

Dokumentacja – laboratorium nr 2

Wprowadzenie do Sztucznej Inteligencji

Dominika Wyszyńska 318409

15 listopada 2023

1 Opis treści zadania

Celem zadania było zaprojektowanie oraz zaimplementowanie algorytmu ewolucyjnego. Zaimplementowana została strategia ewolucyjna ES(1+1), gdzie zamiast populacji początkowej dany jest tylko jeden osobnik początkowy x_0 . Algorytm powinien mieć możliwość obsługi różnych parametrów, takich jak maksymalna liczba iteracji lub zasięg mutacji σ . Dodatkowo została zaimplementowana reguła 1/5 sukcesów. Implementacja solwera powinna być w stanie zoptymalizować każdą zadaną funkcję dla różnych wymiarów przestrzeni R^n .

Należało także zbadać zbieżność algorytmu na funkcjach F1 oraz F9 z benchmarku CEC2017 dla wymiarowości $n = 10$. W związku z tym należało zamieścić wykresy zależności wartości funkcji w punkcie w danym kroku od numeru iteracji/kroku. Na sam koniec należało porównać zbieżność algorytmu ewolucyjnego na wskazanych funkcjach z własną implementacją algorytmu gradientu prostego.

Implementacja programu została umieszczona na [gitlabie wydziałowym](#).

2 Opis planowanych eksperymentów numerycznych

Na potrzeby eksperymentów miało zostać przyjęte $n = 10$, czy przestrzeń 10-wymiarowa oraz także stały punkt startowy – przykładowo wektor $x_0 = [1, 1, 1, 1, 2, 1, 1, 1, 1, 1]$. W ramach przeprowadzonych eksperymentów przetestowano algorytm, wykonując serię około 10 testów dla każdego przypadku. Celem było uniknięcie wyciągania wniosków na podstawie pojedynczego uruchomienia, co mogłoby doprowadzić do błędnej analizy. Podczas analizy skupiono się na średnich wynikach, z pominięciem wartości maksymalnych i minimalnych. Taka strategia miała na celu dostarczenie bardziej reprezentatywnego obrazu efektywności algorytmu, eliminując ekstremalne przypadki, które mogą wprowadzać błędne interpretacje.

Eksperymenty zostały zaprojektowane w następujący sposób:

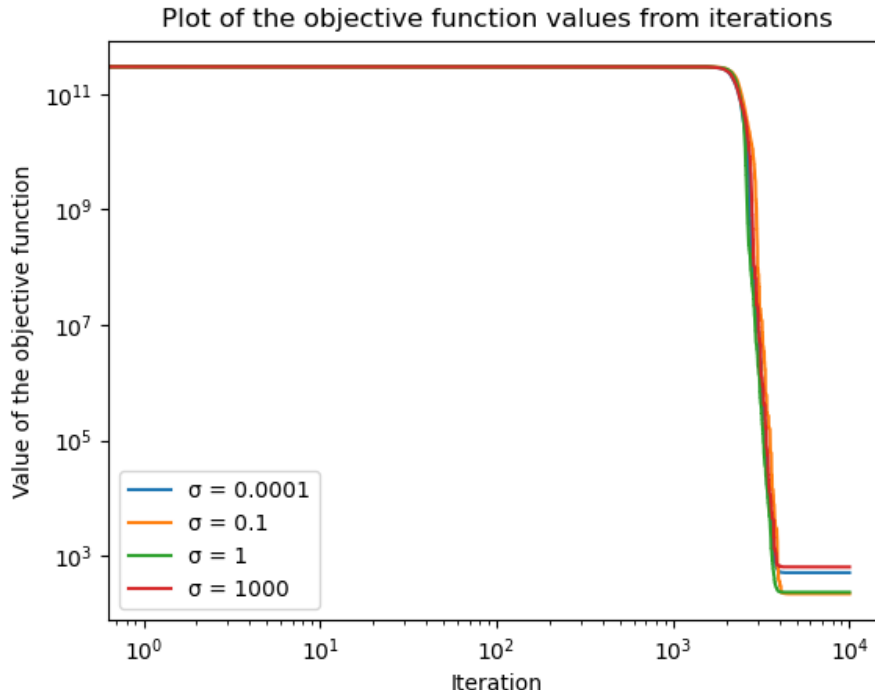
- Analiza wpływu różnych zasięgów mutacji σ .
- Badanie różnej liczby maksymalnych iteracji.
- Badanie wpływu współczynnika `adaptation_interval` w regule 1/5 sukcesu.

