_Toc441847347)

[2. Визуализация на граф 3](#_Toc441847348)

[3. Проблем 3](#_Toc441847349)

[4. Подход и използвани алгоритми 4](#_Toc441847350)

[5. Реализация 4](#_Toc441847351)

[6. Бъдещи идеи за развитие 4](#_Toc441847352)

# Мотивация и задача на курсовата работа

### Основни цели и задачи

Графите са една незаменима част от математиката и информатиката. Изучаването им започва още от първи курс и продължава да се изучава още няколко години.

Предметите, които са пряко или косвено свързани с графите са много. На всеки един език може да се построи граф и да се решават задачи с него.

Ние решихме да направим визуализацията на граф по-добра, за да бъде от помощ на хората, които се опитват да реализират такива или чрез графи да представят някакви данни.

Целите, които си поставихме бяха:

* Визуализация на граф
* Изпълняване на алгоритми върху граф
* Подреждане на създаден граф

За да можем да подреждаме добре един граф, то трябва първо той да се изрисува от потребителя. Това беше първата част от проекта.

# Визуализация на граф

За визуализацията на графа използваме canvas, в който потребителят може да изчертае върховете и ребрата на графа.

Има възможност за ръчно преместване на върхове.

# Проблем

Ако построи голям граф с много пресичания, то той ще бъде трудно възприеман от потребителя.

Това е причината да решим, че би било хубаво ако графът изглежда добре. Под добре изглеждащ граф ние дефинираме – минимално пресичащи се ребра, нодове и възприемчиво разстояние между съседните върхове.

# Подход и използвани алгоритми

Решението, с което ние подходихме е да приложим генетичен алгоритъм с някой модификации. Най-напред започнахме с оптимизиране на броя на пресичанията между ребрата на графа, но това не доведе до много добри резултати тъй като имаше проблеми с пресичането на ребра и върхове. За да решим този проблем приложихме оптимизация и върху това да се минизира броя на пресичанията между върховете и ребрата. Но освен това имаше и още нещо, с което трябваше да се справим – пресичането между самите върхове (върховете са кръгове).

Една от мофидикациите, които приложихме бе да използваме мутацията като момент за подобряване на графа, а не просто рандомизиране на позициите и и вкарване на нови гени. Подобряването на графа се крие в това че при мутация ние проверяваме кои върхове са „проблемни“ и на базата на това ние знаем кои върхове би било най-подходящо да разместим.

Интересен момент е и генерирането на деца от 2-ма произволни родители. В началото, когато генерираме популацията създаваме много наброй графи с различни позиции за техните върхове. При генериране на децата се взимат позиции на върхове от 2-та родителски графа и тези позиции биват използвани за други върхове от дъщерния граф. Така едновременно взимаме „гени“ от родителите и променяме детето, но и вкарваме разновидности.

# Реализация

# Бъдещи идеи за развитие

Търсене на мост в граф и разделяне на подграфи.

Търсене на най-дълъг път в граф.

БФС