

Wydział Elektroniki i Technik Informacyjnych
Politechnika Warszawska

Wstęp do sztucznej inteligencji

Raport z laboratorium 2.

Kacper Bugała

Warszawa, 2022Z

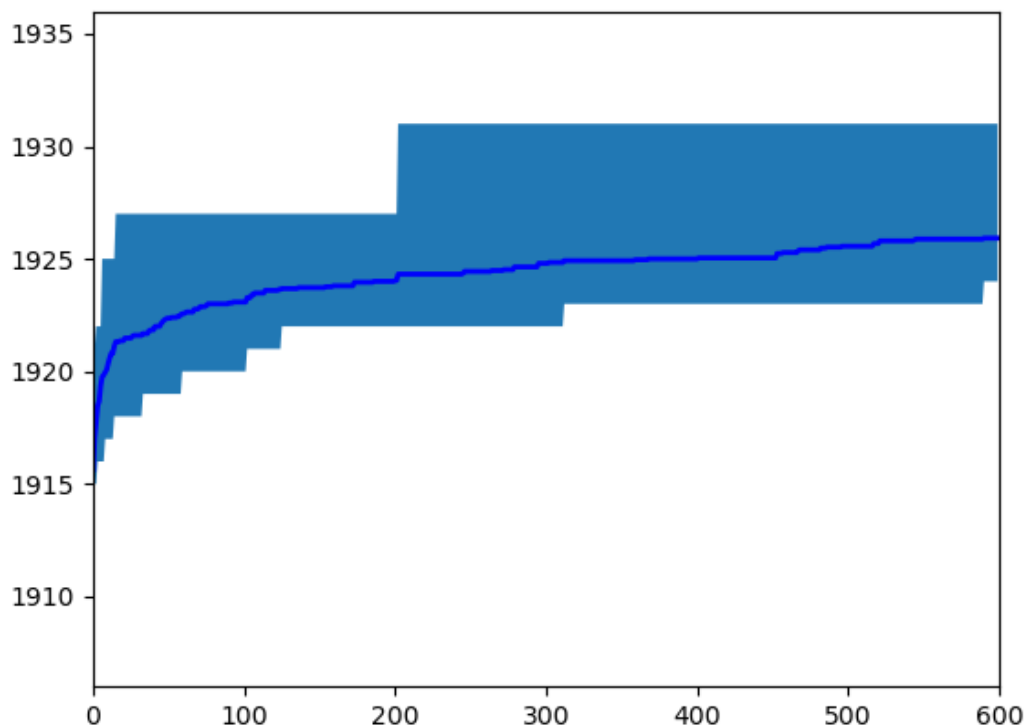
1. Opis rozwiązania

Zaimplementowano algorytm genetyczny z selekcją ruletkową, krzyżowaniem jednopunktowym oraz sukcesją generacyjną, który skutecznie rozwiązuje problemy optymalizacji funkcji celu dla jednowymiarowych binarnych wektorów decyzyjnych. Eksperymenty przeprowadzono dla zadanego problemu lądowania rakiety, na podstawie którego opracowano funkcję celu.

