

Algorytmy Metaheurystyczne - Symulowane wyżarzanie oraz Tabu Search

Piotr Maciejończyk

21 grudnia 2023

1 Symulowane wyżarzanie

1.1 Implementacja algorytmu

Algorytm symulowanego wyżarzania steruję poprzez podanie trzech zmiennych: temperatury początkowej T , współczynnika obniżania temperatury α oraz kryterium stopu - maksymalną liczbę epok z rzędu zwracającą brak lepszego rozwiązania M .

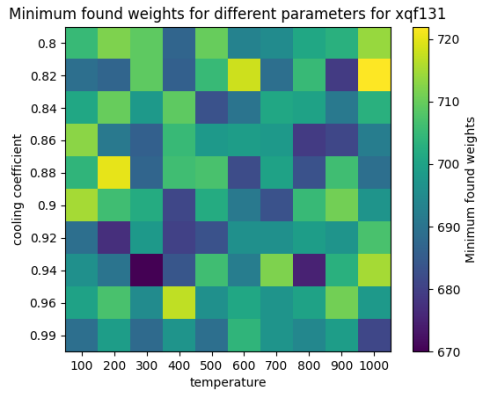
W trakcie każdej epoki generuję i ewaluuję tyle sąsiadów, ile wynosi pięciokrotność liczby wierzchołków w grafie. W odróżnieniu od algorytmu *LocalSearch*, algorytm symulowanego wyżarzania przyjmuje "gorsze" rozwiązanie z prawdopodobieństwem $e^{\frac{f(X)-f(X')}{T}}$, gdzie $f(X)$ to koszt cyklu rozwiązania X .

1.2 Dobieranie parametrów

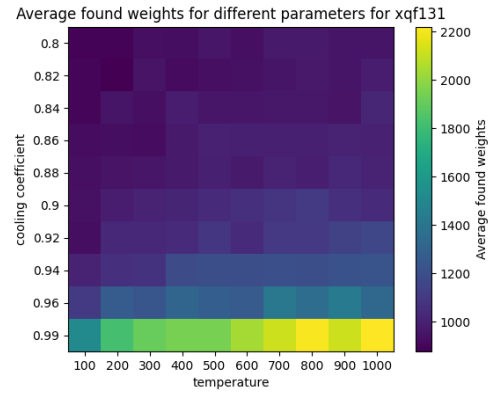
Najpierw, policzyłem średni koszt 50 najlepszych rozwiązań oraz średni koszt 50 średnich kosztów znalezionych rozwiązań dla podanych parametrów:

1. $T \in \{100, 200, \dots, 1000\}$
2. $\alpha \in \{0.8, 0.82, 0.84, \dots, 0.96, 0.99\}$
3. $M = 100$

Parametry te testowałem na plikach testowych, dla których liczba wierzchołków była niższa od 1000 (jako liczbę iteracji w jednej epoce przyjmowałem liczbę wierzchołków grafu):

