ALGORYTMY DECYZYJNE
I
TEORIA ZŁOŻONOŚCI

Problem maksymalnej kliki
Proffesional Klika Solver - dokumentacja

Autorzy:
Michał Pieróg
Jakub Jaśkowiec
Witold Baran

1.Zagadnienie projektu

1.1. Temat : Znalezienie maksymalnej kliki.

Projekt polegał na stworzenie programu umożliwiającego znalezienie maksymalnej kliki w zadanym grafie, za pomocą algorytmu genetycznego. Program 'Proffesional klika solver' został napisany w języku java. Skorzystaliśmy z biblioteki JUNG, umożliwiającej w łatwy sposób tworzenie i rysowanie grafów.

1.2. Algorytmy genetycznie.

Problem definiuje środowisko, w którym istnieje pewna populacja osobników. Każdy z osobników ma przypisany pewien zbiór informacji stanowiących jego genotyp, a będących podstawą do utworzenia fenotypu. Fenotyp to zbiór cech podlegających ocenie funkcji przystosowania modelującej środowisko. Innymi słowy - genotyp opisuje proponowane rozwiązanie problemu, a funkcja przystosowania ocenia, jak dobre jest to rozwiązanie.

Genotyp składa się z chromosomów, gdzie zakodowany jest fenotyp i ewentualnie pewne informacje pomocnicze dla algorytmu genetycznego. Chromosom składa się z genów.

Wspólnymi cechami algorytmów ewolucyjnych, odróżniającymi je od innych, tradycyjnych metod optymalizacji, są:

1. stosowanie operatorów genetycznych, które dostosowane są do postaci rozwiązań,
2. przetwarzanie populacji rozwiązań, prowadzące do równoległego przeszukiwania przestrzeni rozwiązań z różnych punktów,
3. w celu ukierunkowania procesu przeszukiwania wystarczającą informacją jest jakość aktualnych rozwiązań,
4. celowe wprowadzenie elementów losowych.

Najczęściej działanie algorytmu przebiega następująco:

1. Losowana jest pewna populacja początkowa.
2. Populacja poddawana jest ocenie (**selekcja**). Najlepiej przystosowane osobniki biorą udział w procesie reprodukcji.
3. Genotypy wybranych osobników poddawane są operatorom ewolucyjnym:
    1. są ze sobą kojarzone poprzez złączanie genotypów rodziców (**krzyżowanie**),
    2. przeprowadzana jest mutacja, czyli wprowadzenie drobnych losowych zmian.
4. Rodzi się drugie (kolejne) pokolenie i algorytm powraca do kroku drugiego, jeżeli nie znaleziono dostatecznie dobrego rozwiązania. W przeciwnym wypadku uzyskujemy wynik.

2. Szczegółowe informacje na temat algorytmu genetycznego i heurystycznego zawartego w programie

Algorytm genetyczny przebiega w następujących etapach:

1. **Definicja osobników**

Osobniki reprezentowane są przez chromosomy. Każdy chromosom składa się z ilości pól równej liczbie wierzchołków całego grafu. Kodowanie chromosomu jest następujące: „1” oznacza, że dany wierzchołek znajduje się w podgrafie, „0” oznacza, że tego wierzchołka nie ma w podgrafie.

1	0	0	1	0	1	0	1

Przykładowy chromosom dla grafu z 8 wierzchołkami

2. Populacja początkowa

Populacja początkowa (kodowanie chromosomów) jest generowana losowo. Domyślny rozmiar populacji początkowej jest uzależniony od wielkości grafu – im większy graf, tym większa populacja. Jej rozmiar można także zmieniać z poziomu GUI.

3. Ocena dostosowania

Ocena dostosowania zwiększa się wraz ze wzrostem ilości wierzchołków oraz krawędzi podgrafu, który tworzy osobnik. Dodatkowo, karane są osobniki, które nie są klikami (przyznawane im są ujemne punkty w zależności od ilości krawędzi, których brakuje im do stworzenia kliki), a premiowane te, które są klikami. Uproszczony wzór wygląda następująco:

*rating=graph.getEdgeCount() - penalty +g.getVertexCount()*premiumWeight*

Gdzie:

graph.getEdgeCount() – liczba krawędzi w całym grafie, zmniejsza prawdopodobieństwo że funkcja przyjmie wartość ujemną.

penalty – kara, równa liczbie krawędzi brakujących osobnikowi do bycia kliką

g.getVertexCount() – liczba wierzchołków w podgrafie, premiuje większe grafy

premiumWeight – waga premii, zależna od tego czy osobnik jest kliką czy nie. W przypadku nie bycia kliką waga jest bardzo niska.

