

Sprawozdanie z ćwiczenia 1

Algorytmy ewolucyjne i genetyczne

Yan Korzun

12 listopada 2025

1 Cel ćwiczenia

Celem ćwiczenia jest implementacja algorytmu genetycznego wykorzystującego:

- mutację,
- selekcję ruletkową,
- krzyżowanie jednopunktowe,
- sukcesję generacyjną.

Następnie badane jest działanie algorytmu na przykładzie problemu przedstawionego w rozdziale 3 instrukcji.

2 Opis zastosowanego algorytmu genetycznego

2.1 Schemat algorytmu

Poniżej przedstawiono ogólny schemat algorytmu genetycznego użytego w eksperymencie:

1. Inicjalizacja populacji losowej.
2. Ocena przystosowania osobników.
3. Warunek stopu(osiągnięcie progu jakości).
4. Sukcesja generacyjna.
5. Selekcja rodziców metodą ruletkową.
6. Krzyżowanie jednopunktowe.
7. Mutacja genów.
8. Tworzenie nowej populacji (sukcesja generacyjna).
9. Warunek stopu (liczba generacji).
10. Przejście do punktu 2.

2.2 Parametry algorytmu

- Rozmiar populacji: mu
- Maksymalna liczba generacji: $tmax$
- Prawdopodobieństwo krzyżowania: p_c
- Prawdopodobieństwo mutacji: p_m
- Metoda selekcji: ruletkowa
- Populacja startowa: losowa

3 Rozwiązania odniesienia

3.1 Trywialne rozwiązanie

Rozwiązaniem trywialnym jest ciąg jedynek i ciąg zer dla obu silników, czyli silniki są ciągle włączone przez jakiś czas, a na pozostałym odcinku czasu są wyłączone. Wartością funkcji celu tego rozwiązania jest zero albo prawie zero.

3.2 Rozwiązanie losowe

Dla losowych rozwiązań wartość funkcji celu rzadko osiągała liczby większe niż -100000 . Bardzo często wynikiem była liczba $-122500 = -350^2$, czyli obiekt pozostał się na samym starcie.

3.3 Górnne ograniczenie wyniku

Górnym ograniczeniem wynika jest zero, co wynika z funkcji celu, ale według zadania dązymy do wartości większej tylko do -4 , większy wynik jest niepotrzebny.

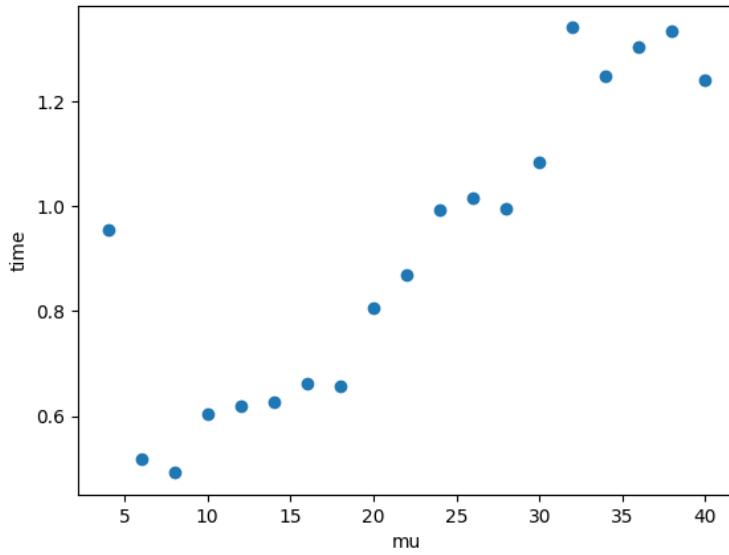
4 Eksperymenty i wyniki

4.1 Dobór hiperparametrów

Najpierw "na oko" dobrano parametry tak, aby algorytm sprawnie działał. Dla prawdopodobieństw wybrano wartości standartowe $p_m = 0.02$, $p_c = 0.8$. Parametr $mu = 10$ dawał stosunkowo szybkie znalezienie rozwiązania zadowalającego (na moim laptopie to było około $0.8s$, ale to oczywiście zależy od urządzenia).