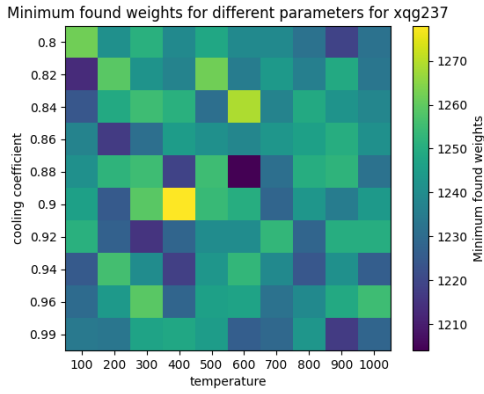


(a) Najmniejsze wartości cykli dla xqf131

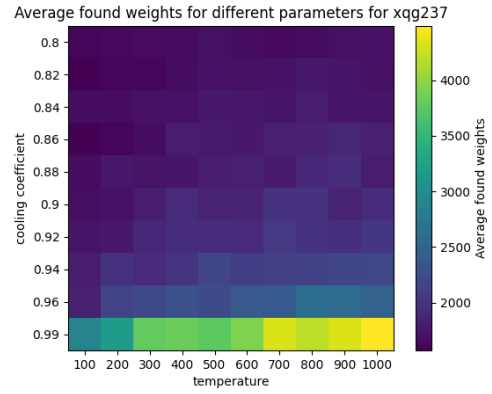


(b) Średnie wartości cykli dla xqf131

Rysunek 1: Wyniki dla xqf131

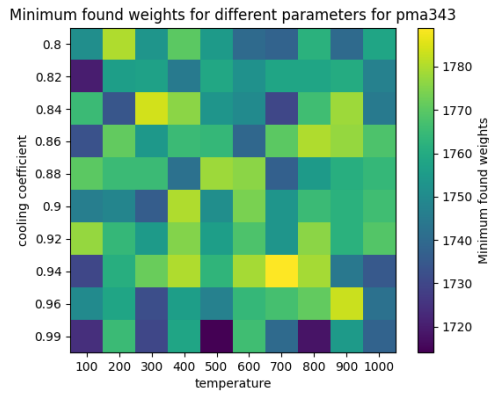


(a) Najmniejsze wartości cykli dla xqg237

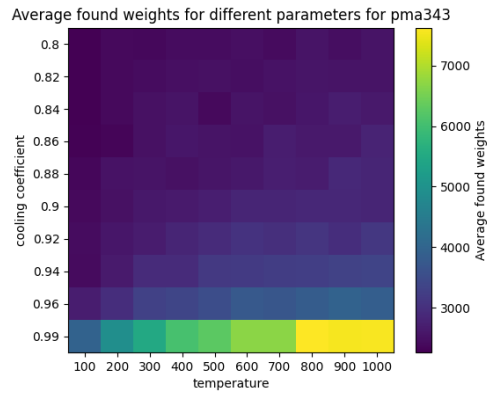


(b) Średnie wartości cykli dla xqg237

Rysunek 2: Wyniki dla xqg237

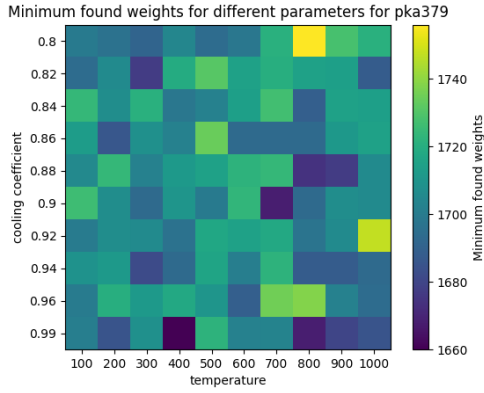


(a) Najmniejsze wartości cykli dla pma343

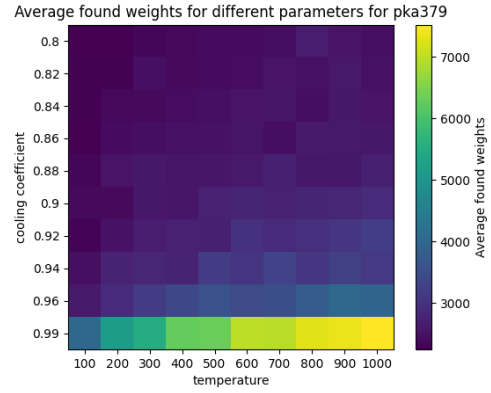


(b) Średnie wartości cykli dla pma343

Rysunek 3: Wyniki dla pma343

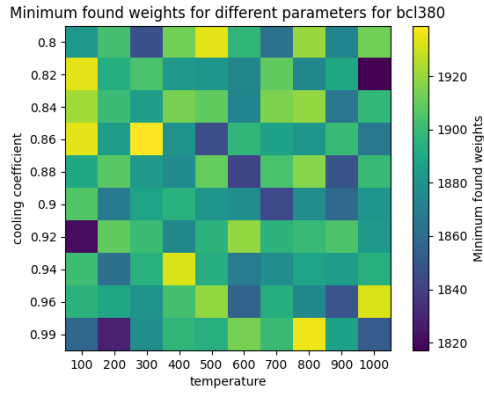


(a) Najmniejsze wartości cykli dla pka379

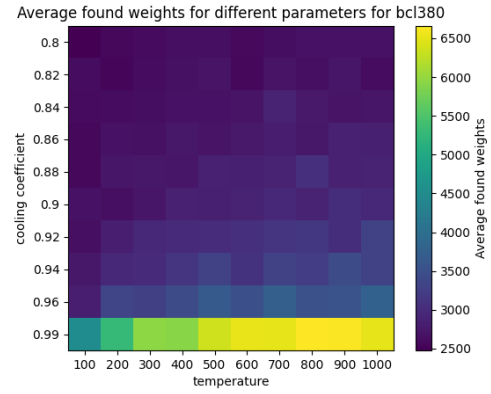


(b) Średnie wartości cykli dla pka379

Rysunek 4: Wyniki dla pka379

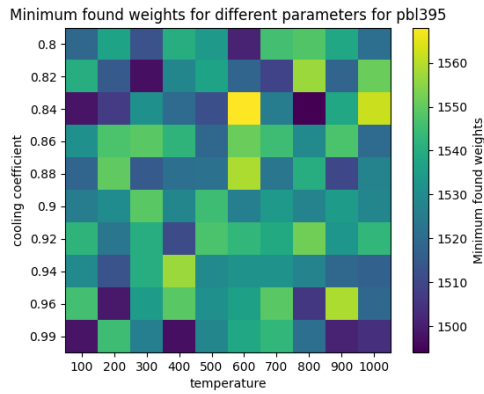


(a) Najmniejsze wartości cykli dla bcl380

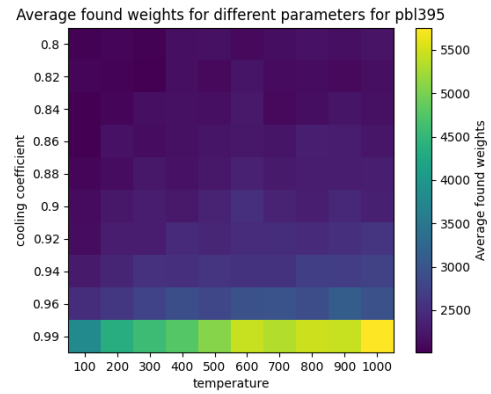


(b) Średnie wartości cykli dla bcl380

Rysunek 5: Wyniki dla bcl380

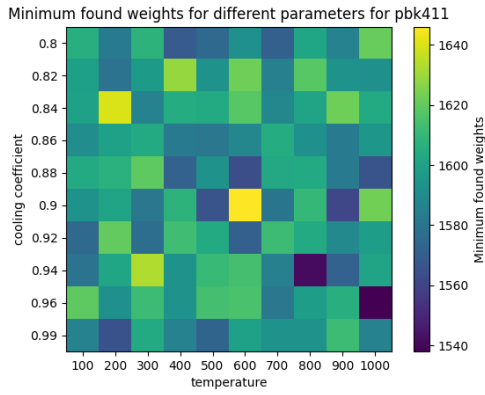


(a) Najmniejsze wartości cykli dla pbl395

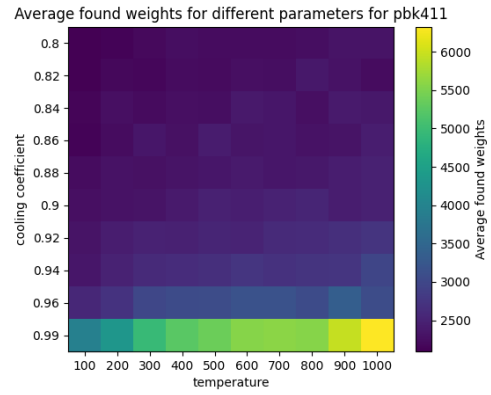


(b) Średnie wartości cykli dla pbl395

Rysunek 6: Wyniki dla pbl395

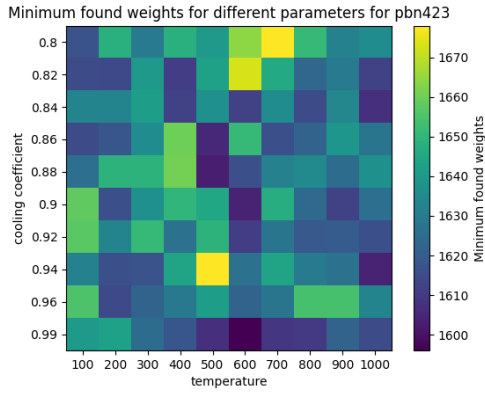


(a) Najmniejsze wartości cykli dla pbk411

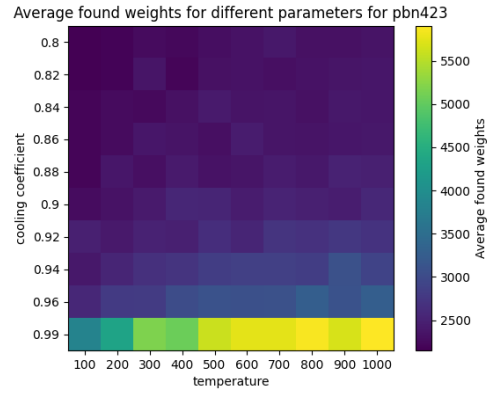


(b) Średnie wartości cykli dla pbk411

Rysunek 7: Wyniki dla pbk411

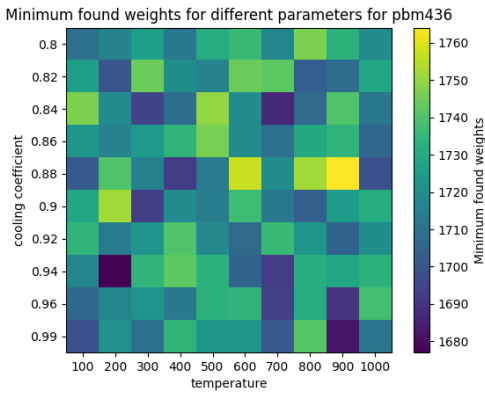


(a) Najmniejsze wartości cykli dla pbn423

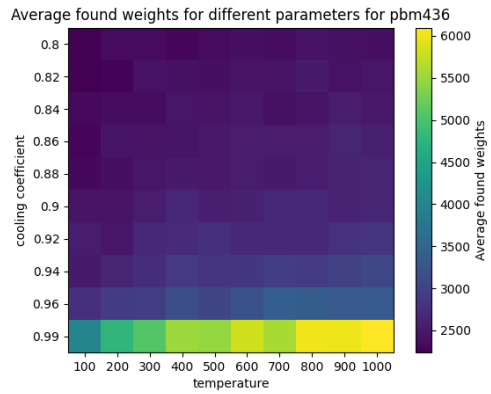