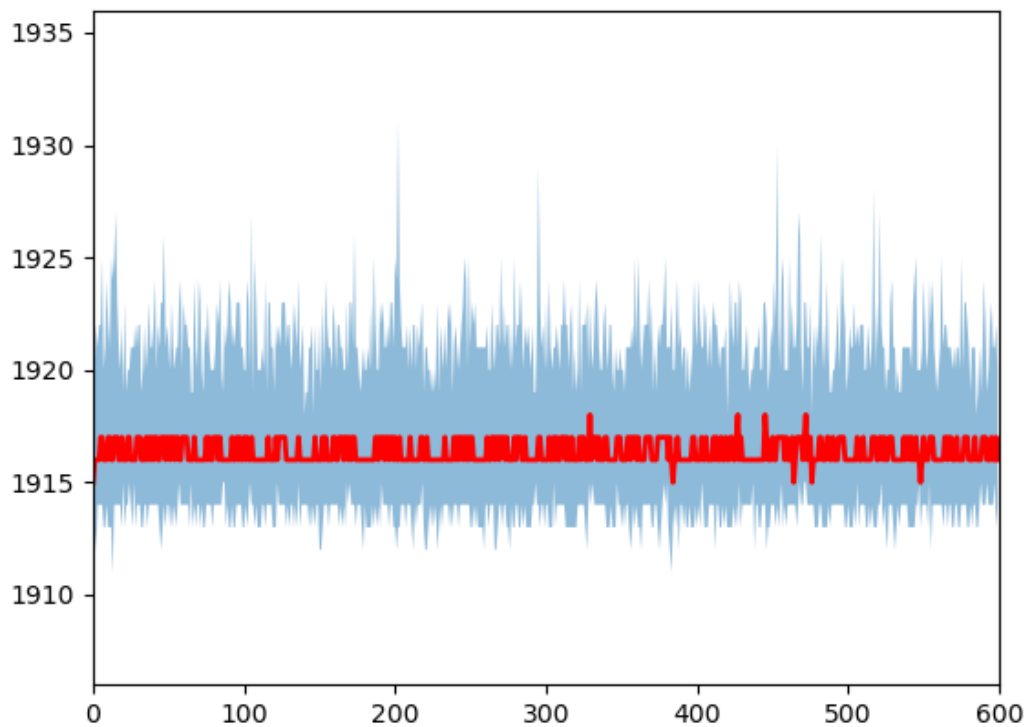
W wyniku eksperymentów, najlepsze jakościowo i wydajnościowo wyniki otrzymano dla ustalonych wartości hiperparametrów algorytmu:

- > Maksymalna liczba iteracji – 600
- > Rozmiar populacji – 400
- > Prawdopodobieństwo krzyżowania – 0.85
- > Prawdopodobieństwo mutacji – 0.15

Wszystkie symulacje uruchamiano 25 razy, co pozwala na rzetelniejsze ocenienie jakości algorytmów dla ustalonych wartości parametrów. Wyniki algorytmu dla wybranych nastaw (przedstawionych powyżej) przedstawiono na rys.1.1 i rys.1.2. Najlepszy uzyskany wynik dla podanych wartości hiperparametrów wynosi $score = 1931$ (zgodnie z punktacją w zadaniu).



Rys. 1.1. Wykres najlepszego wyniku po n-tej iteracji



Rys. 1.2. Wykres najlepszego wyniku w n-tej iteracji

Dane (zarówno te, jak i w procesie późniejszych eksperymentów) przedstawione zostały na wykresach o dwóch rodzajach:

- wykres najlepszego wyniku 'do tej pory' od iteracji (rys. 1.1)
- wykres najlepszego wyniku spośród chromosomów w populacji z aktualnej iteracji od iteracji (rys. 1.2)

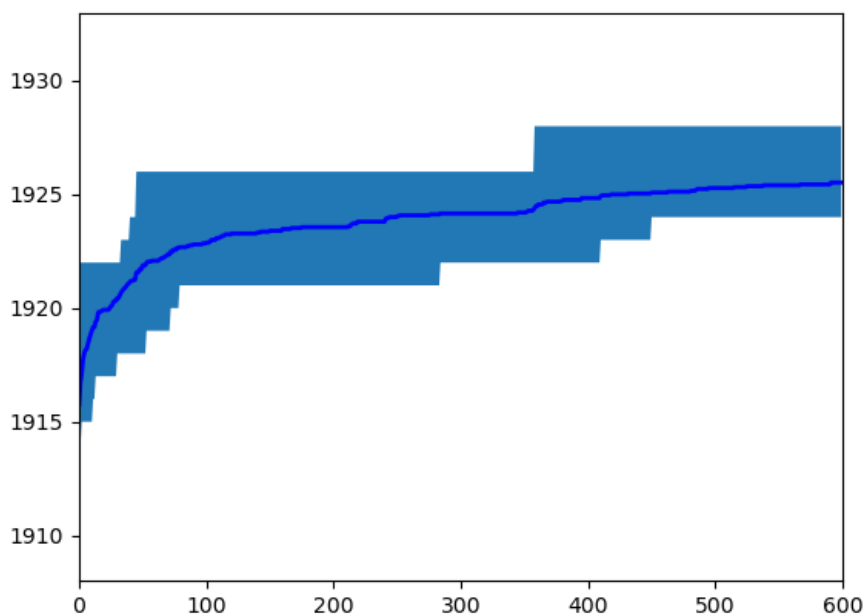
Oba wykresy biorą pod uwagę przekrój statystyczny wszystkich uruchomionych algorytmów, reprezentowany przez szare pole naniesione na tle wykresu średniego wyniku w danej iteracji.

