Maciej Szymczak
Mateusz Jeziorek

Tabu Search dla problemu przepływowego

1. Opis algorytmu

Tabu Search to algorytm metaheurystyczny stosowany do rozwiązania problemów optymalizacyjnych. Wykorzystywany do otrzymywania rozwiązań optymalnych lub niewiele różniących się od niego dla problemów z różnych dziedzin (np. planowanie zadań). Twórcą algorytmu jest Fred Glover.

Podstawową ideą algorytmu jest przeszukiwanie przestrzeni, stworzonej ze wszystkich możliwych rozwiązań, za pomocą sekwencji ruchu. W sekwencji ruchów występują ruchy niedozwolone tzw. ruchy tabu.

2. Opis algorytmu dla problemu przepływowego:

1) Wybierz początkowe ułożenie zadań i oblicz funkcję CMax.
2) Wybierz sąsiedztwo np. za pomocą SWAP lub INSERT (nie można wykonywać działań z tablicy tabu)
3) Oblicz funkcję CMax dla sąsiedztwa
4) Wybierz uszeregowanie zadań z sąsiedztwa dla najmniejszej wartości CMax
5) Jeżeli wybrana kolejność ma mniejszą wartość funkcji CMax niż początkowe ułożenie to je zastąp (jeżeli nie to koniec algorytmu)
6) Wpisz ruch (SWAP) do tablicy tabu
7) Jeżeli tablica tabu osiągnęła limit to usuń ostatni element
8) Jeżeli osiągnięto limit iteracji to zakończ i zwróć ostatnią kolejność, jeżeli nie to wróć do punktu drugiego

3. Próba rozwiązania dla wersji algorytmu TS1 (wstępnie bez tablicy zakazów):

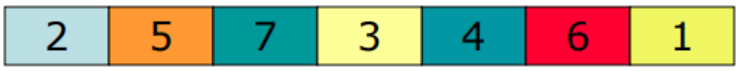
1) Obliczenie funkcji CMax dla początkowej kolejności
2) Ograniczenie liczby iteracji
3) Zamiana funkcją „swap” 5 losowych pozycji z początkowej kolejności (wykonanie ruchu)
4) Obliczenie funkcji Cmax dla każdej z sąsiedzkich kolejności i wybór najmniejszej wartości funkcji
5) Porównanie początkowej i wybranej wartości CMax
6) Jeżeli początkowa jest mniejsza to zakończ
7) Jeżeli nie to zastąp początkową kolejność przez wybraną
8) Zwiększenie nr iteracji o 1

9) Sprawdzenie czy nie została osiągnięta maksymalna liczba iteracji (jeśli tak to zwróć ostatnią kolejność)

4. Przykładowy algorytm

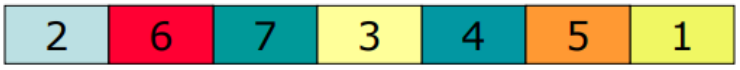
- Sytuacja wejściowa

Długość uszeregowania = 60



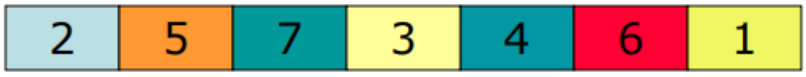
- Ruch typu wymiana (ang. swap, exchange)

i=5, j=6



Długość uszeregowania = 62

Iteracja 0



Długość uszeregowania = 60

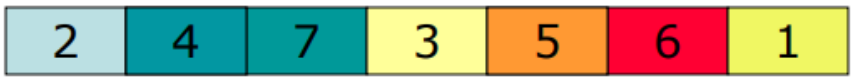
Lista tabu

	2	3	4	5	6	7
1						
2						
3						
4						
5						
6						

Top 5

Ruch	Z
5,4	-6
7,4	-4
3,6	-2
2,3	0
4,1	-1

Iteracja 1 - ruch (5,4)



Długość uszeregowania = 60-6=54

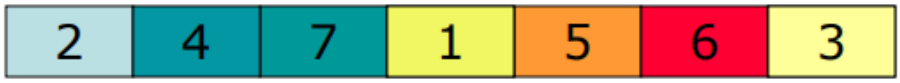
Lista tabu

	2	3	4	5	6	7
1						
2						
3						
4				3		
5						
6						

Top 5

Ruch	Z
3,1	-2
2,3	-1
3,6	1
7,1	2
6,1	4

Iteracja 2 - ruch (3,1)



Długość uszeregowania = 54-2=52

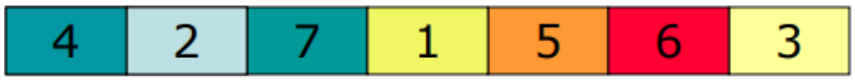
Lista tabu

	2	3	4	5	6	7
1		3				
2						
3						
4				2		
5						
6						

Top 5

Ruch	Z
1,3	2
2,4	4
7,6	6
4,5	7
5,3	9

Iteracja 3 - ruch (2,4)



Długość uszeregowania = 54+4=58

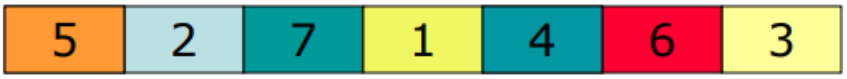
Lista tabu

	2	3	4	5	6	7
1		2				
2			3			
3						
4				1		
5						
6						

Top 5

Ruch	Z
4,5	-6
5,3	-2
7,1	0
1,3	3
2,6	6

Iteracja 4 - ruch (4,5)



Długość uszeregowania = 58-6=52

Lista tabu

	2	3	4	5	6	7
1		1				
2			2			
3						
4				3		
5						
6						

Top 5

Ruch	Z
7,1	0
4,3	3
6,3	5
5,4	6
2,6	8

5. Problemy i wnioski:
   - Ze względu na opóźnienia oddawania programów nie zdążyliśmy zrobić Tabu Search-a
   - Ten algorytm okazał się dość ciężki w zrozumienie i implementowaniu w przeciągu krótkiego czasu