3 Opis uzyskanych wyników

3.1 Analiza wpływu różnych zasięgów mutacji σ .

Eksperymenty zostały przeprowadzone dla stałej wartości liczby maksymalnych iteracji równej 10000 oraz `adaptation_interval` o wartości 30 dla funkcji F1, a 10 dla funkcji F9.

- Dla funkcji F1:

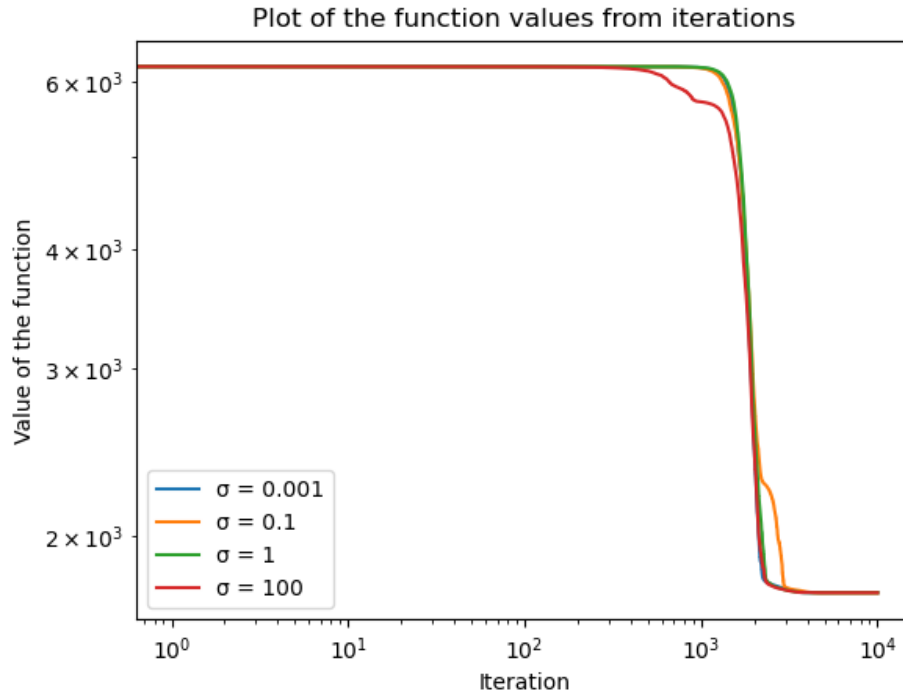


Rysunek 1: Wykres zależności średniej wartości funkcji od iteracji dla różnych wartości σ .

Krok mutacji σ	Średnia wartość funkcji w punkcie końcowym
0.0001	882.89
0.1	458.77
1.0	388.85
1000	1394.53

Algorytm ewolucyjny 1+1 przemieszcza się przez przestrzeń rozwiązań, rozpoczynając od częściowo losowych generacji, co umożliwia eksplorację różnorodnych możliwości. Na początku rozwiązania niekoniecznie skupione są wokół optimum. Jednak w miarę postępu czasu, algorytm zaczyna koncentrować się na obiecujących obszarach, co przyspiesza poszukiwania optymalnego rozwiązania. To podejście pomaga w efektywnym przeszukiwaniu przestrzeni rozwiązań i znalezieniu optymalnego punktu w bardziej efektywny sposób. Proces ten widać na wykresach w danym podpunkcie.

- Dla funkcji F9:



Rysunek 2: Wykres zależności średniej wartości funkcji od iteracji dla różnych wartości σ .

Krok mutacji σ	Średnia wartość funkcji w punkcie końcowym
0.001	1746.79
0.1	1746.35
1.0	1748.35
100	1687.49