Wybór maksymalnej liczby iteracji i liczby osobników w populacji oparto o czas obliczeń, który był niefunkcjonalnie długi dla większych wartości. Analiza wyników pokazuje wciąż progresującą wartość najlepszego wyniku uzyskanego przez algorytm, uśredniony przez 25 różnych populacji początkowych. Najlepszy globalnie wynik jednak od dłuższego czasu pozostaje niezmienny, co pozwala na zatrzymanie analizy na 600-nym kroku. Wykres 1.2 pokazuje, że największy globalnie wynik uzyskano w wyniku piku oceny chromosomów w okolicy kroku 200, później porównywalnie wysokie skoki uzyskano w krokach 300 i 480. Wnioski wyciągane z wielu różnych populacjach początkowych pokazują jak ważne jest uruchomienie algorytmu wielokrotnie.

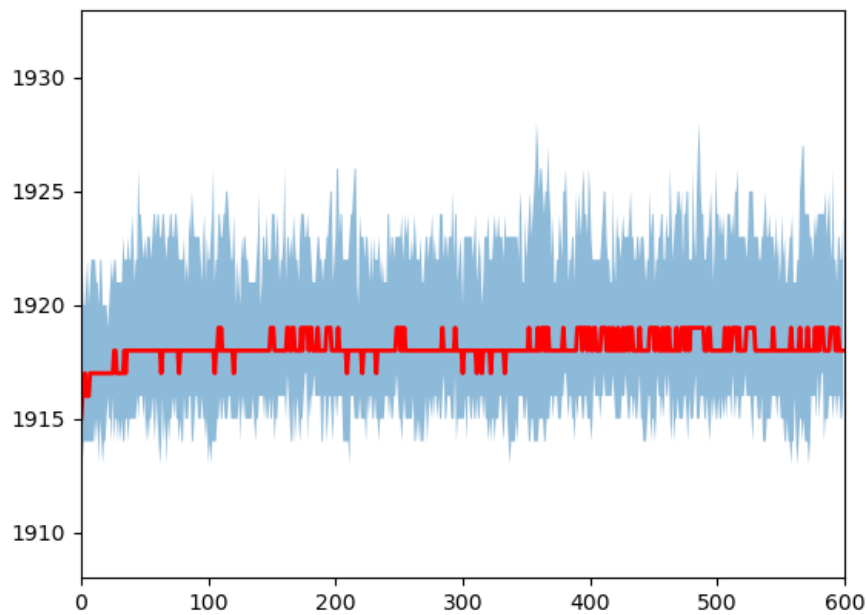
2. Badanie wpływu zmiany hiperparametrów

Następnie przeprowadzono eksperymenty, w celu zbadania wpływu zmiany wybranego hiperparametru na wyniki algorytmu. Wybrano hiperparametr **prawdopodobieństwa mutacji**, który wpływa na zmienność chromosomów względem populacji początkowej i pozwala algorytmowi na 'uciekanie' z ekstremów lokalnych w poszukiwaniu ekstremum globalnego.

