

# Algorytmy Ewolucyjne

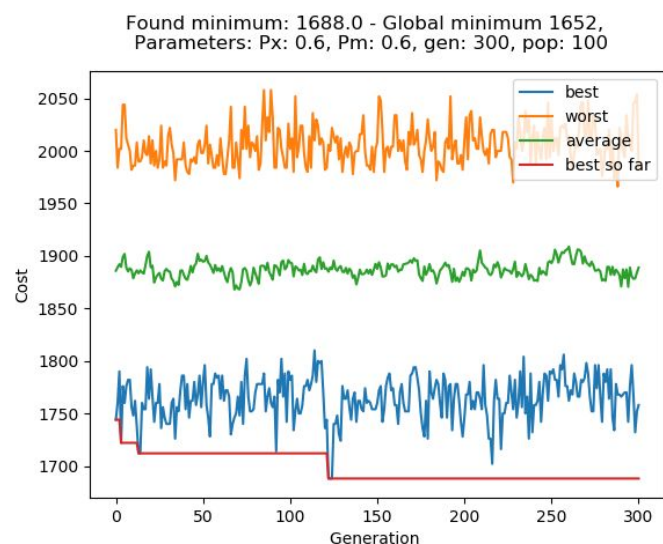
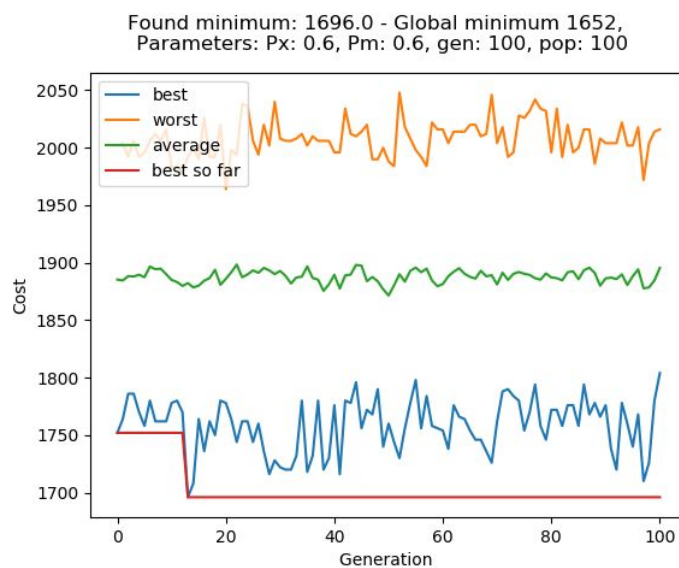
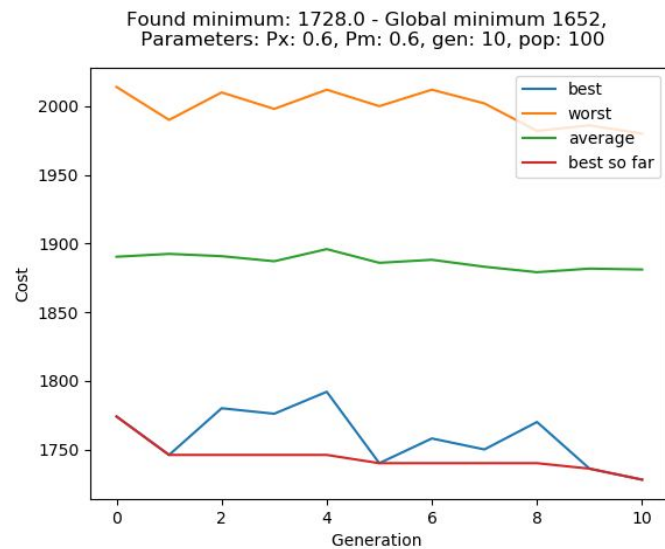
Julian Jurec 228094, 15.03.2018

## Plan badań

1. Badanie wpływu liczby pokoleń na wydajność
2. Badanie wpływu wielkości populacji na wydajność
3. Badanie wpływu prawdopodobieństwa mutacji na wydajność
4. Badanie wpływu prawdopodobieństwa krzyżowania na wydajność
5. Porównanie metod selekcji
6. Badanie opcji zachowania najlepszego osobnika
7. Porównanie czasów wykonywania i wydajności w zależności od wielkości problemu
8. Porównanie z algorytmem nieewolucyjnym.