(b) Średnie wartości cykli dla pbn423

Rysunek 8: Wyniki dla pbn423

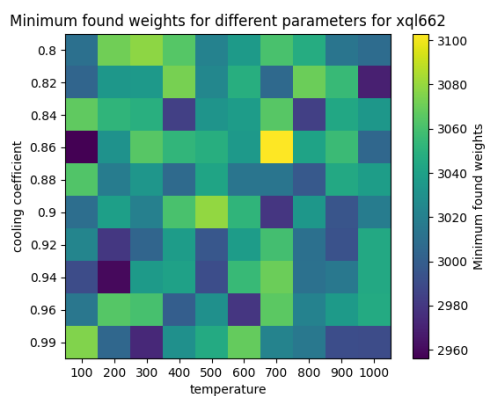


(a) Najmniejsze wartości cykli dla pbm436

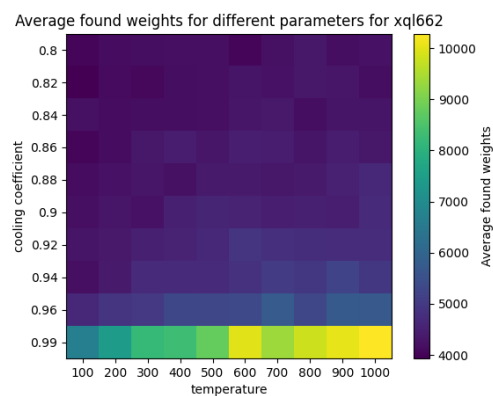


(b) Średnie wartości cykli dla pbm436

Rysunek 9: Wyniki dla pbm436



(a) Najmniejsze wartości cykli dla xql662



(b) Średnie wartości cykli dla xql662

Rysunek 10: Wyniki dla xql662

Patrząc na wyniki, dla najmniejszego współczynnika obniżania temperatury i najmniejszej temperatury otrzymujemy najlepsze średnie wyniki eksperymentów. Natomiast, jeśli chodzi o najlepsze otrzymane rozwiązania, to wyniki są bardzo do siebie podobne, zatem ostatecznie postanowiłem wybrać takie parametry:

$$T = 100; \alpha = 0.9; M = 100$$

Na końcu eksperymentowałem dla tych parametrów dobór liczby prób w ciągu jednej epoki. Zauważyłem, że zwiększając tę liczbę prób (np. mnożąc liczbę wierzchołków przez 2, 3 lub 5) otrzymywałem trochę lepsze wyniki, lecz nie różniły się one od siebie znacznie. **Postanowiłem przyjąć za liczbę prób w epoce wartość odpowiadającą pięciokrotności wierzchołków w grafie.**

1.3 Wyniki dla większych przykładów

Z powyżej wymienionymi parametrami uruchomiłem algorytm symulowanego wyżarzania na grafach z liczbą wierzchołków większą niż tysiąc. Dodatkowo, dla każdego przykładu, załączyłem powstałe najlepsze rozwiązania (o najmniejszej wadze) Otrzymane rozwiązania porównałem z wynikami dla algorytmu Local Search dla losowych cykli z pełnym sąsiedztwem z poprzedniej listy.