4. Selekcja

Możliwe są 3 rodzaje selekcji (można je zmieniać z poziomu GUI programu):

- Selekcja turniejowa

  Polega na „wymieszaniu” populacji i podzieleniu jej na małe grupy najczęściej złożone z czterech osobników. Z każdej grupy wybiera się dwóch osobników o najwyższym dostosowaniu. Pary wybranych w ten sposób osobników stanowią rodziców przyszłego potomstwa.

- *Selekcja ruletkowa*

  Im większa wartość funkcji dostosowania osobnika, tym większe prawdopodobieństwo, że osobnik stanie się on rodzicem. Do selekcji wybieranych jest n-par, gdzie n – rozmiar populacji.

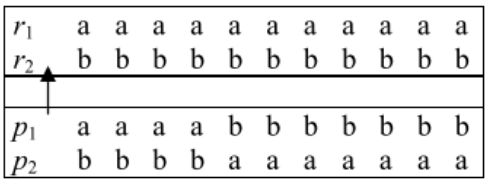
- *Selekcja „Harem”* (autorska)

  Najlepiej dostosowany osobnik krzyżowany jest z każdym innym osobnikiem, potem drugi najlepiej dostosowany krzyżowany jest z gorszymi od niego, i tak do momentu, gdy wybranych jest n nowych par. Ta selekcja wpływa na dużą zbieżność algorytmu, wiec powinna być stosowana tylko w niektórych przypadkach, lub z dużą mutacją.

5. Krzyżowanie

Dostępne są 4 metody krzyżowania osobników (można je zmieniać z poziomu GUI programu):

- *Krzyżowanie jednopunktowe z dwoma potomkami*

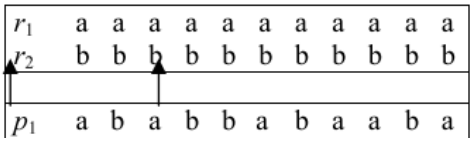


W górnej części rysunku przedstawiono parę rodziców (r1, r2). Losowo wybierany jest punkt rozcięcia łańcuchów. Pierwszy potomek powstaje przez kopiowanie genów pierwszego rodzica do punktu rozcięcia; resztę genów pobiera się od drugiego rodzica. Drugi potomek tworzony jest w analogiczny sposób z tym, że początkowy segment tworzą geny drugiego rodzica, a końcowy – geny pierwszego rodzica.

- *Krzyżowanie jednopunktowe z jednym potomkiem*

Podobne to powyższego z tą różnicą, że powstaje 1 potomek (p1)

- *Krzyżowanie wielopunktowe z jednym potomkiem*



Każdy gen jest z jednakowym prawdopodobieństwem wybierany od obu rodziców,
tzn. przed wypełnieniem kolejnej pozycji rzucamy monetą: „orzeł” nakazuje pobrać gen od pierwszego rodzica, „reszka” – od drugiego.

- *Krzyżowanie wielopunktowe ważone z jednym potomkiem* (autorskie)

Podobne do powyższego z tą różnicą, że prawdopodobieństwo, czy gen zostanie pobrany od pierwszego czy od drugiego rodzica, jest wprost proporcjonalne do wartości funkcji dostosowania każdego z rodziców.

6. Mutacja

Mutacja polega na zamianie genu „1” na gen „0” (i odwrotnie). Prawdopodobieństwo mutacji można zmieniać z poziomu GUI. Mutacji podlegają wszystkie geny wszystkich osobników.

7. Wprowadzanie nowych osobników do populacji

Rozmiar populacji jest utrzymywany na stałym poziomie. Dostępne są 2 metody zastępowania (można je zmieniać z poziomu GUI programu):

- *zastępowanie według absolutnego dostosowania*

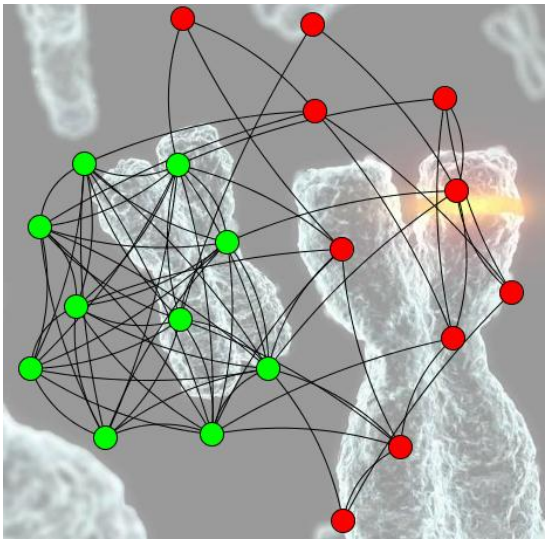
  najsłabiej dostosowane osobniki starej populacji zastępowane są przez nowe osobniki.