# Badanie wpływu liczby pokoleń na wydajność

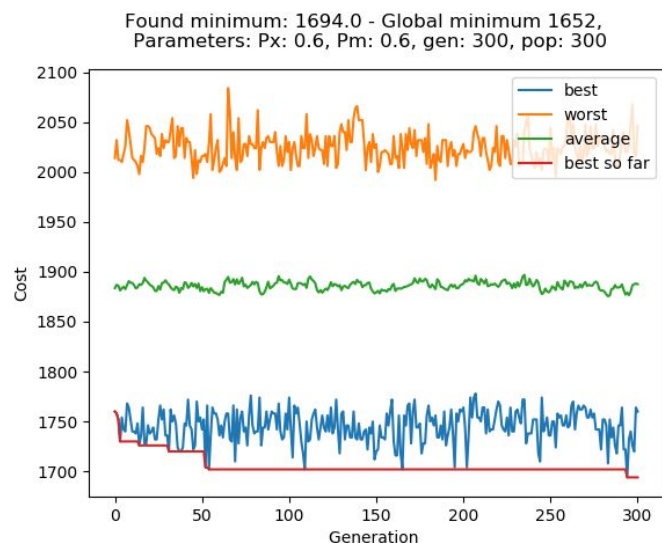
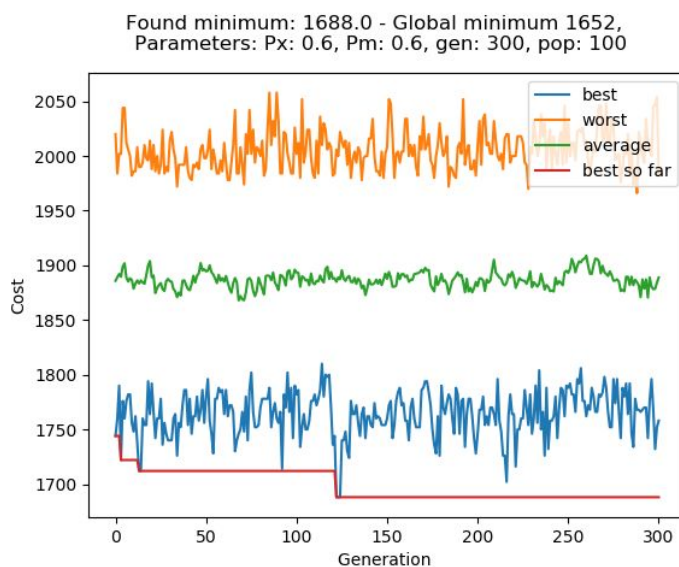
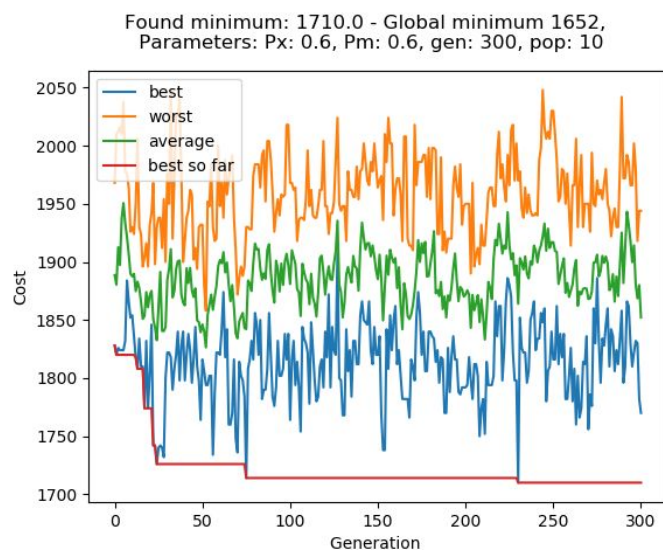
Jako metody selekcji użyto ruletki bez zachowania najlepszego



Wyniki algorytmu poprawiają się wraz ze zwiększaniem liczby pokoleń: średnie najlepsze dopasowania: (10->1734.8, 100->1704.4, 300->1688.4) mimo dużej fluktuacji, bo jest więcej czasu, żeby stochastycznie trafić na dobre rozwiązanie.

