

Elitarny Algorytm Genetyczny (EGA) dla Problemu Zbioru Plonów Zad 1b

Michał Zarzycki 184856 , Michał Kolada 189729

1 Cel Projektu

Celem projektu było zaimplementowanie Elitarnego Algorytmu Genetycznego (EGA) w celu rozwiązania problemu zbioru plonów, który jest zadaniem optymalizacji z ograniczeniami. Problem ten polega na maksymalizacji funkcji celu, która reprezentuje całkowita wydajność zbioru, przy jednoczesnym spełnieniu równania wzrostu oraz nałożonych ograniczeń. W szczególności celem było uzyskanie maksymalnej wydajności zbioru, przy zachowaniu stanu początkowego i końcowego równego oraz nieujemnych wartości dla zbiorów.

Wyniki porównano z rozwiązaniem przedstawionym w książce: *Michalewicz Z., (1996): Algorytmy genetyczne + struktury danych = programy ewolucyjne, WNT Warszawa.*

2 Opis Problemu

Problem zbioru plonów można opisać następującą funkcją celu:

$$J = \sum_{k=0}^{N-1} \sqrt{u_k} \quad (1)$$

przy ograniczeniach:

- **Równanie wzrostu:**

$$x_{k+1} = a \cdot x_k - u_k \quad (2)$$

- **Ograniczenia sterowania:**

$$u_k \geq 0, \quad \text{dla } k = 0, 1, \dots, N-1 \quad (3)$$

- **Ograniczenie równości:**

$$x_0 = x_N \quad (4)$$

Gdzie:

- J — funkcja celu reprezentująca całkowita wydajność zbioru,
- u_k — ilość zbioru w okresie k ,
- x_k — stan systemu w okresie k ,
- a — współczynnik wzrostu,
- N — liczba okresów,
- x_0 — stan początkowy.

2.1 Rozwiązanie Analityczne

Dla tego problemu można wyprowadzić rozwiązanie analityczne, które pozwala obliczyć optymalne wartości sterowań u_k oraz maksymalna wartość funkcji celu J^* .

2.1.1 Optymalna Wartość Funkcji Celu

Optymalna wartość funkcji celu J^* oblicza się ze wzoru:

$$J^* = \sqrt{\frac{x_0 (a^N - 1)^2}{(a - 1)a^{N-1}}} \quad (5)$$

2.1.2 Optymalne Sterowania

Optymalne sterowania u_k dla $k = 1, 2, \dots, N$ wyrażają się wzorem:

$$u_k = \frac{x_0 (a^N - 1)}{N \cdot a^{N-k}} \quad (6)$$

2.2 Obliczenia Dla Parametrów Problemu

Dla danych:

- $a = 1,1$,
- $x_0 = 100$,
- Różne wartości N (np. $N = 2, 4, 10, 20, 45$).

2.2.1 Przykład Obliczeń dla $N = 2$

Obliczamy optymalną wartość funkcji celu oraz sterowania.

Optymalna Wartość Funkcji Celu

$$\begin{aligned} J^* &= \sqrt{\frac{100 (1,1^2 - 1)^2}{(1,1 - 1) \cdot 1,1^{2-1}}} \\ &= \sqrt{\frac{100 (1,21 - 1)^2}{0,1 \cdot 1,1}} \\ &= \sqrt{\frac{100 \cdot 0,0441}{0,11}} \\ &= \sqrt{\frac{4,41}{0,11}} \\ &= \sqrt{40,0909} \\ &\approx 6,3317 \end{aligned}$$

Optymalne Sterowania Dla $k = 1$:

$$\begin{aligned} u_1 &= \frac{100 (1,1^2 - 1)}{2 \cdot 1,1^{2-1}} \\ &= \frac{100 \cdot 0,21}{2 \cdot 1,1} \\ &= \frac{21}{2,2} \\ &\approx 9,5455 \end{aligned}$$

Dla $k = 2$:

$$\begin{aligned}
u_2 &= \frac{100(1,1^2 - 1)}{2 \cdot 1,1^{2-2}} \\
&= \frac{100 \cdot 0,21}{2 \cdot 1,1^0} \\
&= \frac{21}{2} \\
&= 10,5
\end{aligned}$$

2.2.2 Interpretacja Wyników

Dla $N = 2$, optymalne sterowania to $u_1 \approx 9,5455$ oraz $u_2 = 10,5$, a maksymalna wartość funkcji celu wynosi $J^* \approx 6,3317$.

2.2.3 Obliczenia dla Innych Wartości N

Podobne obliczenia można przeprowadzić dla innych wartości N . Na przykład dla $N = 4$, $N = 10$, $N = 20$ i $N = 45$, co pozwala porównać wyniki algorytmu EGA z rozwiązaniem analitycznym.

2.3 Znaczenie Rozwiązania Analitycznego

Rozwiązanie analityczne stanowi punkt odniesienia dla wyników uzyskanych za pomocą algorytmu genetycznego. Pozwala to na ocenę skuteczności i dokładności algorytmu EGA w rozwiązywaniu problemu zbioru plonów.

3 Elitarny Algorytm Genetyczny (EGA)

Elitarny Algorytm Genetyczny (EGA) jest jednym z wariantów algorytmów genetycznych, który korzysta z idei elitarności, polegającej na bezpośrednim przenoszeniu najlepszych osobników do następnej generacji. Celem algorytmu było zoptymalizowanie wartości zbiorów u_k , aby zmaksymalizować funkcję celu, spełniając wszystkie ograniczenia.