Wyniki algorytmu dla prawdopodobieństwa mutacji równego $pm = 0.15$ zaprezentowano w poprzednim rozdziale. Wyniki dla $pm = 0.01$, $pm = 0.05$, $pm = 0.25$ (1928), $pm = 0.4$ (1930) i $pm = 0.8$ przedstawiono na rys.2.1-2.10

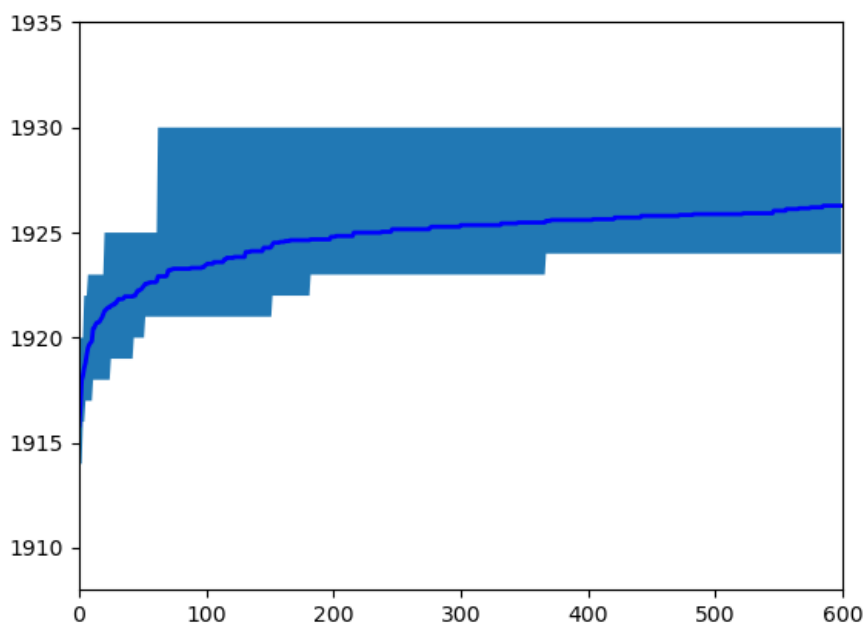
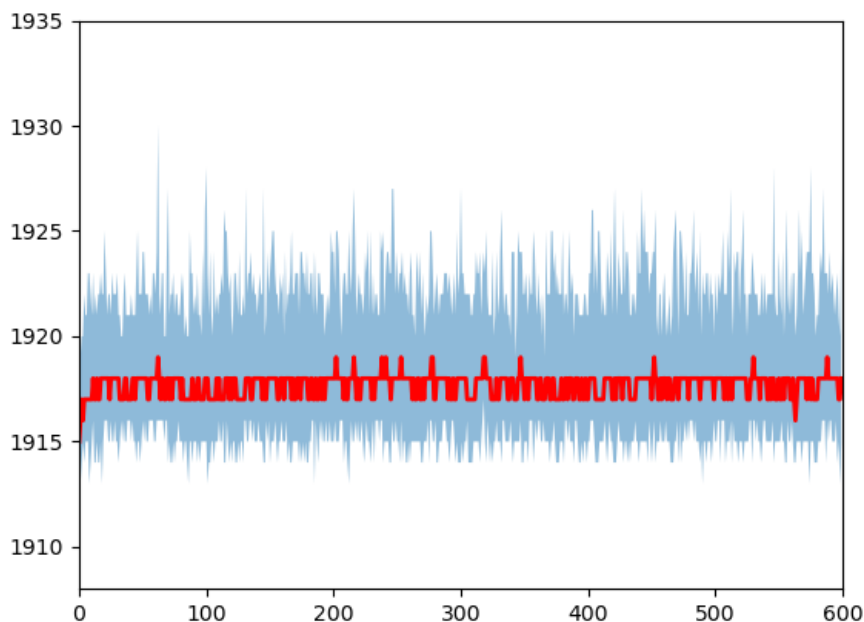