# Badanie wpływu wielkości populacji na wydajność

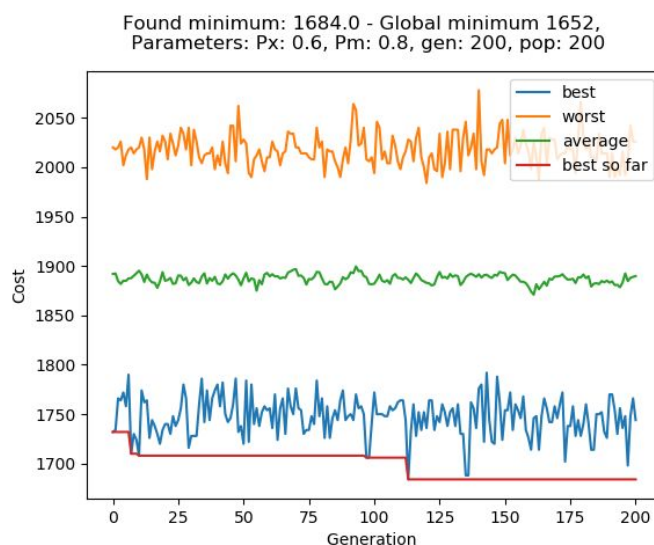
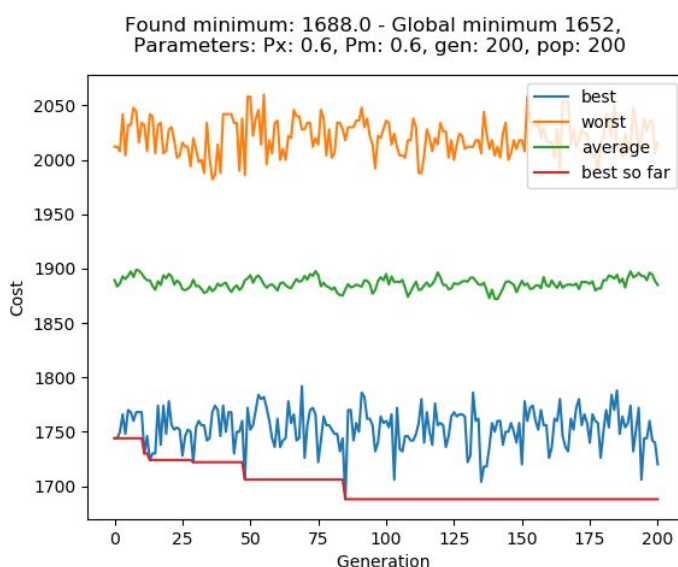
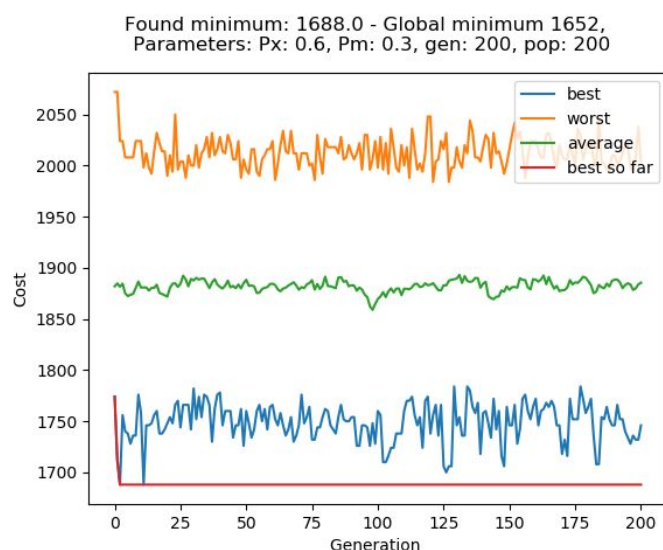
*Jako metody selekcji użyto ruletki bez zachowania najlepszego*



