

## Simulated annealing:

Dla danego osobnika generujemy sąsiadów dla kolejności poprzez zamianę dwóch sąsiednich par (cyklicznie, dlatego możemy zamienić także pierwszą z ostatnią) i dla zasobów poprzez zamianę jednego zasobu przydzielonego dla jednego zadania na sąsiedni w liście dozwolonych zasobów dla tego zadania. Jeżeli funkcja oceny danego sąsiada jest mniejsza niż osobnika, to przyjmujemy go na miejsce osobnika. Jeżeli nie, to każdy sąsiad ma drugą szansę w postaci prawdopodobieństwa określonego wzorem:

$$e^{\frac{f(\text{osobnika}) - f(\text{sąsiada})}{\text{temperatura}}}$$

Parametrami dla tej metody jest temperatura początkowa i wartość o jaką temperatura się zmniejsza co każdą iterację. Wyniki dla 200 wywołań i 1000 iteracji dla instancji 100\_5\_20\_9\_D3.def:

Temp. maks.	Różnica temp.	Min. ocena	Średnia ocena	Maks. ocena	Odchylenie stand.
1	0.001	390.0	412.43	451.0	12.382176013459139
5	0.005	391.0	411.84	460.0	12.692732351446420
25	0.025	391.0	417.77	448.0	12.224214254561073
125	0.125	397.0	430.74	460.0	13.993230686675078
625	0.625	407.0	459.31	517.0	22.939840074121317

Dla zbyt dużych temperatur wykres jest zbyt chaotyczny. Porównanie dla innych metod ponownie przy 200 wywołaniach:

Nazwa	Min. ocena	Średnia ocena	Maks. ocena	Odchylenie stand.
Przeszukiwanie lokalne	398.0	466.35	536.0	30.528468250755285
Przeszukiwanie lokalne ++	390.0	415.40	451.0	13.924036335732103
Algorytm genetyczny	387.0	388.53	393.0	1.0773611607567226

Jak widać w moich badaniach symulowane wyżarzanie przy odpowiednio dobranych parametrach działa skuteczniej od przeszukiwania lokalnego, ale słabiej od algorytmu genetycznego. „Przeszukiwanie lokalne ++” to zmodyfikowane przeszukiwanie lokalne, które w przypadku gdy nie ma lepszego rozwiązania, ale są równie dobre wybiera inne równie dobre, co pozwala uzyskać porównywalnie dobre rozwiązania jak przy użyciu SA czy TS. Przykładowy wykres dla temperatury maksymalnej 25:

**Harmonogramowanie**