- *losowa*

  wstawianie potomka w losowo wybrane miejsce

8. Zakończenie algorytmu:

- Algorytm może zakończyć się w 3 sytuacjach: jeśli wartość funkcji przystosowania najlepszego osobnika nie zmienia się przez 100 iteracji; jeśli liczba iteracji przekroczy ustawione przez użytkownika maksimum; lub jeśli użytkownik naciśnie przycisk „break”.
- W trakcie działania algorytmu, osobnik z najlepszym dostosowaniem jest zapamiętywany i jest wyświetlany jako rozwiązanie w przypadku zakończenia działania algorytmu.



Zakończenie działania algorytmu i podświetlenie najlepszego znalezionego osobnika.

Algorytm heurystyczny przebiega w następujących etapach:

1. Każdemu wierzchołkowi przyporządkowuję 'wagę' - dzieląc ilość jego krawędzi przez ilość wierzchołków
2. Tworze podgraf (o najmniejszym rozmiarze)
3. Do podgrafu wkładam wierzchołki o najwyższej wadze
4. Sprawdzam czy ten podgraf jest klika, jeśli nie, to wymieniam jego losowy wierzchołek na inny, skok do pkt 4
5. Jeśli podgraf jest klika, to zwiększam rozmiar podgrafu o 1, skok do pkt 3.

3.Informacje o programie

3.1. Informacje ogólne

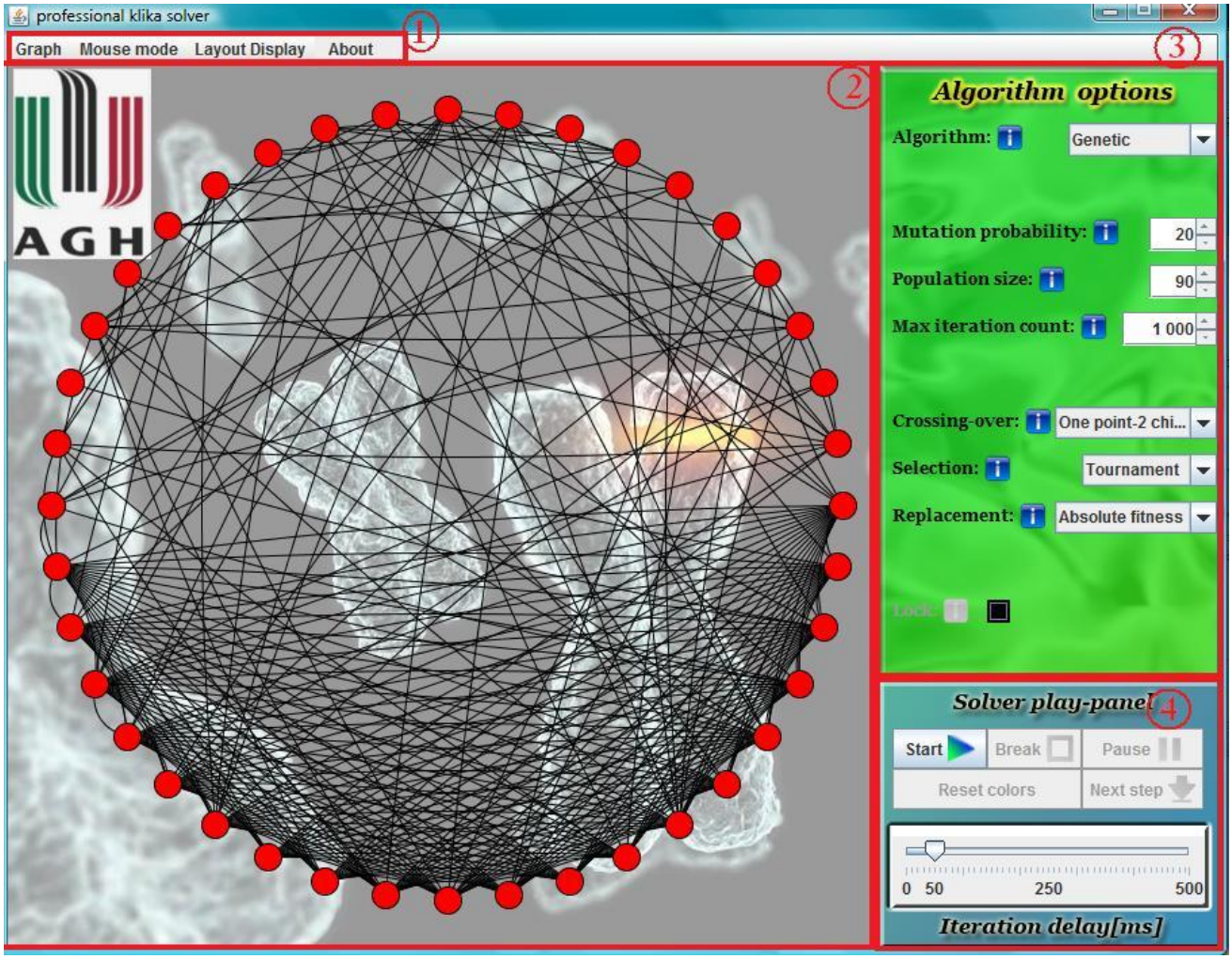
Program umożliwia ręczne, a także losowe generowanie grafu jak i również wczytywanie oraz zapis gotowego grafu do popularnego formatu .graphml. Po uruchomieniu programu automatycznie generowany jest graf, a program jest już gotowy do użytku.