Wyniki algorytmu poprawiają się wraz ze zwiększaniem wielkości populacji: średnie najlepsze dopasowania: (10->1717.2, 100->1704.4, 300->1688.4), ponieważ w każdym pokoleniu jest więcej osobników, z których każdy ma szansę na poprawienie wartości najlepszego dopasowania. Wyjątkowo mały rozmiar populacji skutkuje bardzo dużą fluktuacją dopasowania.

# Badanie wpływu prawdopodobieństwa mutacji na wydajność

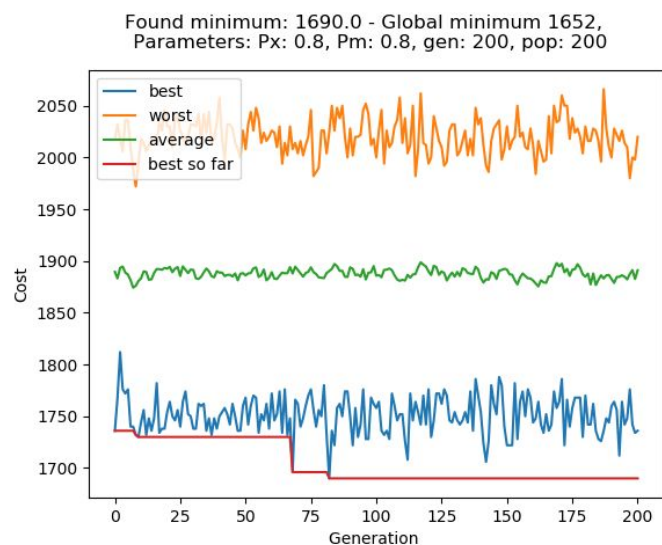
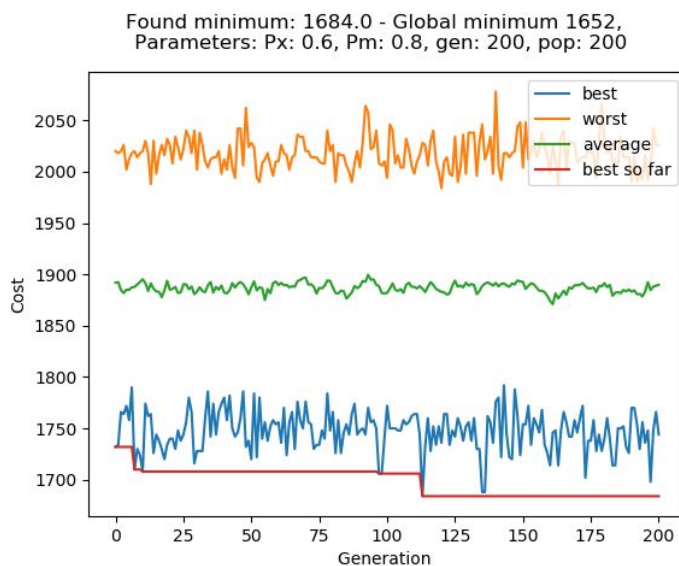
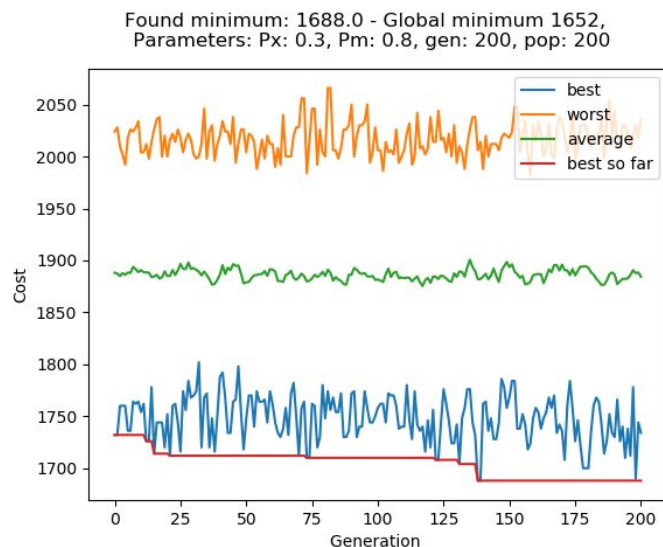
Jako metody selekcji użyto ruletki bez zachowania najlepszego.  $GEN=POP=200$  dla szybkości obliczeń.