Rys. 2.1. Wykres postępu optimum wartości funkcji celu algorytmu dla $pm = 0.01$



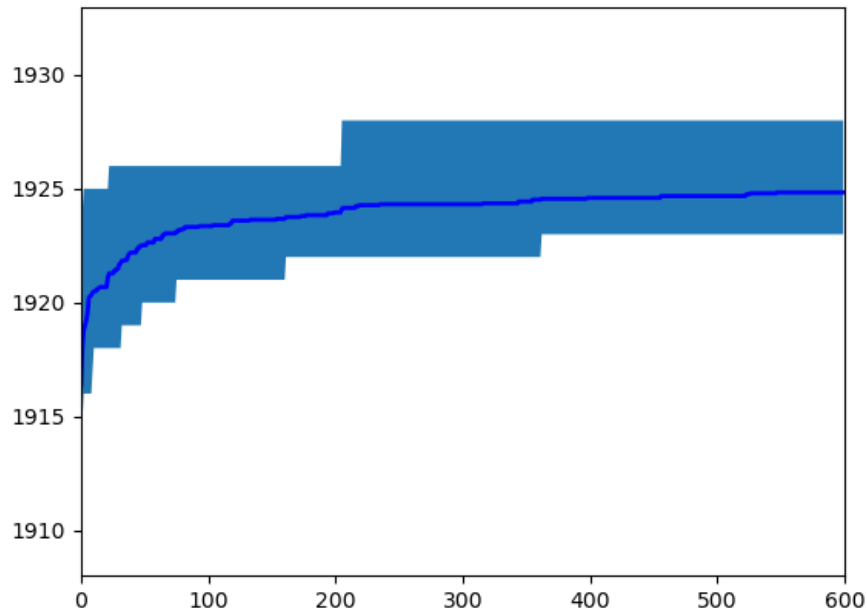
Rys. 2.2. Wykres optimum wartości funkcji celu algorytmu w danej iteracji dla $pm = 0.01$

Wykresy rys.2.1 i 2.2 przedstawiają wyniki algorytmu dla bardzo niskiego prawdopodobieństwa mutacji. Sprawia to, że wyniki algorytmu są bardzo zależne od populacji początkowej. Najlepszy osiągnięty wynik to $score = 1928$, a wykres przeglądu kolejnych populacji (rys. 2.2) potwierdza niskie odchylenie standardowe i zmiany wyników pomiędzy iteracjami. Pokazuje to, że wszystkie chromosomy "krążą" wokół tego samego obszaru, brak losowo wygenerowanych odchyłeń w kierunku lepszego wyniku.

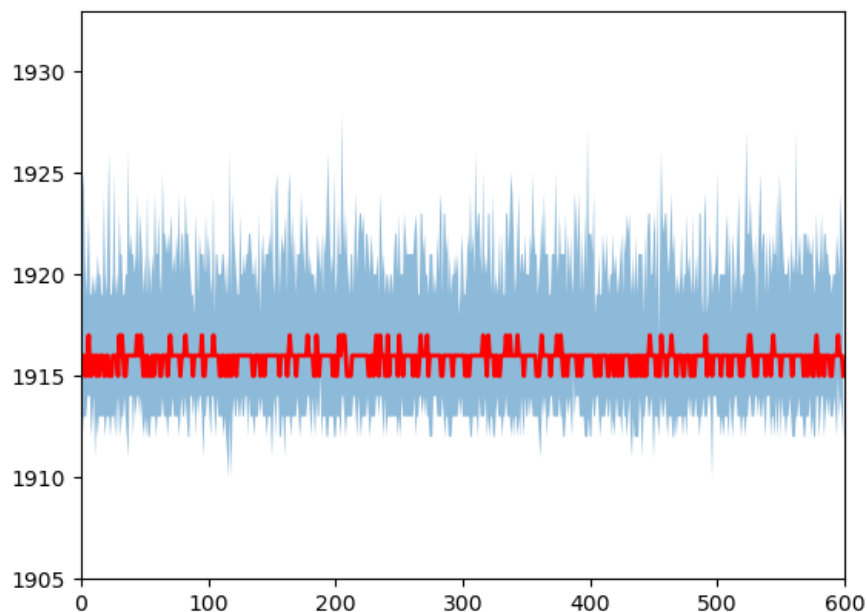
Rys. 2.3. Wykres postępu optimum wartości funkcji celu algorytmu dla $pm = 0.05$ Rys. 2.4. Wykres optimum wartości funkcji celu algorytmu w danej iteracji dla $pm = 0.05$