Filename	Simulated Annealing		Local Search	
	The best solution weight	Average solution weight	The best solution weight	Average solution weight
xit1083	4157	4257.21	4173	4271.19
icw1483	4918	5056.45	5010	5122.12
djc1785	6954	7100.03	6997	7129.9
dcb2086	7508	7667.51	7543	7722.32
pds2566	8693	8913.45	8862	9011.12

Tabela 1: Wyniki dla większych przykładów

Wyniki otrzymane za pomocą symulowanego wyżarzania są nieco lepsze niż te otrzymane za pomocą algorytmu *LocalSearch* dla losowych cykli z przeszukiwaniem pełnego sąsiedztwa.

2 Tabu Search

2.1 Implementacja algorytmu

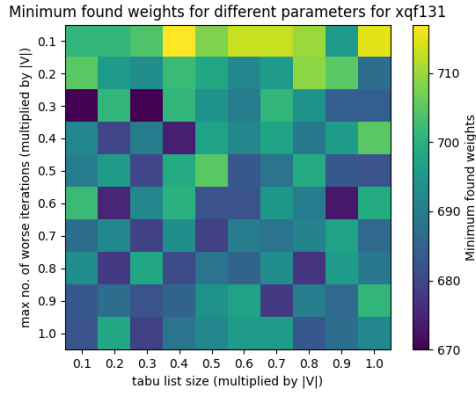
Dla każdego grafu testowego dobieierałem losowo znaleziony cykl, po czym dla odpowiednio dobranych parametrów rozmiaru listy tabu oraz maksymalnej dopuszczalnej liczby iteracji bez poprawy, przeszukiwałem pełne sąsiedztwo aktualnego rozwiązania.

2.2 Dobór parametrów

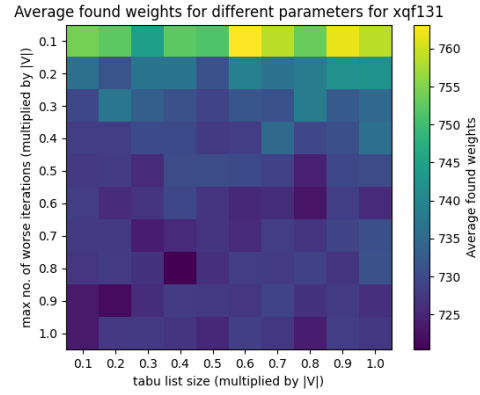
Postanowiłem uzależnić parametry od liczby wierzchołków w aktualnie obliczanym grafie. Zatem podane poniżej testowane parametry są odpowiednio mnożone przez liczbę $|V|$ (dla każdego parametru wykonałem 50 iteracji):

1. $S \in \{0.1, 0.2, \dots, 1.0\}$ (rozmiar listy tabu)
2. $M = \{0.1, 0.2, \dots, 1.0\}$ (maksymalna liczba dopuszczalnych iteracji bez poprawy rezultatu)

Poniżej przedstawiłem wyniki dla grafów, w których $|V| < 1000$:

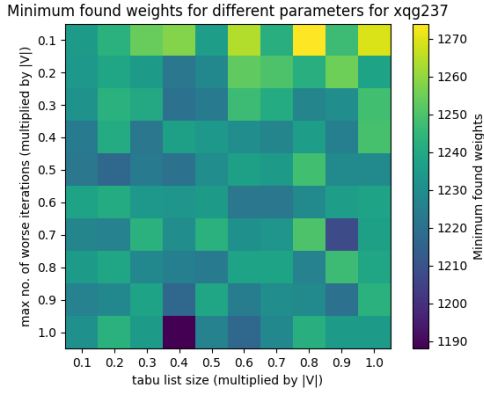


(a) Najmniejsze wartości cykli dla xqf131

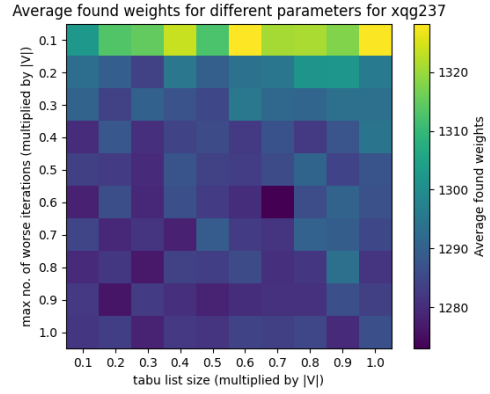


(b) Średnie wartości cykli dla xqf131

Rysunek 11: Wyniki dla xqf131

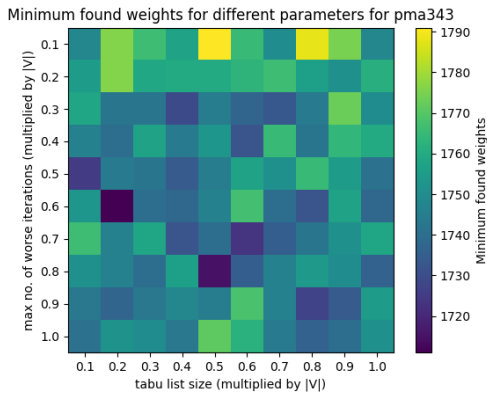


(a) Najmniejsze wartości cykli dla xqg237

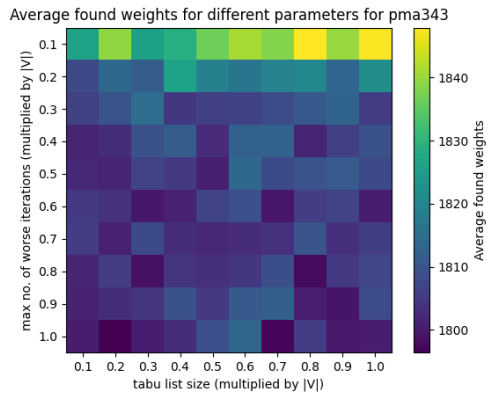


(b) Średnie wartości cykli dla xqg237

Rysunek 12: Wyniki dla xqg237

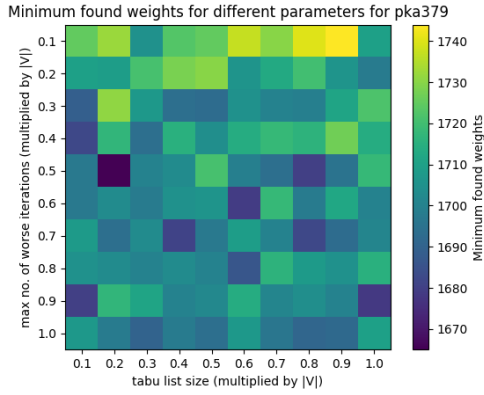


(a) Najmniejsze wartości cykli dla pma343

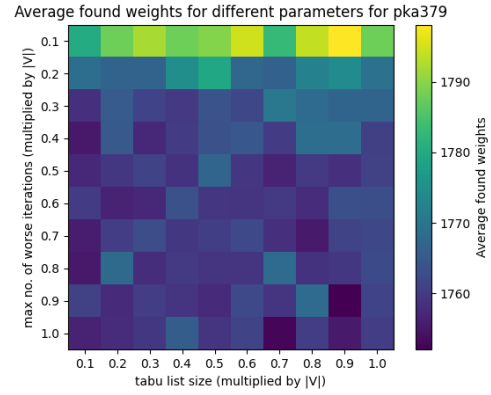


(b) Średnie wartości cykli dla pma343

Rysunek 13: Wyniki dla pma343

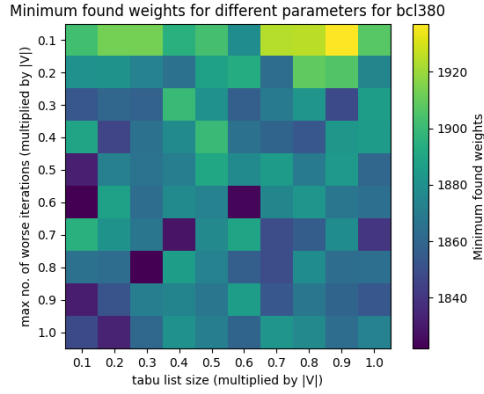


(a) Najmniejsze wartości cykli dla pka379