Średnie najlepsze dopasowania: (0.3->1696.4, 0.6->1690.4, 0.8->1689.2) osiągają najlepszą wartość dla wartości parametru 0.8, co oznacza że zwiększanie różnorodności genetycznej (przynajmniej do pewnego pułapu) skutkuje zwiększeniem najlepszego dopasowania. Do następnych obliczeń zachowano taką wartość tego parametru.

# Badanie wpływu prawdopodobieństwa krzyżowania na wydajność

Jako metody selekcji użyto ruletki bez zachowania najlepszego.  $GEN=POP=200$  dla szybkości obliczeń.

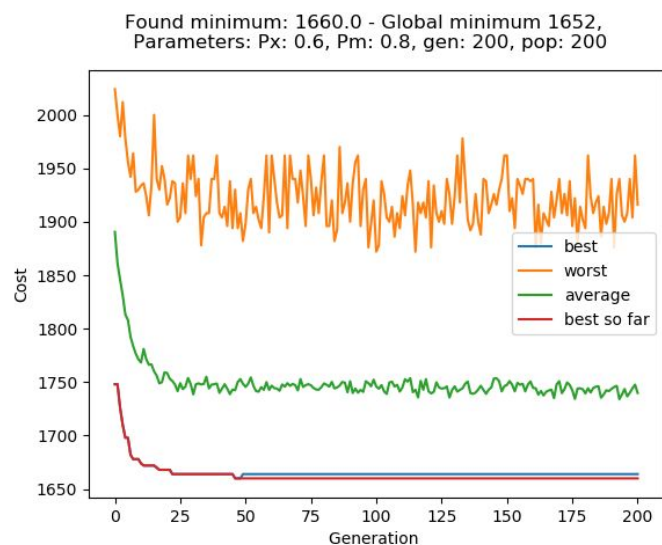
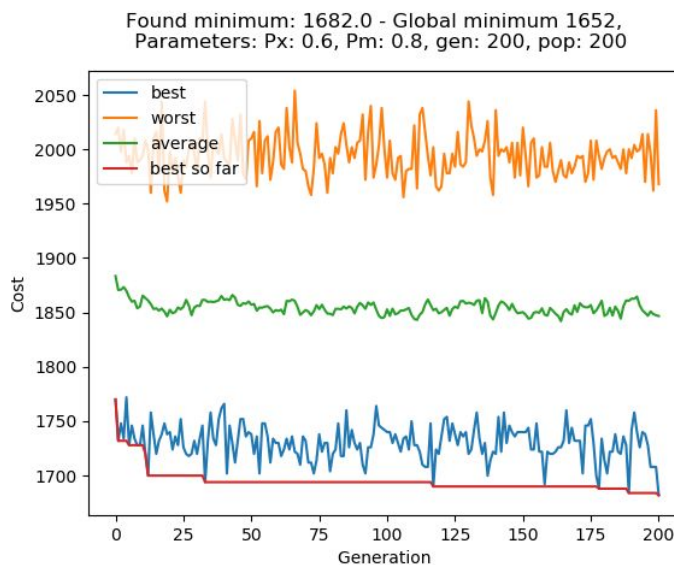
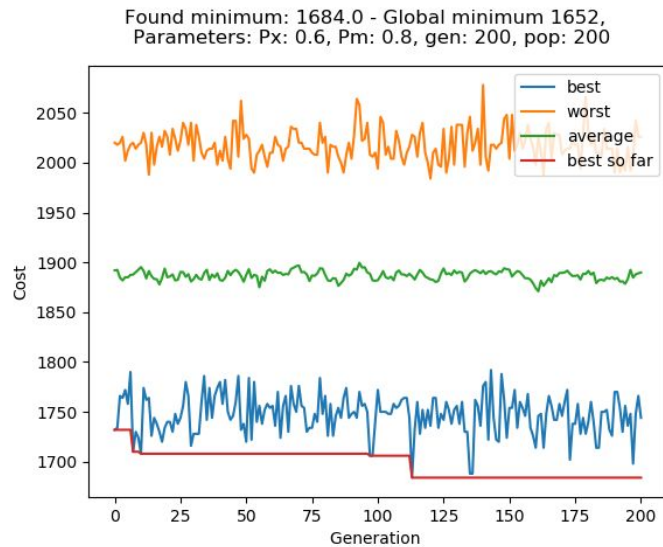