Wykresy rys.2.3 i 2.4 przedstawiają wyniki algorytmu dla $pm = 0.05$. Od razu rzuca się w oczy zwiększenie rozrzutu wyników pomiędzy iteracjami, względem eksperymentu dla $pm = 0.01$. Pojedyncze chromosomy mutują wewnątrz siebie, co pozwala na znalezienie lepszych rozwiązań. Ze względu na losowość, algorytm dla niektórych populacji początkowych znalazł

(wygenerował dzięki mutacji) najlepszy swój wynik ($score = 1930$) już w okolicach 70 iteracji. Wykres rys.2.3 pokazuje jednak, że prawdopodobieństwo mutacji jest wciąż na tyle niskie, że średnia z najlepszych wyników jest bliższa dolnej granicy rozkładu, co pokazuje że tylko nieliczne uruchomienia pozwoliły na szybkie znalezienie swojego optimum.

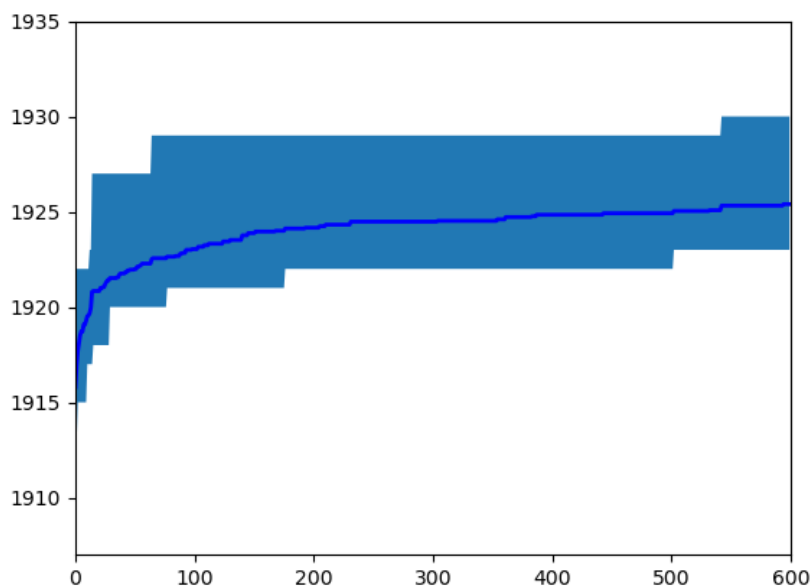


Rys. 2.5. Wykres postępu optimum wartości funkcji celu algorytmu dla $pm = 0.25$

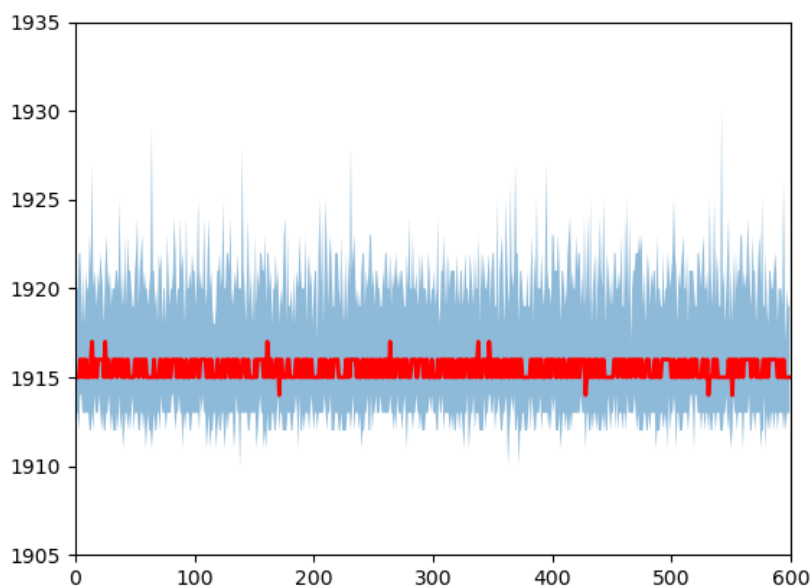


Rys. 2.6. Wykres optimum wartości funkcji celu algorytmu w danej iteracji dla $pm = 0.25$