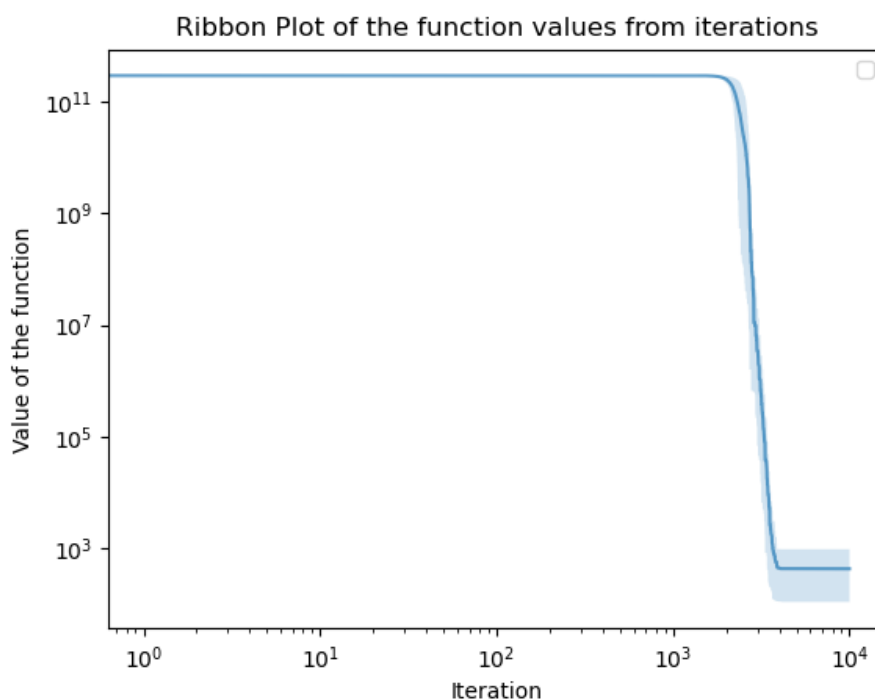
Wielkość kroku mutacji (oznaczana jako σ) wpływa na eksplorację i eksploatację przestrzeni rozwiązań. Zbyt duża wartość sigma może doprowadzić do tego, że ciężko będzie znaleźć optimum, słaba eksploatacja. Natomiast zbyt mała wartość może sprawić, że algorytm utknie w ekstremum lokalnym. Dla problemów z wieloma lokalnymi minimami, większe sigma mogą pomóc uniknąć utknięcia w minimach. Natomiast dla problemów z jednym globalnym minimum, mniejsze sigma mogą skoncentrować się na precyzyjnym lokalizowaniu tego minimum.

3.2 Badanie różnej liczby iteracji.

Maksymalna liczba iteracji w algorytmie ewolucyjnym wpływa na jego zdolność do zbieżności i skuteczność. Zbyt niska liczba iteracji może uniemożliwić znalezienie optymalnego rozwiązania. Natomiast odpowiednio dostosowana liczba iteracji pozwala na pełne przeszukanie przestrzeni rozwiązań. Jednocześnie wyższa liczba iteracji oznacza większe zużycie zasobów obliczeniowych i dłuższy czas wykonania.

- Dla funkcji F1:

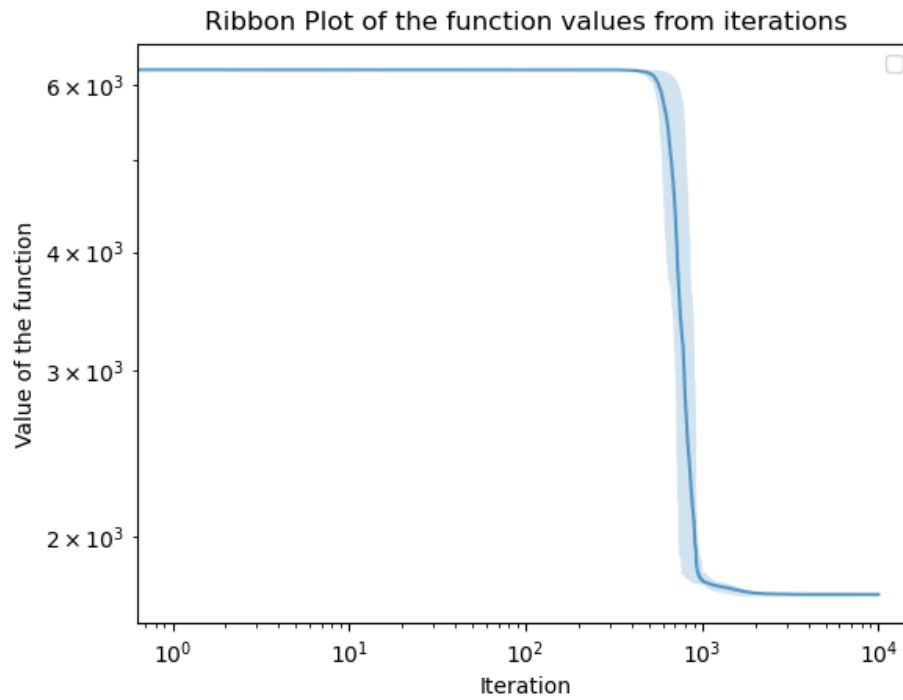
Max liczba iteracji	Średnia wartość funkcji w punkcie końcowym
100	$2.91e12$
1000	$1.02e11$
5000	$1.71e3$
10000	$1.69e3$