Średnie najlepsze dopasowania: (0.3->1693.6 0.6->1690.4, 0.8->1692.8) osiągają najlepszą wartość dla wartości parametru 0.6, co oznacza że zwiększanie wpływu cech dziedziczonych powyżej pewnego pułapu zaczyna obniżać wartość najlepszego przystosowania. Do następnych obliczeń zachowano najlepszą wartość tego parametru, czyli 0.6.



# Porównanie metod selekcji

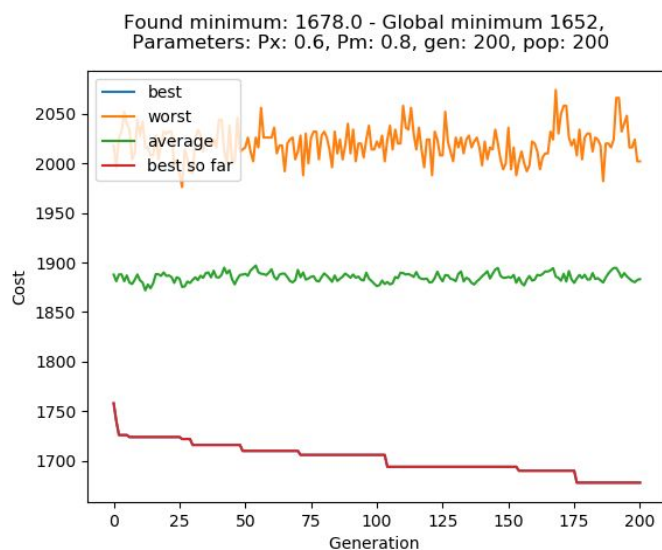
*GEN=POP=200 dla szybkości obliczeń.  $P_x=0.6$ ,  $P_m = 0.8$  - najlepsze wartości dotychczas*



Na wykresach kolejno przedstawiono ruletkę, turniej o wielkości 2 i turniej o wielkości 10. Średnie wartości minimalne kolejno wynoszą (1689.2, 1682.0, 1653.6). Widać że najlepszym przystosowaniem charakteryzuje się większy turniej, co nie jest zaskakujące, ponieważ faworyzuje on przekazywanie genów odnoszących sukces.