Wykresy rys.2.5 i 2.6 pokazują negatywny wpływ za wysokiego prawdopodobieństwa mutacji. Eksperymenty przeprowadzono dla $pm = 0.25$, a osiągnięty wynik to $score = 1928$. Wykresy jednak pokazują już gorszą stronę algorytmu, średni uzyskany wynik dla 25 różnych populacji początkowych to 1924, a wykres 2.6 pokazuje, że średni wynik w każdej iteracji ma wartość w granicy $score = 1926$. Losowość pokazana przez ten wykres jest wysoka, niestety także w negatywnym kierunku. Powstały iteracje, dla których najgorszy wynik to 1910 (iteracja 1910).

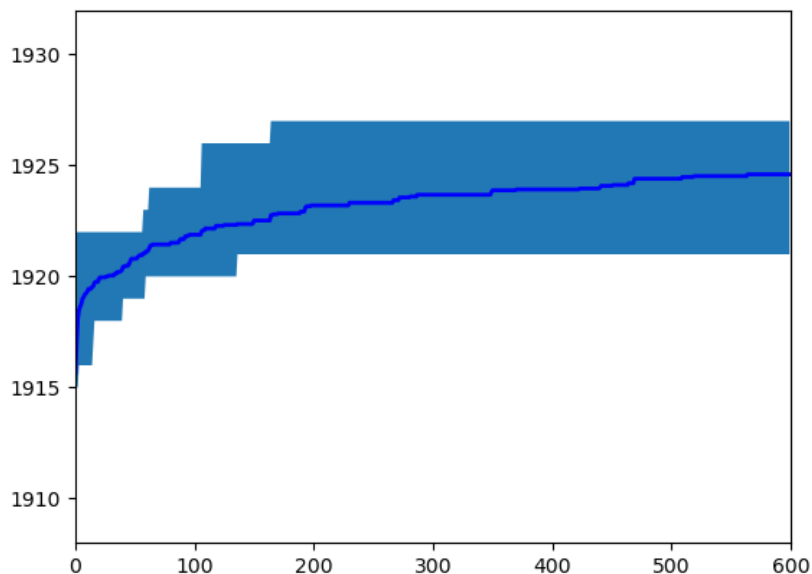


Rys. 2.7. Wykres postępu optimum wartości funkcji celu algorytmu dla $pm = 0.4$

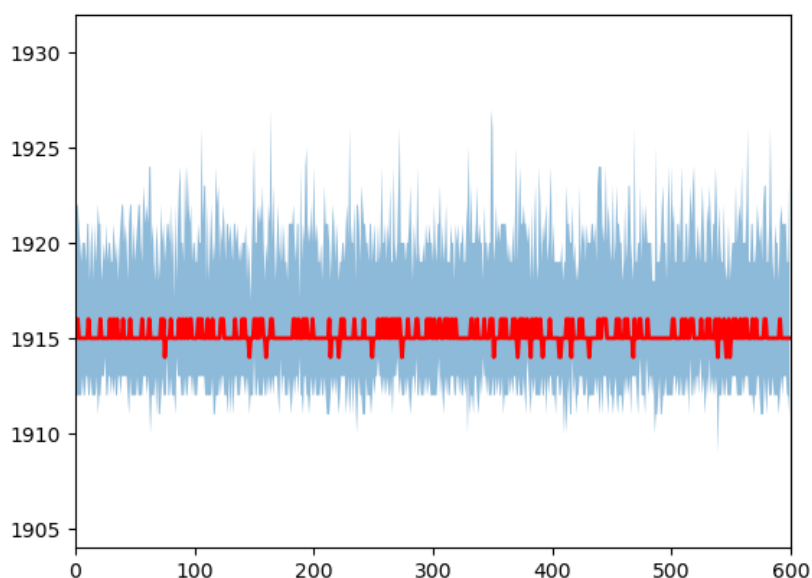


Rys. 2.8. Wykres optimum wartości funkcji celu algorytmu w danej iteracji dla $pm = 0.4$

