**Projekt 1**

**NEH:**

Zadania są sortowane od tego o najkrótszym czasie do tego o najdłuższym czasie wykonania. Następnie za pomocą **funkcji firstTwoTask** tworzona jest ramka danych z pierwszych dwóch zadań o licznym najkrótszym czasie wykonania. Następnie każde kolejne zadanie jest wstawiane w miejsce przed, miedzy lub za wszystkimi zadaniami, zależnie w którym miejscu jest najkrótszy liczny czas wykonania zadań.

**Funkcja firstTwoTaska** tworzy ramkę danych z pierwszych dwóch zadań.

**Funkcja addTask** dodaje kolejne zadania i zwraca połączone dane i pozostałe dane do dodania

Badane parametry: brak

**HillClimbing:**

Pierwsze rozwiązanie wybierane jest losowo za pomocą **funkcji getRandomSolution.**

Następnie obliczany jest czas obecnego rozwiązania, następnie za pomocą funkcji getNeighbours wybieramy sąsiadów i z nich za pomocą getBestNeighbour najlepszego sąsiada o najkrótszym czasie rozwiązania. Sąsiedzi tworzeni są za pomocą zamiany dwóch zadań miejscami, kolejno 1 z 2, 1 z 3, 1 z 4, 2 z 3, 2 z 4 itd

Jeśli proces z zamieniona kolejnością dwóch zadań ma lepszy czas jest ono zapisywane jako nowe rozwiązanie i proces trwa aż do momentu, gdy algorytm nie będzie się już poprawiał.

Dodatkowo stworzona jest funkcja multiStart, która przyjmuje 3 parametry, dane, wielkość sąsiedztwa oraz liczbę startów. Działa on w taki sposób, że określoną liczbę startów wykonuje algorytm wspinaczki i wybiera rozwiązanie o najkrótszym czasie.

Badane parametry:

-n - wielkość sąsiedztwa ,

-liczba startów (multiStart)

Wnioski:

Im większa liczba startów tym lepszy wynik, im większe sąsiedztwo tym lepszy wynik. Jednak większe sąsiedztwo = dłuższy czas wykonania.

**Algorytm wyżarzania**

Parametry:

data-dane

T-temp początkowa

F - temperatura zamrożenia

L-ilość powtórzeń pętli

r-współczynnik wygaszania

n-wielkość sąsiedztwa

Pierwsze rozwiązanie jest wybierane losowo, następnie dopóki ustalona temperatura - T nie spadnie poniżej temperatury zamrożenia F, przez określona liczbę razy - L tworzone jest nowe rozwiązanie (poprzez wylosowanie jakiegoś sąsiedniego rozwiązania) i obliczane jest różnica miedzy czasem nowego rozwiązania a czasem starego rozwiązania. Jeśli nowe rozwiązanie jest lepsze to jest ono przyjmowane jako rozwiązanie i pętla jest wykonywana ponownie. Natomiast jeśli jest ono gorsze to obliczana jest wartość e do minus różnica czasów przez T. następnie losowana jest liczba losowa z zakresu od 0 do 1 i jeśli obliczona wartość jest większa od wylosowanej liczby to gorsze rozwiązanie jest przyjmowane jako rozwiązanie. W przeciwnym przypadku zostajemy przy aktualnym rozwiązaniu. Następnie wartość T jest aktualizowana poprzez wymnożenie przez r - współczynniki wygaszania. Proces jest powtarzany do momentu aż T spadnie poniżej F.

Badane parametry:

-wielkość sąsiedztwa

-temperatura początkowa

-współczynnik wygaszania

Wnioski:

Im większa liczba sąsiadów tym lepsze wyniki, temperatura początkowa i wsp wygaszania nie wykazały zauważalnych trendów.

**Algorytm tabu:**

Parametry:

data-dane

-n-wielkość sąsiedztwa

-s-długość listy tabu

-iteratywna - liczba ile razy wykonywana jest dana pętla

Pierwsze rozwiązanie jest wybierane losowo. Następnie jest ono wyznaczane jako najlepsze rozwiązanie. Przez określona liczbę iteracji generujemy sąsiednie rozwiązania, następnie wybierane jest najlepsze z tych rozwiązań. Następnie sprawdzane jest czy rozwiązanie znajduje się na liście tabu. Jeśli jest to z listy sąsiadów wyrzucane jest rozwiązanie z listy tabu i proces powtarzany, aż wybranego rozwiązania nie będzie na liście tabu.

Następnie rozwiązanie zapisywane jest jako nowe rozwiązanie i sprawdzane jest czy jest lepsze od starego rozwiązania. Jeśli jest lepsze to jest zapisywane jako nowe najlepsze rozwiązania. Na koniec nowe rozwiązania jest dodawane do listy tabu i sprawdzane jest czy lista tabu nie jest zbyt długa, wtedy ta zamiana, która na liście tabu jest najdłużej jest z niej usuwana. Proces jest powtarzany określona liczbę razy.

Na liście tabu są pary numerów indeksów poszczególnych zadań, które zostały zamienione wcześniej.

Badane parametry:

-wielkość sąsiedztwa,

-długość listy tabu,

-liczba iteracji.

Wnioski:

Największy wpływ ma wielkość sąsiedztwa, im większe sąsiedztwo tym lepsze wyniki, Długość listy tabu tez jest istotna, gdy jest zbyt krótka możemy wykonywać w kolko te same rozwiania, zbyt długa pomijać najlepsze rozwiązania. Liczba iteracji również jest istotna, gdy algorytm wykona się zbyt mało razy możliwe, ze nie poprawi się maksymalnie.

**Algorytm genetyczny:**

Parametry:

data-dane

iterations-liczba iteracji

population- wielkość generowanej populacji

crosses-liczba krzyżowań w jednej iteracji

mutations-liczba mutacji w jednej iteracji

maskLen-dlugosc maski (długość genu przekazywanego od rodzica potomkowi)

Generowane są rozwiązania losowe (tyle ile wynosi wielkość populacji), nastepnie wykonywana jest określona liczba krzyżowań (crosses). Polegają one na wyborze dwóch rodziców za pomocą turnieju (tournament - polega on na podzieleniu populacji na dwie grupy i z obu grup wybraniu najlepszego osobnika), a nastepnie są oni krzyżowani ze sobą za pomocą ordered crossover. Polega ono na rozcięciu rodziców w losowo wybranym miejscu na geny o określonej długości. Nastepnie zadania z drugiego rodzica są przepisywane do pierwszego rodzica, zaczynając od drugiego punktu ciecia, pomijane są oczywiście zadania będące już w rodzicu. Nastepnie z otrzymanych rozwiązań wybierana jest określona ilość najlepszych rozwiązań (populations), w których dochodzi do określonej liczby mutacji (mutations). Podczas jednej mutacji wybierane jest losowe rozwiązania, w którym zamieniane miejscami są 2 losowe zadania. Cały proces powtarzany jest na populacji określona liczbę razy (iterations).

Badane parametry:

-wielkość populacji,

-liczba krzyżowań

-długość genotypu przekazywanego potomkowi

Wnioski:

Duża populacja dobrze wpływa na otrzymane wyniki, znacząco je poprawia. Liczba krzyżowań także dobrze wpływa na otrzymane wyniki, im więcej krzyżowań tym więcej możliwości powstania dobrych potomków. Długość genu przekazywanego potomkom nie może być natomiast zbyt krótka ani zbyt długa, aby ani nie przekazać zbyt mało ani zbyt dużo cech rodzica.