**Optymalizacja Kombinatoryczna – laboratoria**

Wykonali:

Marcin Staszak 127241 I5

Piotr Furmankiewicz 127282 I5

**Temat projektu:**

Problem 1

Flowshop, liczba maszyn m=2, liczba zadań n,

operacje niewznawialne,

dla pierwszej i drugiej maszyny po k okresów przestoju (na każdą maszynę),

o losowym czasie rozpoczęcia i trwania (określonym przez generator instancji problemu), okresy te nie mogą się na siebie nakładać (tj. czas przestoju

na I maszynie nie może nakładać się z czasem dowolnego przestoju

na II maszynie), k >= n/10,

*minimalizacja sumy czasów zakończenia wszystkich operacji*

**Obsługa programu:**

Ścieżka do gotowego projektu w Visual Studio:

OK/projekt\_ok/projekt\_ok.sln

Wynik działania programu zapisywany jest do pliku: OK/Solutions/Solutionbest.txt

Wygenerowane instancje:

OK/Instances/

**Opis działania algorytmu:**

**Generator instancji** losuje czas trwania operacji pierwszej oraz drugiej dla danego zadania z przedziału [1,ustalona wartość], „ready time” dla pierwszej operacji z przedziału [0,ustalona wartość] (dla drugiej nie, bo „ready time” dla operacji drugiej to czas zakończenia operacji pierwszej). Następnie losuje maintenance’y dla pierwszej i drugiej maszyny w taki sposób, że się na siebie nie nakładają. Czas i liczba maintenance’ów jest ustalona. Powyższe operacje generator wykonuje tyle razy, ile jest zadań.

**Generator rozwiązań losowych** dla każdej z maszyn wstawia najpierw maintenance’y, a następnie w pierwsze wolne miejsce w czasie każdą operację. Na końcu obie maszyny są sortowane wg czasu startu operacji.

Zaimplementowany algorytm składa się kolejno z mutacji, krzyżowania i selekcji.

**Mutacja** losuje dwie operacje na pierwszej maszynie. Jeśli może je zamienić, to robi to; jeśli nie, to losuje dalej. Następnie szuka dwóch operacji na drugiej maszynie, które mają ten sam numer zadania co operacje wylosowane uprzednio na pierwszej maszynie. Potem poprawia dla nich czasy startu tak, aby druga operacja danego zadania zaczynała się po zakończeniu pierwszej operacji danego zadania na pierwszej maszynie. Mutacja zależna jest od czasu trwania metaheurystyki, z każdą sekundą jest ich robionych mniej o 10%.

**Krzyżowanie** losuje dwa już istniejące rozwiązania, z każdego z nich bierze połowę operacji i wstawia do nowych rozwiązań, a następnie drugą połowę uzupełnia kolejno brakującymi operacjami z przeciwnego rozwiązania. Podczas jednej iteracji krzyżowanie generuje 100 nowych rozwiązań.

**Selekcja** składa się z dwóch części turnieju oraz ruletki.

Turniej polega na wybieraniu najlepszych rozwiązań, a ruletka na losowaniu rozwiązań, które mają pozostać.

Liczba rozwiązań wybieranych przez turniej początkowo jest równa 30% i jest zwiększana z każdą sekundą o 10%. W sumie po każdej iteracji pozostawianych jest 100 rozwiązań.

**Test ilości mutacji w pojedynczej iteracji.**

W naszym programie wraz z każdą sekundą wykonywania zmniejszamy liczbę mutacji o 10, aż do 0.

Podana na osi x liczba mutacji jest to maksymalna(początkowa) liczba.

Wartość funkcji celu to średnia wartość z 10 prób dla każdej z ilości mutacji.

Jak można zauważyć na wykresie w naszym programie optymalna liczba mutacji to 30, jednak jeśli nie wykonamy żadnych mutacji wartość funkcji celu będzie niewiele wyższa.

**Test czasu trwania algorytmu.**

Funkcja celu osiąga najlepszą wartość przy czasie trwania programu równym 10.

Można zauważyć, że im dłuższy czas działania tym lepsze są wyniki generowane przez algorytm, jednak największą poprawę obserwujemy w sekundach 1-4, potem najprawdopodobniej

**Test proporcji podczas selekcji.**

Jak to zostało opisane na początku sprawozdania selekcja wykorzystuje turniej oraz ruletkę w zmiennych proporcjach. Wartość procentowa rozwiązań wybieranych przez turniej jest co sekundę zwiększana o 10 aż do 100.

Liczby podane na osi x wykresu są to wartości początkowe.

Wartość funkcji celu to średnia z 10 prób dla każdej

z proporcji.

Można zauważyć , że proporcja początkowa 50 % jest optymalna. Jeśli zwiększymy ją lub zmniejszymy wartości funkcji celu wzrosną.