Program dzieli się na kilka podstawowych okien:
-Główne okno programu – służy do wizualizacji rozwiązania, zmieniania parametrów algorytmu, wyświetlania grafu, edycji grafu oraz zapewnia dostęp do wszystkich opcji zawartych w programie.
-Okno generatora grafów – służy do wprowadzania parametrów generowanego grafu
-Okno przystosowania – wyświetla wykres wartości funkcji przystosowania w kolejnych iteracjach.

3.2. GUI

Graficzny Interfejs Użytkownika jest zaprojektowany w sposób przejrzysty i intuicyjny. Umożliwia łatwe i wygodne korzystanie ze wszystkich funkcji programu. Po uruchomieniu programu otwarte zostanie Główne okno programu.

3.2.1 Główne okno programu

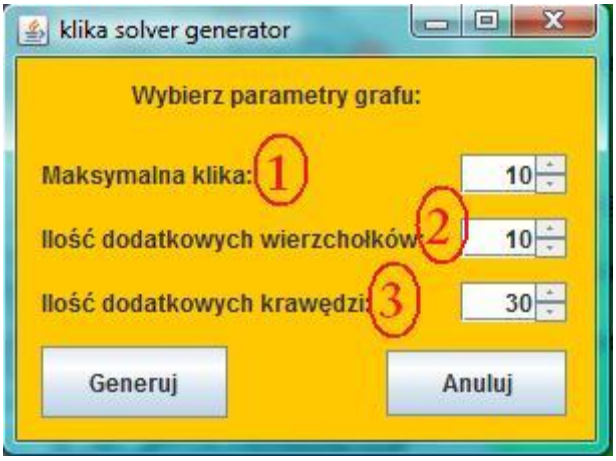


Główne okno programu dzieli się na kilka paneli (zaznaczone kolorem czerwonym):
1 – Menu programu
2 – Panel wizualizacji/edycji grafu
3 – Panel opcji algorytmu
4- Panel kontroli solvera

3.3. Menu programu

3.3.1. Zakładka Graph – znajdują się tutaj podstawowe opcje do tworzenia, wczytywania i zapisu grafu.

Tworzenie grafów:
Tworzenie ręczne – *Graph-> Create new*
Po wybraniu tej opcji włączy się tryb edycji. By stworzyć wierzchołek po prostu klikamy w miejsce na polu edycji gdzie chcemy żeby powstał. By stworzyć krawędź, klikamy na jeden z wierzchołków, przytrzymujemy przycisk myszy i przeciągamy na drugi wierzchołek. Po puszczeniu przycisku powstanie krawędź między nimi.
Tworzenie losowe – *Graph-> Generate random*
Po wybraniu tej opcji pojawi się okno z parametrami nowego grafu. Ustalamy jak maksymalna klika ma się w nim znajdować, ile dodatkowych wierzchołków i ile dodatkowych krawędzi. Po naciśnięci przycisku „Generuj" w polu edycji pojawi się gotowy, losowo wygenerowany na podstawie parametrów graf.



Okno umożliwiające wygenerowanie losowego grafu. Generowane grafy są spójne, bez pętli.

1) Maksymalna klika – oznacza liczbę wierzchołków w klice którą będzie zawierał graf, rozmiar kliki może być większy przy podaniu dużej liczby krawędzi, jednak nigdy nie będzie mniejszy od wprowadzonej wartości.

2) Ilość dodatkowych wierzchołków – liczba wierzchołków nie należących do kliki

3) Ilość dodatkowych krawędzi – liczba krawędzi łączących wierzchołki nie należące do kliki. Jeśli jest mniejsza niż liczba krawędzi potrzebnych do utworzenia grafu spójnego zostaną dodane brakujące krawędzie.