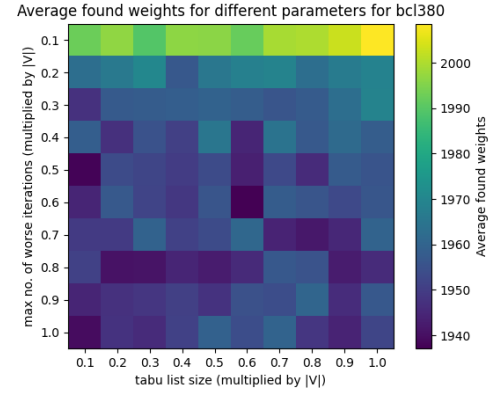


(b) Średnie wartości cykli dla pka379

Rysunek 14: Wyniki dla pka379

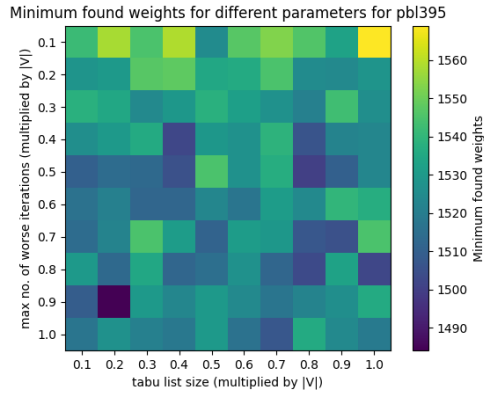


(a) Najmniejsze wartości cykli dla bcl380

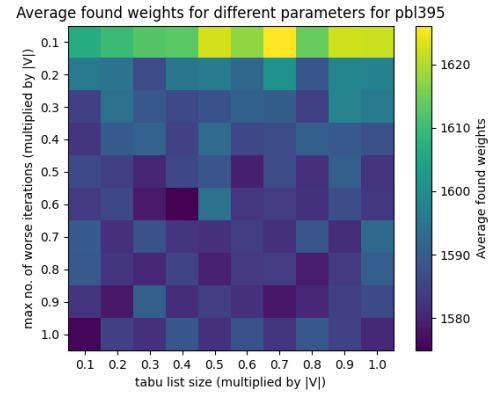


(b) Średnie wartości cykli dla bcl380

Rysunek 15: Wyniki dla bcl380

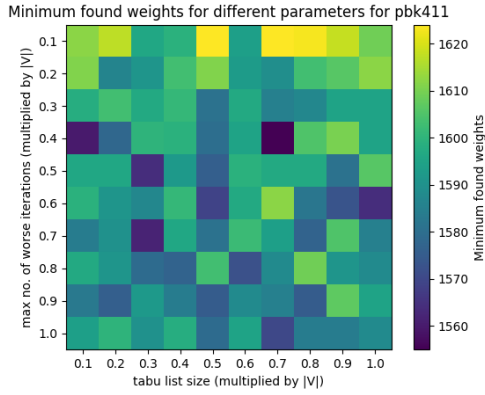


(a) Najmniejsze wartości cykli dla pbl395

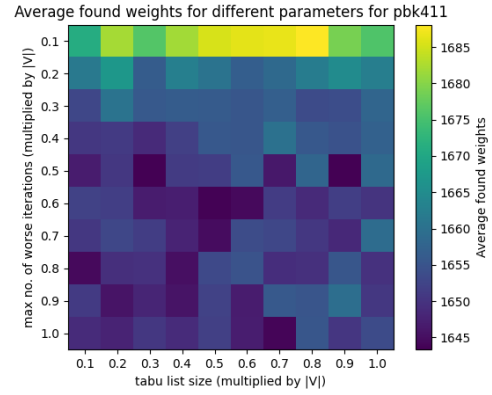


(b) Średnie wartości cykli dla pbl395

Rysunek 16: Wyniki dla pbl395

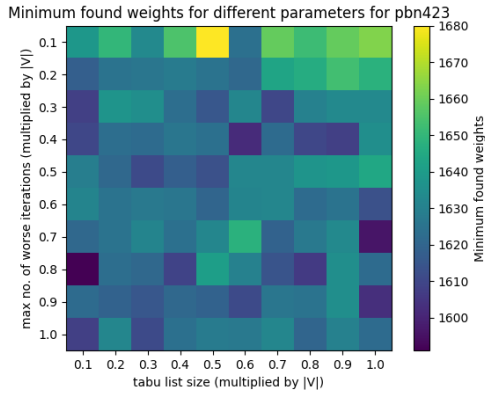


(a) Najmniejsze wartości cykli dla pbk411

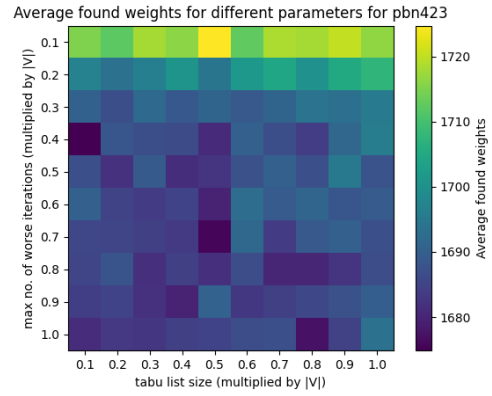


(b) Średnie wartości cykli dla pbk411

Rysunek 17: Wyniki dla pbk411