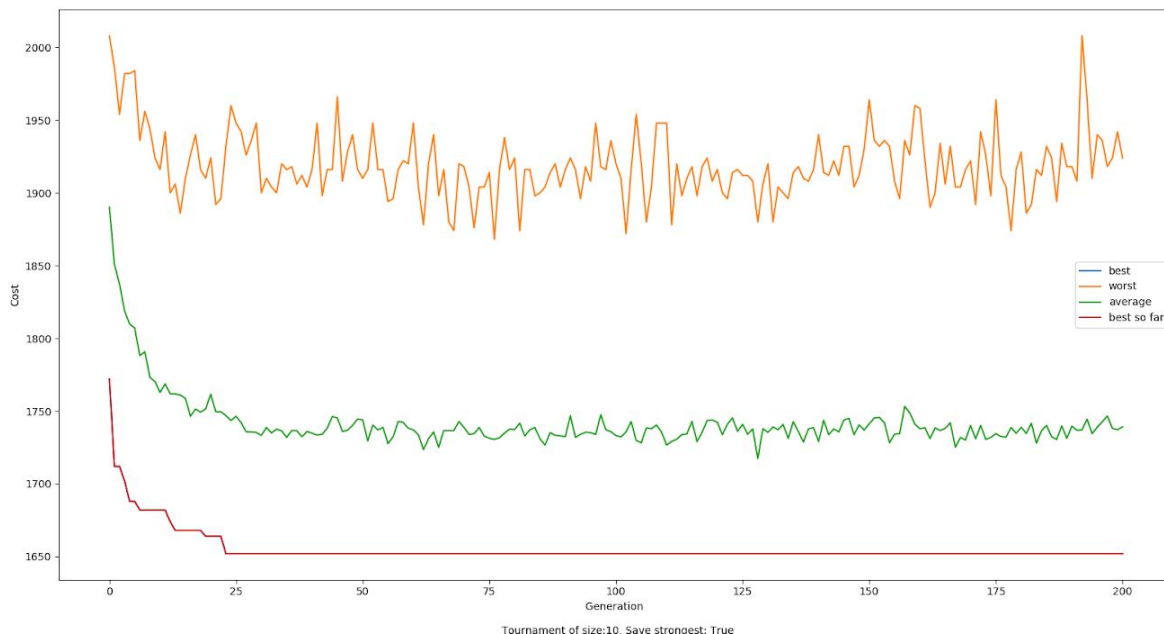
# Badanie opcji zachowania najlepszego osobnika

GEN=POP=200 dla szybkości obliczeń.  $P_x=0.6$ ,  $P_m = 0.8$  - najlepsze wartości dotychczas



## Ruletka Save Strongest (1678.8)

Found minimum: 1652.0 - Global minimum 1652,  
Parameters:  $P_x$ : 0.6,  $P_m$ : 0.8, gen: 200, pop: 200



## Turniej wielkości 10 Save Strongest (1653.4)

Dodanie opcji zachowania najlepszego osobnika zwiększa wydajność algorytmu, ponieważ niweluje fluktuacje najlepszego osobnika w danym pokoleniu co zwiększa jakość puli genetycznej. W ruletce różnica jest mniejsza, ponieważ nadal mniej faworyzowane są lepsze geny. W turnieju opcja ta znacząco przyspiesza osiągnięcie lokalnego minimum kosztu.

# Porównanie czasów wykonywania i wydajności w zależności od wielkości problemu

Średni czas wykonywania programu dla 12 miast i POP=GEN=300: 55.315s

Średni czas wykonywania programu dla 12 miast i POP=GEN=200: 28.420s

Średni czas wykonywania programu dla 20 miast i POP=GEN=200: 132.946s

Jak widać zwiększanie liczby miast (długość genotypu) dużo bardziej zwiększa czas wykonywania się algorytmu niż porównywalne zmiany liczby pokoleń i wielkości populacji. Wpływ na zastosowanie praktyczne jest zwiększony o tyle, że każde badanie należy wykonać wielokrotnie w celu zwiększenia trafności wniosków.

## Porównanie z algorytmem nieewolucyjnym

Dla 12 miast POP=GEN=200,  $P_m = 0.8$ ,  $P_x=0.6$ , Turniej wielkości 10, SS:

- Średni czas wykonywania się algorytmu: 55.315s
- Średnie najlepsze przystosowanie: 1653.4

Dla 12 miast losowe przeszukiwanie przestrzeni rozwiązań:

- Czas wykonywania się algorytmu: 60s
- Średnie najlepsze przystosowanie: 1694.0

Dla 20 miast POP=GEN=200,  $P_m = 0.8$ ,  $P_x=0.6$ , Turniej wielkości 10, SS:

- Średni czas wykonywania się algorytmu: 122.946s
- Średnie najlepsze przystosowanie: 6943.2

Dla 20 miast losowe przeszukiwanie przestrzeni rozwiązań:

- Czas wykonywania się algorytmu: 130s
- Średnie najlepsze przystosowanie: 7244.0

Algorytmy genetyczne są dużo wydajniejsze od losowego przeszukania w określonym czasie przy założeniu wybrania odpowiednich parametrów dla algorytmu.