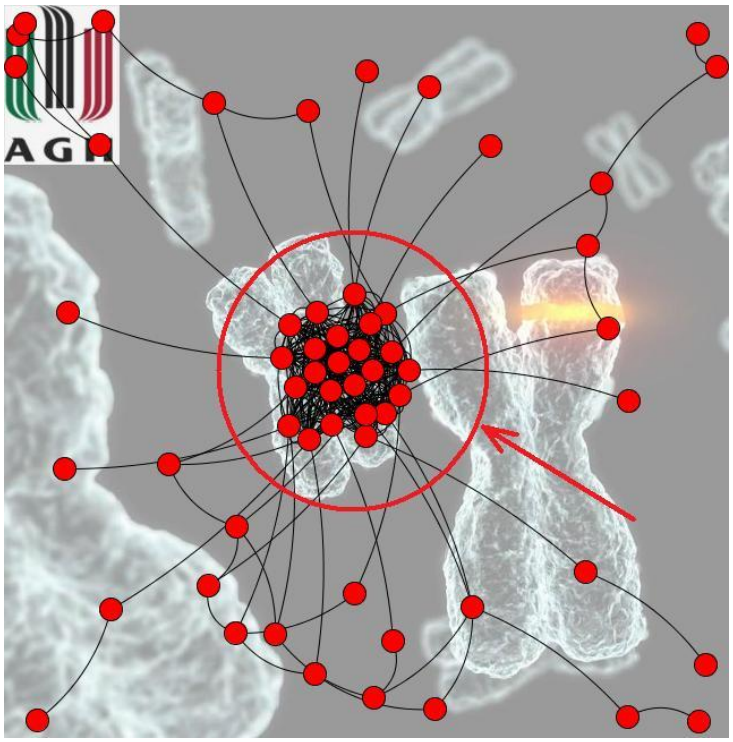
Wczytywanie/Zapis
Opcje wczytywania i zapisu znajdują się w zakładce *Graph.*
Wczytywanie- *Graph->Load.*
Po wybraniu tej opcji pojawi się dobrze znane okno dialogowe gdzie możemy wybrać plik z grafem do wczytania. Domyślnym folderem w którym przechowywane są grafy związane z programem jest folder *Graphs* znajdujący się w folderze programu.
Zapis- *Graph->Save.*
Po wybraniu tej opcji zostaniemy poproszeni przez okno dialogowe o podanie docelowego miejsca zapisu. Domyślnie jest nim folder *Graphs.*

3.3.2. Mouse mode
Umożliwia zmiane trybu myszki.
Tryb *Edition* – włączany w celu dodawania/usuwania krawędzi i wierzchołków. Domyślnie aktywny podczas tworzenia nowych grafów.
Tryb *Viev Mode* – umożliwia jedynie zmianę położenia grafu w oknie edycji.
Tryb *Picking Mode* – umożliwia przemieszczanie pojedynczych wierzchołków względem okna edycji.

3.3.3. Layout Display

Zakładka umożliwia wybór Layoutu. Layout określa w jaki sposób wierzchołki i krawędzie są rozmieszczane w Panelu wizualizacji. Dostępne Layouty:

FR Layout

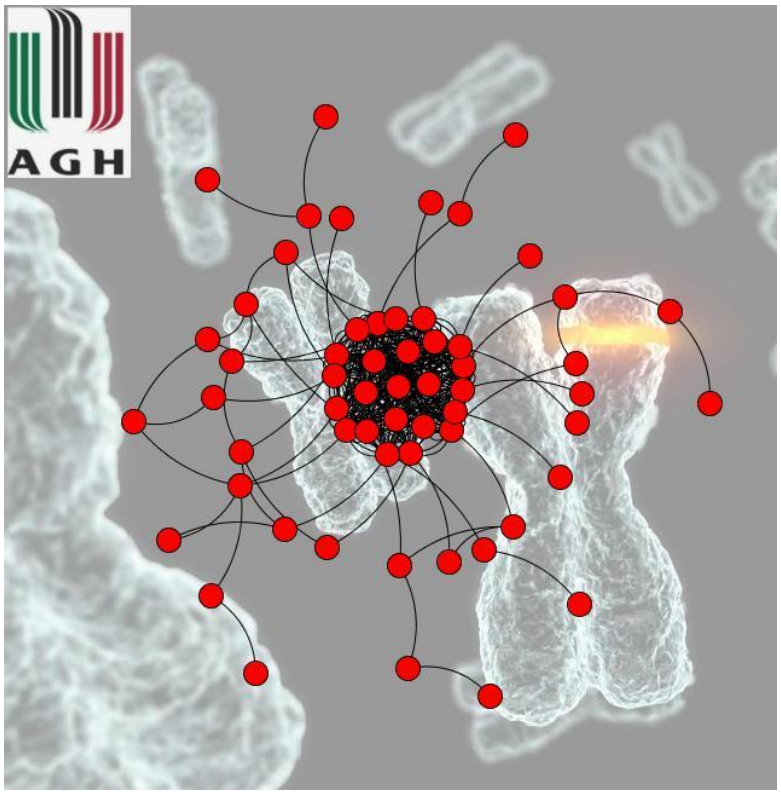


FR Layout- layout wykorzystuje algorytm Fruchterman'a-Reingold'a typu force-directed.