Rysunek 3: Wykres zależności średniej wartości funkcji od iteracji dla funkcji F1.

- Dla funkcji F9:

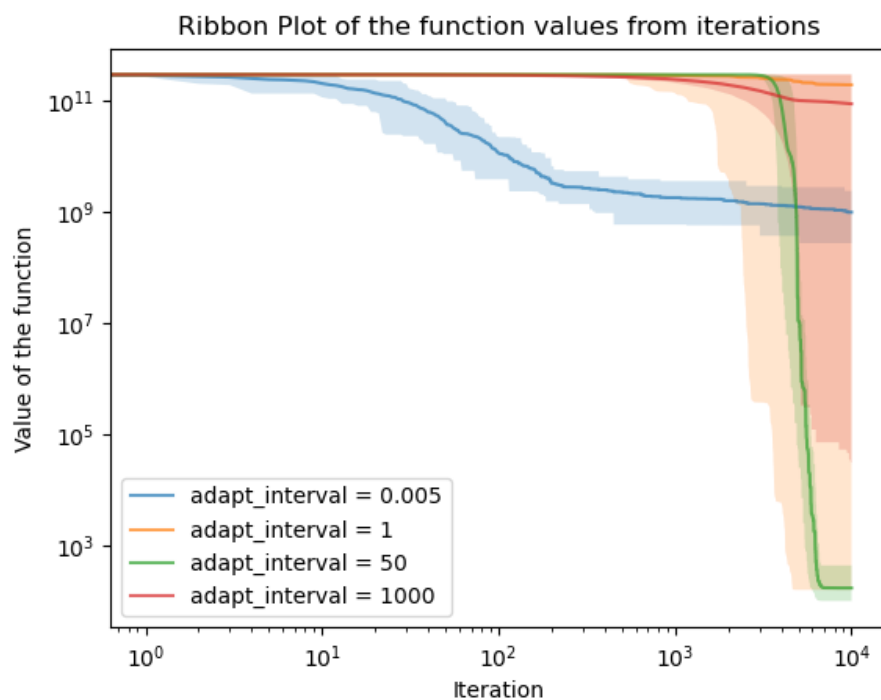
Max liczba iteracji	Średnia wartość funkcji w punkcie końcowym
100	$5.13e3$
500	$3.94e3$
5000	$1.69e3$
10000	$1.65e3$



Rysunek 4: Wykres zależności średniej wartości funkcji od iteracji dla funkcji F9.

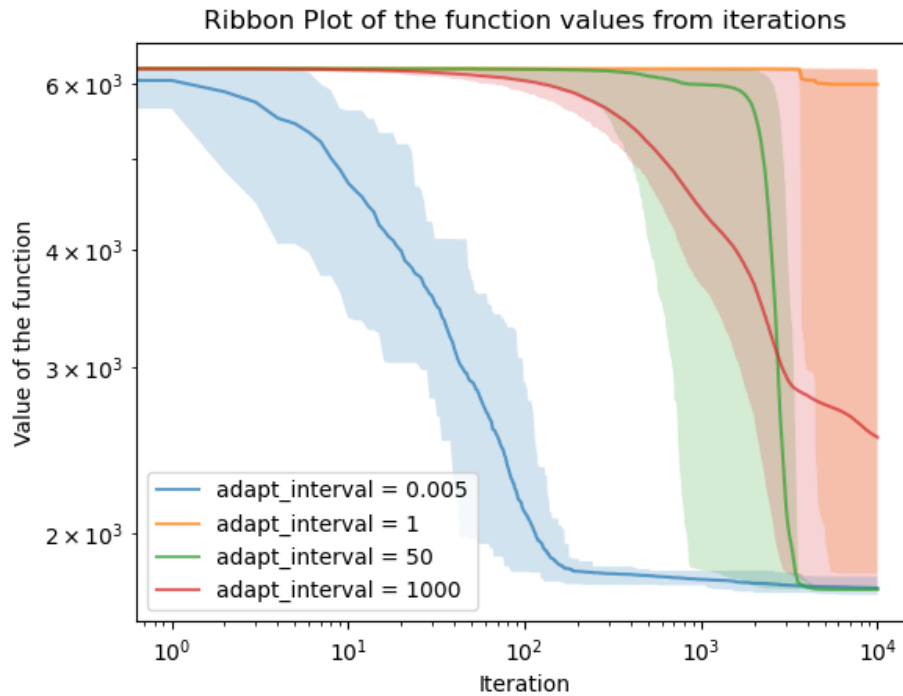
3.3 Badanie wpływu współczynnika `adaptation_interval` w regule 1/5 sukcesu.