Wykresy rys.2.7 i 2.8 to kolejny przykład nieskutecznego zwiększania wartości wspomnianego hiperparametru. osiągnięty wynik to $score = 1930$, choć wynik średni nie przekroczył $score = 1925$. Wynik średni dla danej iteracji to także okolice 1915, a piki występują zarówno w górę jak i w dół



Rys. 2.9. Wykres postępu optimum wartości funkcji celu algorytmu dla $pm = 0.8$



Rys. 2.10. Wykres optimum wartości funkcji celu algorytmu w danej iteracji dla $pm = 0.8$

Wykresy rys.2.9 i 2.10 nieznacznie różnią się od wykresów dla $pm = 0.25$. Eksperyment przeprowadzony dla $pm = 0.8$ pokazuje przebieg algorytmu, gdy większość genotypu zamienia

swoją wartość binarną. Widoczne jest to w pikach na przebiegu postępu optimum (rys.2.10), gdzie właściwie co iteracje jest wynik lepszy, a następnie gorszy. Wynika to z małej stabilności algorytmu, w kontekście sukcesji generacyjnej. Co drugie pokolenie jest do siebie bardziej podobne, niż pokolenie po pokoleniu. Najlepszy uzyskany wynik to $score = 1927$, ale średni uzyskany wynik to już tylko $score = 1924$.

3. Wnioski

Zaimplementowany algorytm genetyczny zadziałał bez zarzutów, oparto go o wcześniej wspomnianą selekcję ruletkową, krzyżowanie jednopunktowe, oraz sukcesję generacyjną. Przeprowadzone badania wpływu zmiany prawdopodobieństwa mutacji pokazały istotność tej operacji dla jakości uzyskanego wyniku przez algorytm. Bardzo małe wartości tego parametru ($pm = 0.01$, $pm = 0.05$) sprawiły, że algorytm słabo sobie radził z przeszukiwaniem przestrzeni. Dla każdej kolejnej iteracji, ze względu na selekcję ruletkową, osobniki w populacji były do siebie podobne. Tworzenie osobników dla kolejnej generacji głównie (lub jedynie) poprzez krzyżowanie jednopunktowe nie pozwala na uzyskanie różnorodności wyników. Bardzo mało jest osobników, którzy "proponują" odmienny wektor decyzji, względem tych już zaproponowanych przez poprzednie populacje. Wykres postępu optimum algorytmu po n -tej iteracji (rys. 2.1, 2.3) jest zdecydowanie bardziej wypłaszczony, a wykres najlepszych wyników w n -tej iteracji (rys. 2.2, 2.4) jest bardzo stłumiony (brak widocznych znaczących pików).

Z drugiej strony, za duże wartości tego parametru ($pm = 0.4$, $pm = 0.8$) wprowadzają za dużą losowość, co w przypadku binarnych problemów może wręcz odwracać większość wektorów decyzyjnych. Wykresy tych eksperymentów (rys. 2.7-2.8) charakteryzują się niskimi wartościami średniego wyniku w n -tej iteracji, a wyniki optimum uzyskane zostały dość późno. Wartości te, w przeciwieństwie do za niskich, pozwalają jednak na zwiększenie różnorodności rozwiązań i wyników. Dzięki temu algorytm jest w stanie przeszukać większą przestrzeń, niż zaproponowaną w populacji początkowej. Nie zatrzymuje się w ekstremach, co może być zarówno zaletą jak i wadą. Sprawia to, że najlepsze rozwiązanie może wymagać więcej iteracji do znalezienia go, ale znacząco zwiększone są no to szanse.

Wnioski pokazują, że za duże prawdopodobieństwo mutacji może negatywnie wpłynąć na szybkość znajdowania lepszych wyników, ale zwiększa szanse na końcowy sukces. Za małe prawdopodobieństwo może sprawić, że nawet dla bardzo dużej liczbie iteracji, nie znajdzie się optimum globalnego.