Zachowanie jest uzależnione od:
współczynnika atrakcji – krawędzie utrzymują wierzchołki blisko siebie
współczynnik odpychania – wierzchołki odpychają się od siebie.
Im większa liczba krawędzi, tym bliżej siebie znajdują się wierzchołki. W przypadku grafu z kliką oraz z wierzchołkami z mniejszą liczbą krawędzi, wierzchołki należące do kliki będą leżały blisko siebie.

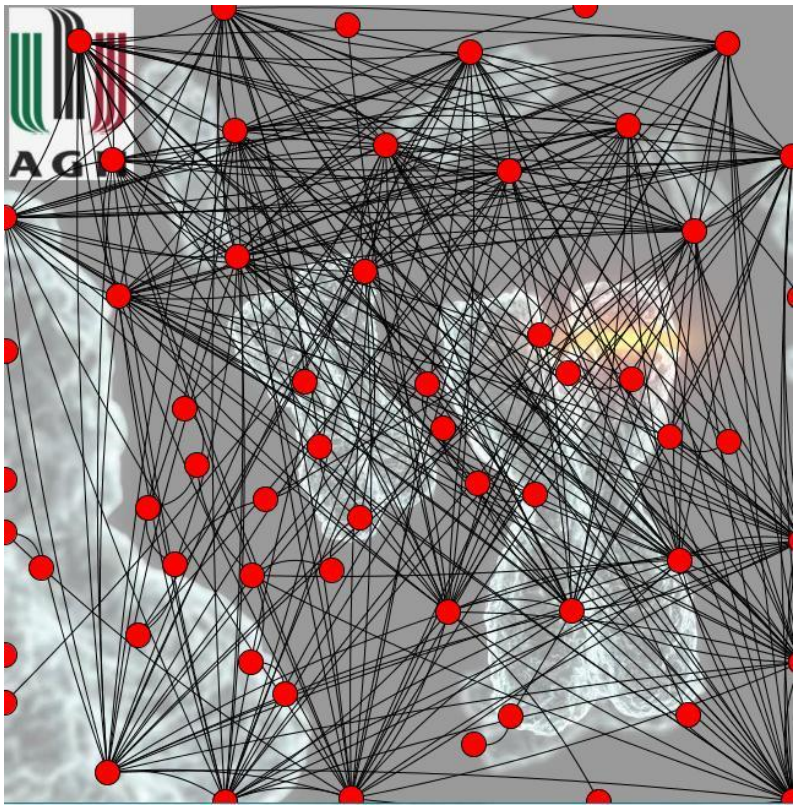
Klika znajduje się w czerwonym okręgu, a wierzchołki z małą liczbą krawędzi poza nim.

KK layout



KK layout- layout bazujący na algorytmie Kamada-Kawai. Tak jak algorytm FR jest algorytmem typu force-directed. Dąży do rozmieszczenia wierzchołków by uzyskać graf o minimalnej „energii". Podobnie jak w przypadku algorytmu FR wierzchołki należące do kliki leżą blisko siebie.