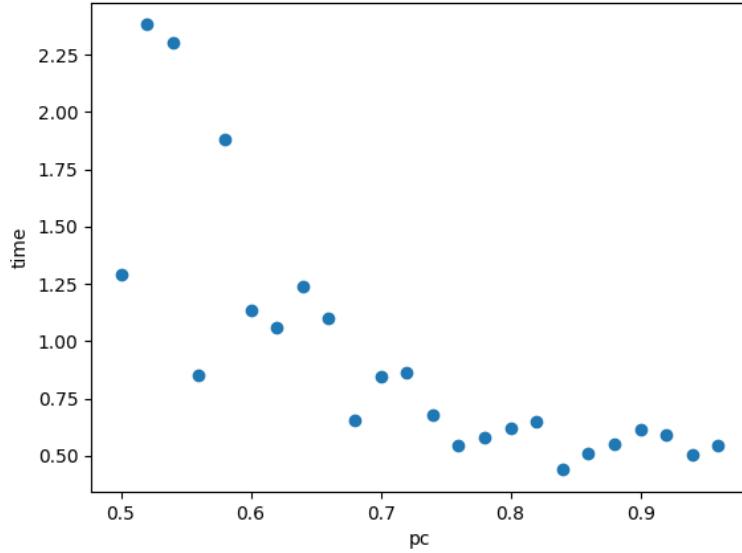
4.2 Wpływ hiperparametrów

Zbadano wpływ prawdopodobieństwa mutacji p_m , prawdopodobieństwa krzyżowania p_c oraz liczby osobników w populacji mu . Żeby zmniejszyć wpływ przypadku, w testach został użyty parametr *repeats*, który określa, ile razy trzeba powtórzyć zmierzanie czasu działania algorytmu dla różnych wartości parametru badanego. Na wykres podana zostaje wartość czasu średnia. Wyniki badania wpływu mu przedstawiono na wykresie:



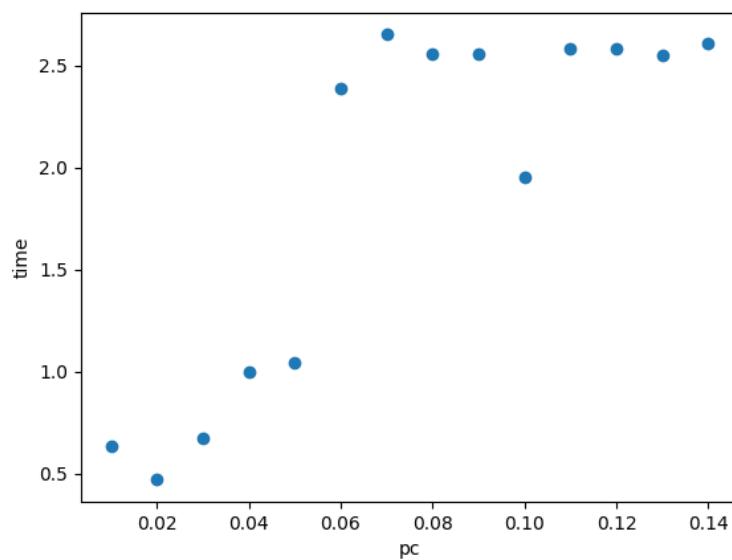
Rysunek 1: Wpływ wartości μ na czas znalezienia rozwiązania zadowalającego.

Jako parametr optymalny wybrano $\mu = 8$, zostanie użyty w dalszych testach.



Rysunek 2: Wpływ wartości p_c na czas znalezienia rozwiązania zadowalającego.

Przy wartościach parametra p_c większych od 0.8 czas wykonania algorytmu prawie się nie zmienia, losowość ma większy wpływ. Dla testów parametru p_m wybrano $p_c = 0.86$



Rysunek 3: Wpływ wartości p_m na czas znalezienia rozwiązania zadowalającego.

W przypadku p_m wartość optymalna jest jednoznaczna - 0.2. Mniejsze wartości spowolniają algorytm, większe - destabilizują go.

Zbiór optymalnych parametrów - $\mu u = 8$, $p_c = 0.86$, $p_m = 0.02$.