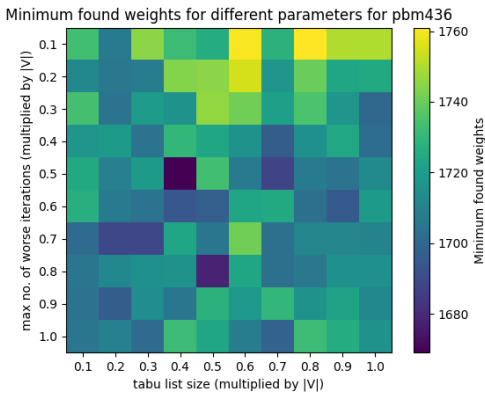


(a) Najmniejsze wartości cykli dla pbn423

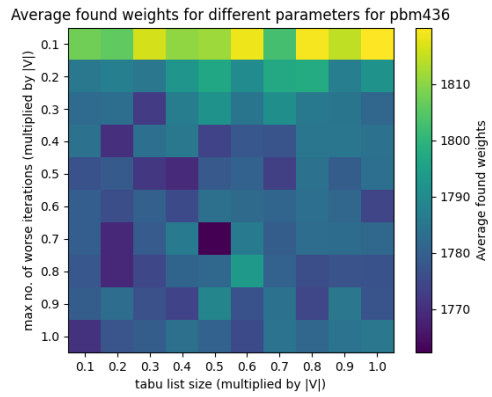


(b) Średnie wartości cykli dla pbn423

Rysunek 18: Wyniki dla pbn423

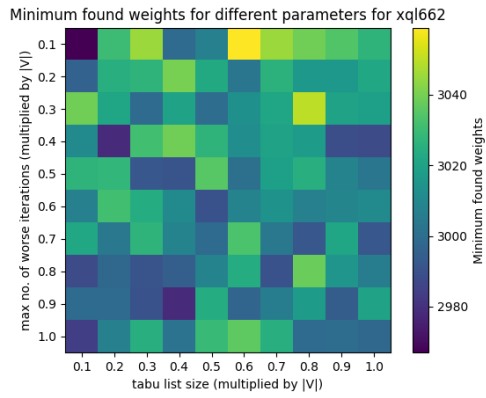


(a) Najmniejsze wartości cykli dla pbm436

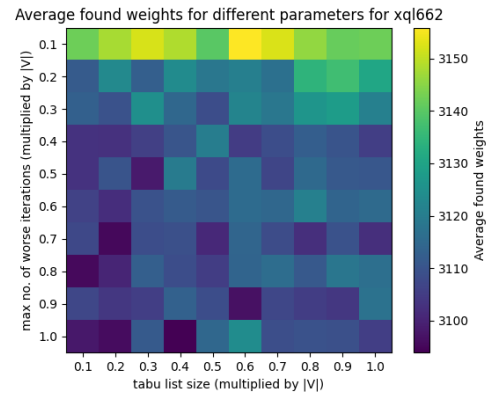


(b) Średnie wartości cykli dla pbm436

Rysunek 19: Wyniki dla pbm436



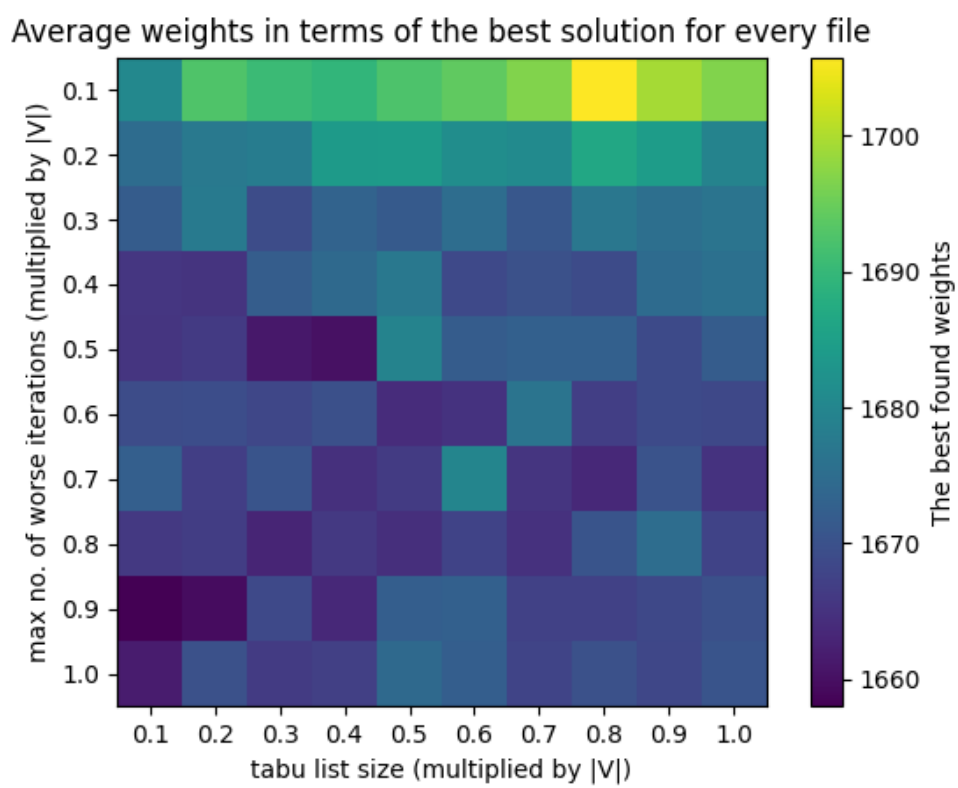
(a) Najmniejsze wartości cykli dla xql662



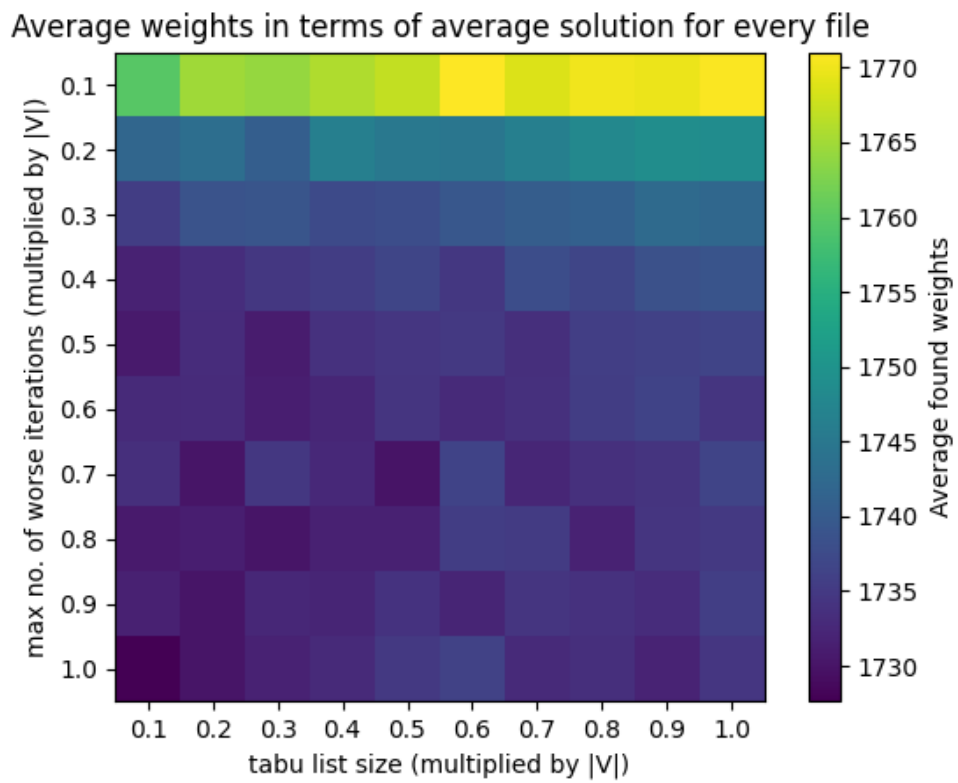
(b) Średnie wartości cykli dla xql662

Rysunek 20: Wyniki dla xql662

Na podstawie tych 10 rezultatów obliczyłem z nich średnią i uzyskałem ostateczne wyniki w postaci poniższych diagramów:



Rysunek 21: Średnia najlepszych uzyskanych rozwiązań



Rysunek 22: Średnia z wszystkich średnio uzyskanych rozwiązań

Wyniki były bardzo mocno porównywalne, niezależnie od doboru parametrów. Da się jednak zauważyć, że dla $M = 0.9$ oraz $S = 0.1$ uzyskiwano średnio jedno z najlepszych rozwiązań, zatem postanowiłem przyjąć te parametry jako:

$$S = 0.1 \cdot |V|$$

$$M = 0.9 \cdot |V|$$

2.3 Wyniki dla większych przykładów

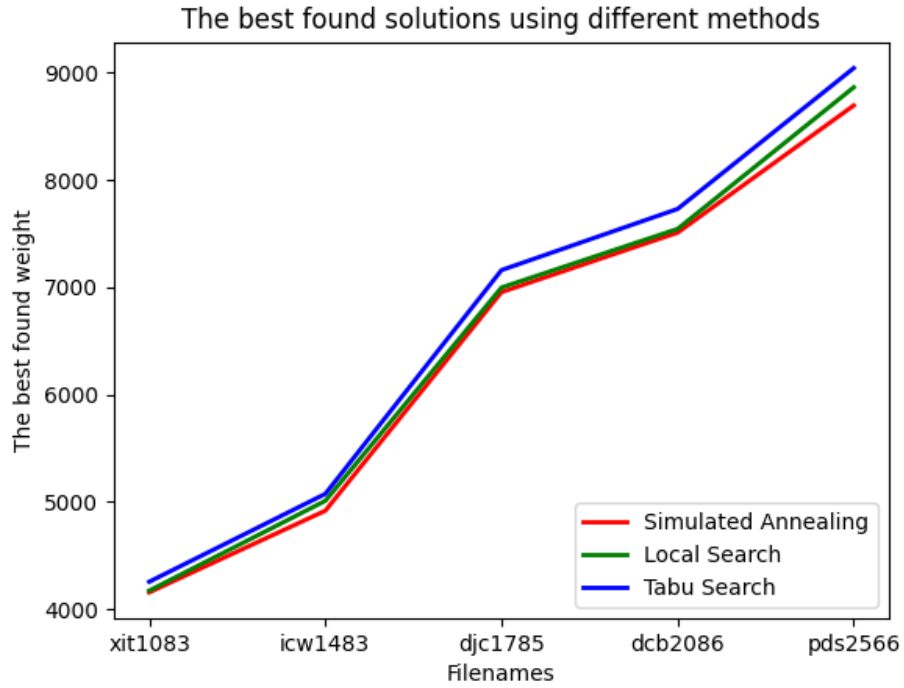
Z powyżej wymienionymi parametrami uruchomiłem algorytm *TabuSearch* na grafach z liczbą wierzchołków większą niż tysiąc. Otrzymane rozwiązania porównałem z wynikami dla algorytmu *LocalSearch* dla losowych cykli z pełnym sąsiedztwem z poprzedniej listy:

Filename	Tabu Search		Local Search	
	The best solution weight	Average solution weight	The best solution weight	Average solution weight
xit1083	4256	4375.2	4173	4271.19
icw1483	5075	5218.19	5010	5122.12
djc1785	7159	7312.91	6997	7129.9
dcb2086	7729	7905.26	7543	7722.32
pds2566	9041	9222.67	8862	9011.12

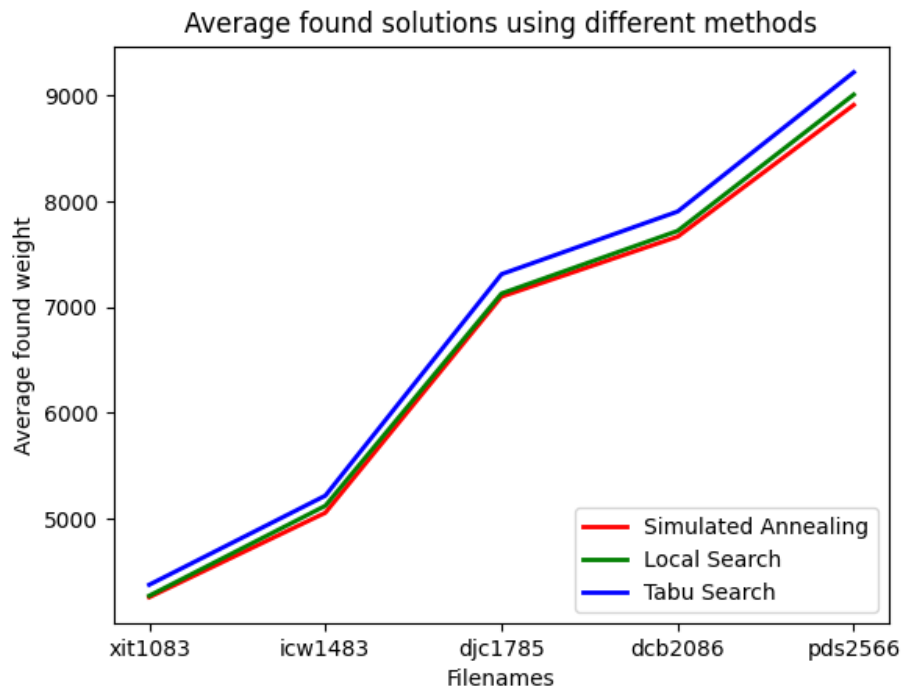
Tabela 2: Wyniki dla większych przykładów

3 Podsumowanie

Najlepsze wyniki uzyskałem stosując algorytm symulowanego wyżarzania na ograniczonym sąsiedztwie przeszukiwanego rozwiązania, w porównaniu do rezultatów algorytmu *LocalSearch* działającym na pełnym sąsiedztwie. Najgorsze wyniki, lecz bardzo podobne do wspomnianego algorytmu *LocalSearch*, uzyskałem dla algorytmu *TabuSearch* na ograniczonym sąsiedztwie.



Rysunek 23: Porównanie najlepszych znalezionych rozwiązań za pomocą różnych metod



Rysunek 24: Porównanie średnich wag znalezionych rozwiązań za pomocą różnych metod