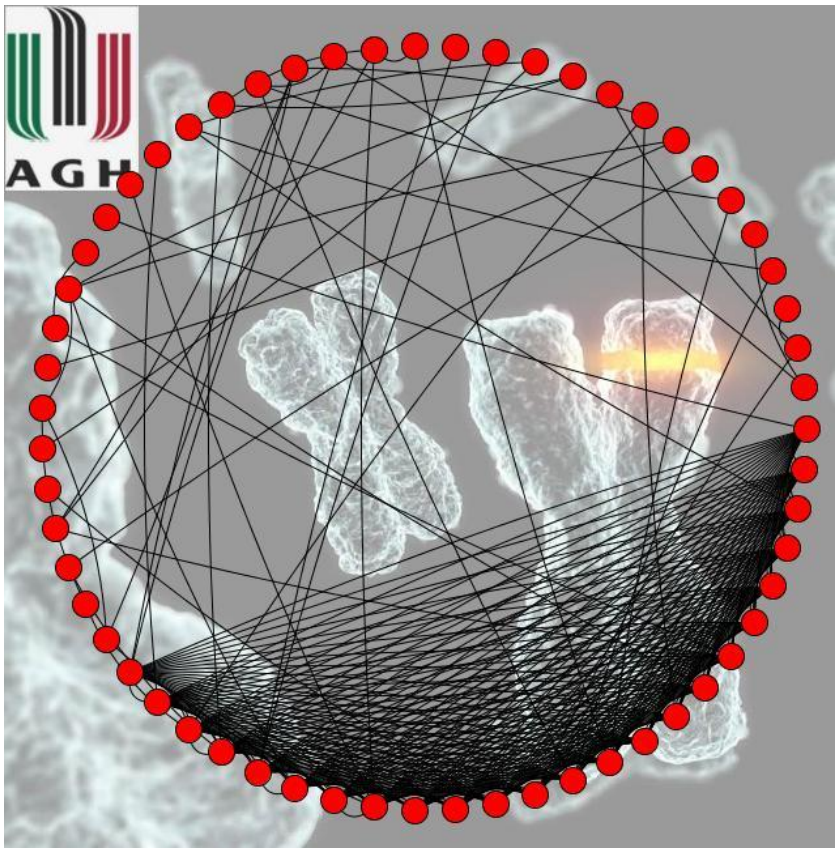
Spring Layout oraz Spring Layout2



Spring Layout oraz Spring Layout2 są do siebie bardzo podobne. Krawędzie mają nadane współczynniki sprężystości, dzięki czemu wierzchołki mogą się poruszać. Symuluje to połączenie sprężyną. Przesuwając jeden wierzchołek przesuwają się również wierzchołki z nim połączone.

Layout ten może wyglądać efektywnie w przypadku małych grafów, które tworzy użytkownik, jest jednak mało czytelny w przypadku większych grafów.

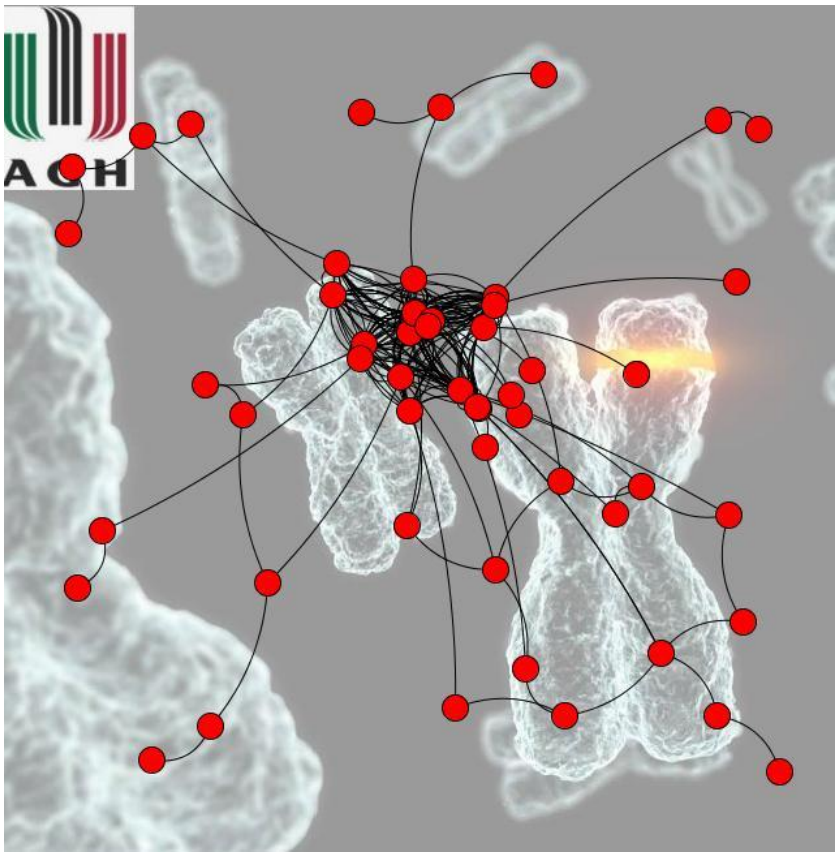
Circle Layout



Circle Layout – layout kołowy. Wszystkie wierzchołki leżą na obwodzie koła, a krawędzie wewnątrz niego. Potrzebuje najmniej zasobów komputera, gdyż do rozmieszczenia wierzchołków nie są wykorzystywane skomplikowane algorytmy. W przypadku grafów zawierających klikę oraz wierzchołki ze znacznie mniejszą liczbą krawędzi jest bardzo dobrze czytelny.

Wierzchołki należące do kliki nie muszą leżeć koło siebie, wierzchołki są wyświetlane w kolejności dodania do grafu i nie zależą od innych czynników.

ISOM Layout



ISOM Layout – layout oparty o samoorganizującą mapę, bazującą na samoorganizujących metodach Meyer'sa dla grafów, wykorzystujących sieci neuronowe.

Wierzchołki należące do kliki leża blisko siebie, ale niektóre mogą być niewidoczne gdyż nachodzą na siebie.

3.4. Panel wizualizacji/edycji grafu

Panel zapewnia interakcje między użytkownikiem a grafem. W trakcie działania solvera oraz po jego zakończeniu na zielono zaznaczone są wierzchołki tworzące najlepszego osobnika.
W różnych trybach działania myszki umożliwia inne interakcje z użytkownikiem. Sposób w jaki wyświetlany jest graf, zależy od wybranego Layout'u. Wszystkie dostępne Layout'y zostały opisane szczegółowo w poprzednim rozdziale.