- Dla funkcji F1:



Rysunek 5: Wykres zależności średniej wartości funkcji od iteracji dla różnych wartości `adaptation_interval` wraz z przedziałami ufności.

- Dla funkcji F9:

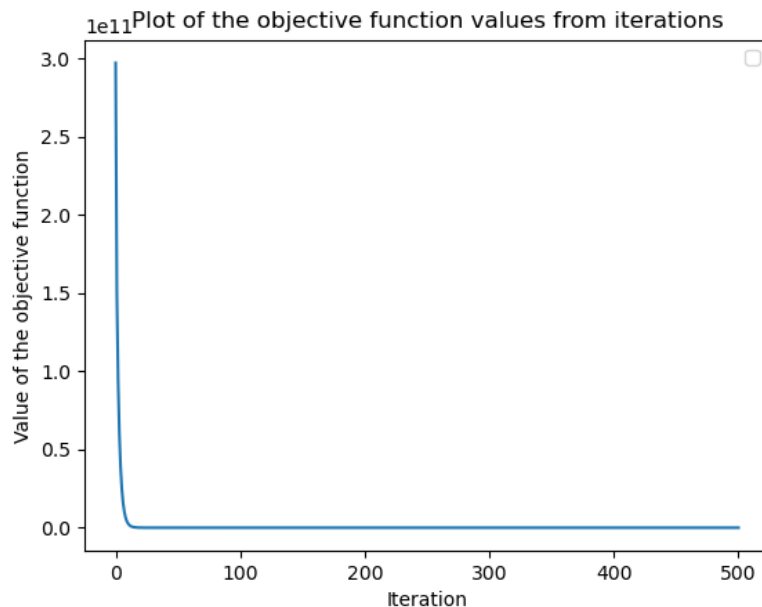


Rysunek 6: Wykres zależności średniej wartości funkcji od iteracji dla różnych wartości `adaptation_interval` wraz z przedziałami ufności.

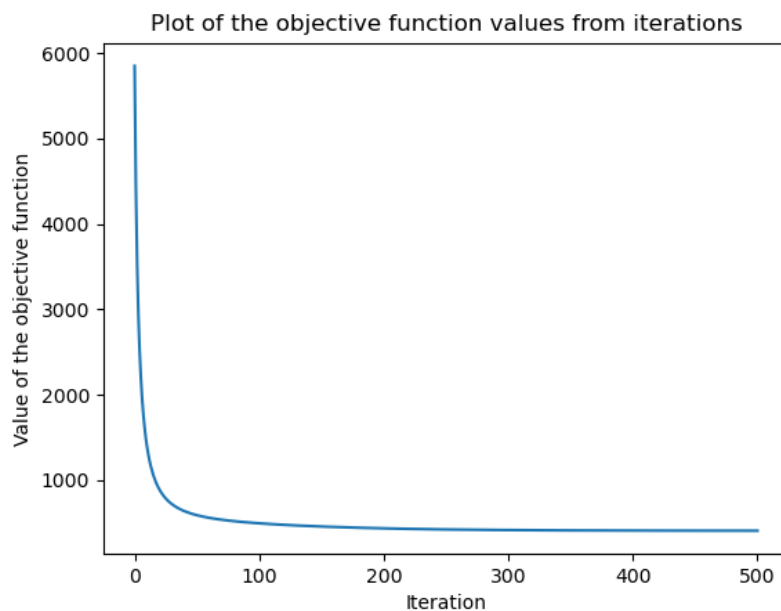
Współczynnik `adaptation_interval` w regule 1/5 sukcesu wpływa na zbieżność algorytmu ewolucyjnego. Odpowiednio dostosowany może przyspieszyć zbieżność poprzez dynamiczne dostosowywanie wielkości mutacji (σ) w zależności od sukcesów i porażek mutacji. Optymalna wartość parametru wymaga eksperymentalnego dostosowania. Algorytm może być bardziej elastyczny w dostosowywaniu się do zmieniającej się charakterystyki funkcji celu. Wartości zbyt duże lub zbyt małe mogą prowadzić do suboptymalnych wyników.

4 Porównanie z algorytmem gradientu prostego

Eksperymenty zostały przeprowadzone dla dwóch różnych funkcji (F1 oraz F4) z benchmarku CEC2017 dla wymiarowości $n = 10$ przy stałej wartości epsilon równej $1e - 6$, bety stałej równej $1e - 8$ (dla F1) oraz $1e - 2$ (dla F4), maksymalnej liczbie iteracji 500 oraz przy wszystkich aktywnych warunkach stopu.



Rysunek 7: Wykres zależności wartości funkcji od iteracji dla funkcji F1 dla algorytmu gradientu prostego.



Rysunek 8: Wykres zależności wartości funkcji od iteracji dla funkcji F4 dla algorytmu gradientu prostego.

Po max 500 iteracjach algorytm gradientu prostego dla funkcji F1 osiąga wartość 101,53. Natomiast dla F4 osiąga wartość funkcji w punkcie końcowym równą 407,75.

Przy obu funkcjach nie było większych problemów z wybraniem odpowiednich parametrów, jednakże szybkość zbiegania do optimum znacząco zależy od wartości beta. Wartość funkcji dość szybko zbiega do wartości bliskiej zeru. Natomiast funkcje, które nie są różniczkowalne - m.in. funkcja F9, mogą okazać się problematyczne dla algorytmu gradientu prostego.

Algorytm gradientu prostego jest efektywny w przypadku gładkich funkcji celu, różniczkowalnych, z jednym wyraźnym minimum lokalnym, szybko zbiega do rozwiązania. Jednak może napotykać problemy z zbieżnością w przypadku funkcji celu zawierających lokalne minima lub gdy są źle uwarunkowane. Natomiast algorytm ewolucyjny nie ma pewnych ograniczeń, jakie miał algorytm gradientu prostego. Strategia ewolucyjna ES(1+1) jest bardziej elastyczna, radzi sobie dobrze z funkcjami celu o nieregularnej strukturze, wieloma lokalnymi minimami lub w przypadku źle uwarunkowanej przestrzeni rozwiązań. Może jednak wymagać więcej iteracji do zbieżności niż algorytm gradientu prostego (wnioskując z poprzednich eksperymentów), zwłaszcza w przypadku gładkich funkcji celu z jednym minimum. Ostateczny wybór zależy od charakterystyki funkcji celu i wymagań zadania optymalizacyjnego.

5 Podsumowanie i wnioski

Celem tych eksperymentów było zrozumienie wpływu różnych parametrów na zachowanie się algorytmu ewolucyjnego. Główną zaletą strategii ES(1+1) jest adaptacja zasięgu mutacji σ oraz prostota implementacji. Natomiast wadą jest to, że relatywnie łatwo utyka w optimum lokalnych. Wnioski wskazują, że optymalne dostosowanie zasięgu mutacji i liczby maksymalnych iteracji może istotnie wpłynąć na skuteczność algorytmu, umożliwiając lepszą równowagę między eksploracją a eksploatacją przestrzeni rozwiązań. Zbyt mały zasięg mutacji σ może prowadzić do utknięcia w lokalnych minimach, podczas gdy zbyt duży może utrudnić zbieżność do optymalnego rozwiązania. Liczba maksymalnych iteracji ma istotny wpływ na skuteczność algorytmu. Optymalne ograniczenie liczby iteracji pozwalało na pełne przeszukanie przestrzeni rozwiązań, jednocześnie unikając nadmiernego zużycia zasobów. W kontekście reguły 1/5 sukcesu, *adaptation_interval* okazał się decydującym parametrem wpływającym na adaptacyjność algorytmu.