3.5. Panel opcji algorytmu

Panel opcji algorytmu zawiera wszystkie dostępne właściwości algorytmu, które użytkownik może modyfikować. Dostępne opcje:

Algorithm –wybór algorytmu, można wybrać algorytm genetyczny lub heurystyczny. W przypadku algorytmu heurystycznego nie jest możliwa modyfikacja innych opcji, gdyż algorytm ich nie wykorzystuje. Również nie jest wyświetlane okno przystosowania, gdyż algorytm nie korzysta z funkcji przystosowania. Algorytm genetyczny posiada domyśle ustawienia, które są ładowane wraz ze startem programu.

Mutation probabilisty – prawdopodobieństwo mutacji wyrażone w promilach.

Population size – rozmiar populacji, stała ilość osobników w trakcie pracy solvera

Max iteration count – liczba iteracji po których wykonanie algorytmu zostanie wstrzymane

Crossing-over – wybór metody krosowania osobników, do wyboru jednopunktowe, dwupunktowe, ważone dwupunktowe, jednopunktowe z dwoma potomkami.

Selection – sposób selekcji osobników do puli rodzicielskiej, dostępne: ruletkowa (zmodyfikowana), turniejowa, haremowa.

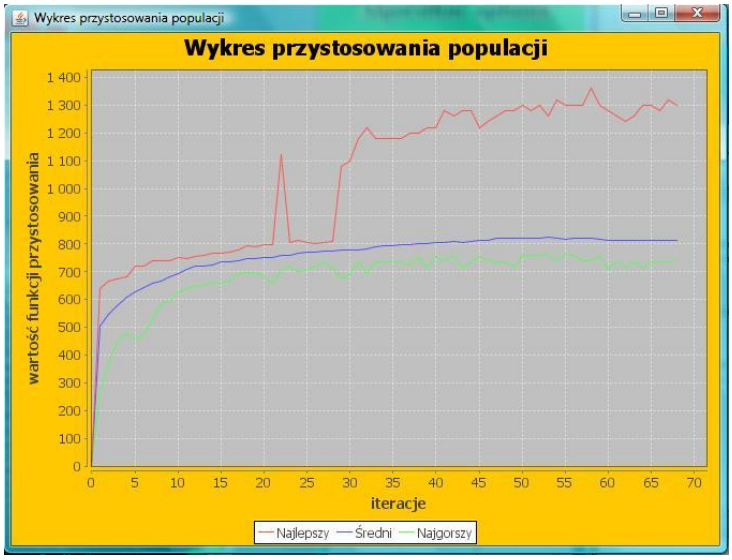
Replacement - metoda wprowadzania potomków do populacji, stare osobniki są zastępowane przez nowe. Do wyboru zastępowanie losowe lub według dostosowania (najgorsze osobniki są zastępowane).

Lock – zabezpieczenie przed przypadkowym zmienieniem opcji algorytmu. Opcje algorytmu genetycznego mogą być zmieniane w trakcie jego pracy, po uprzednim odznaczeniu pola Lock. Jednak wprowadzenie populacji początkowej nadal będzie nie możliwe, aby ją zmienić należy przerwać wykonanie algorytmu. Blokada wprowadzania opcji jest aktywowana automatycznie po uruchomieniu solvera i dezaktywuje się po zakończeniu pracy solvera.

3.6. Panel kontroli solvera

Panel kontroli solvera służy do kontroli przebiegu wykonywania algorytmu.
Przycisk start uruchamia solver lub wznawia jego działanie po użyciu pauzy.
Break –przerywa natychmiastowo wykonanie algorytmu.
Pause- umożliwia tymczasowe wstrzymanie solvera. Jego wykonanie może zostać wznowione po wciśnięciu przycisku Start.
Next step- w trybie wstrzymania algorytmu umożliwia wykonanie pojedynczej iteracji.

3.7. Okno przystosowania



Po uruchomieniu algorytmu genetycznego, zostanie wyświetlony okno przystosowania populacji z wykresem. Wykres jest aktualizowany po każdej iteracji. Wykres można przybliżać rozciągając zaznaczenie na części wykresu, którą chcemy zbliżyć. Cofamy przybliżenie w taki sam sposób.

Przedstawia on wykres zależności funkcji przystosowania od iteracji dla najlepszego i najgorszego osobnika, oraz medianę przystosowania.

4. Autorzy

Osoby odpowiedzialne za projekt:

Dokumentacja - tworzona wspólnie

Jakub Jaśkowiec - implementacja klas realizujących algorytm genetyczny oraz heurystyczny, generator grafów,

Michał Pieróg - klasy odpowiedzialne za synchronizację i współdziałanie klas, główne okno programu, wizualizacja i edycja grafów,

Witold Baran - Operacje zapisu/odczytu grafów, GUI generatora grafów oraz wykres funkcji przystosowania.