3.1 Parametry Algorytmu

- **Parametry problemu:**
 - Współczynnik wzrostu $a = 1,1$
 - Stan początkowy $x_0 = 100$
 - Liczba okresów $N = [2, 4, 10, 20, 45]$
- **Parametry algorytmu:**
 - **Rozmiar populacji:** 20 000 osobników
 - **Liczba generacji:** 500
 - **Współczynnik mutacji:** 0,1
 - **Współczynnik elitarności:** 5%

3.2 Populacja Osobników

- **Reprezentacja osobników:** Każdy osobnik był wektorem rzeczywistych liczb reprezentujących wartości zbiorów w poszczególnych okresach ($u = [u_0, u_1, \dots, u_{N-1}]$).
- **Inicjalizacja populacji:** Wartości u_k były inicjalizowane losowo z zakresu $[0, a \cdot x_0]$, przy zachowaniu ograniczeń problemu, jak zaimplementowano w funkcji `initialize_population`.

3.3 Funkcja Celu i Ograniczenia

Funkcja celu była maksymalizowana zgodnie ze wzorem:

$$J = \sum_{k=0}^{N-1} \sqrt{u_k} \quad (7)$$

Do oceny przystosowania osobników użyto funkcji `calculate_fitness`, która uwzględniała kary za niespełnienie ograniczeń, takie jak nieujemność wartości zbiorów i stanów oraz spełnienie warunku $x_0 = x_N$.

3.4 Selekcja, Krzyżowanie i Mutacja

- **Selekcja:** Selekcja turniejowa została zaimplementowana w funkcji `tournament_selection`, gdzie osobniki były wybierane w grupach po 5, a do następnej generacji wybierano osobniki o najwyższej wartości funkcji celu.
- **Krzyżowanie:** Zastosowano zaawansowane krzyżowanie BLX-Alpha dla genów rzeczywistych, co pozwoliło na efektywną eksplorację przestrzeni rozwiązań.
- **Mutacja:** Do potomstwa stosowano mutacje niejednorodną, która dynamicznie zmieniała zakres mutacji w zależności od numeru generacji, co umożliwiało precyzyjniejsze dostrajanie rozwiązań w późniejszych etapach algorytmu.

3.5 Elitarność

Najlepsze 5% osobników było bezpośrednio przenoszone do następnej generacji bez zmian. Elitarność była stosowana w każdej iteracji algorytmu, co umożliwiało zachowanie najlepszych rozwiązań przez cały proces ewolucji.

4 Implementacja Algorytmu

Algorytm został zaimplementowany jako klasa `EGA`. Algorytm iterował przez określoną liczbę generacji, w każdej z nich obliczając przystosowanie, wybierając elite, przeprowadzając selekcję, krzyżowanie i mutację oraz zapewniając spełnienie wszystkich ograniczeń. Do obliczeń wykorzystano akcelerację na GPU (`cuda`), co znacząco przyspieszyło proces ewolucji dla dużych rozmiarów populacji.

5 Porównanie Wyników z Rozwiązaniem Analitycznym

W celu weryfikacji działania algorytmu obliczono również rozwiązanie analityczne, zaimplementowane w pliku `analytical_solution.py`. Rozwiązanie to pozwalało uzyskać optymalne wartości zbiorów i funkcji celu, z którymi porównano wyniki uzyskane przez EGA. W poniższej tabeli przedstawiono porównanie wyników dla różnych wartości N :

N	Wartość funkcji celu (EGA)	Wartość funkcji celu (analityczna)	Różnica
2	6,3374	6,3317	0,0057
4	12,7257	12,7210	0,0047
10	32,6488	32,8209	0,1721
20	70,1728	73,2377	3,0649
45	98,4216	279,2752	180,8535

Table 1: Porównanie wartości funkcji celu uzyskanych przez EGA i metodę analityczną

6 Wyniki i Wnioski

- **Dla małych wartości N (2 i 4),** algorytm EGA zdołał znaleźć rozwiązania bardzo zbliżone do optymalnych wartości analitycznych, co potwierdza niewielkie różnice między wartościami funkcji celu.

- Dla średnich wartości N (10 i 20), różnice między wynikami EGA a rozwiązaniem analitycznym są większe, ale nadal akceptowalne. Algorytm wymaga większej liczby generacji lub dostrojenia parametrów dla lepszej dokładności.
- Dla dużych wartości N (45), różnica jest znacząca. Algorytm EGA nie zdołał znaleźć rozwiązania bliskiego optymalnemu w zadanej liczbie generacji. Może to wynikać z większej złożoności problemu przy dużym N i konieczności zastosowania zaawansowanych technik optymalizacji.
- Konwergencja algorytmu jest zauważalna na wykresach dla mniejszych N , gdzie wartość funkcji celu stabilizuje się po pewnej liczbie generacji. Dla większych N algorytm nadal wykazuje tendencję wzrostową nawet w późnych generacjach.
- Wpływ parametrów algorytmu: Duży rozmiar populacji (20 000) oraz wykorzystanie zaawansowanego krzyżowania BLX-Alpha pozytywnie wpłynęły na zdolność eksploracji przestrzeni rozwiązań.

7 Wykresy i Wizualizacje

Przy pomocy narzędzia `Visualization.py` przedstawiono następujące wykresy dla różnych wartości N . Wykresy te pozwalają na analizę działania algorytmu EGA oraz porównanie jego wyników z rozwiązaniem analitycznym.

7.1 Konwergencja algorytmu

Poniższe wykresy przedstawiają zmianę wartości funkcji celu w kolejnych generacjach dla różnych wartości N . Możemy z nich wywnioskować, jak szybko algorytm zbliża się do optymalnego rozwiązania oraz czy osiąga stabilizację.

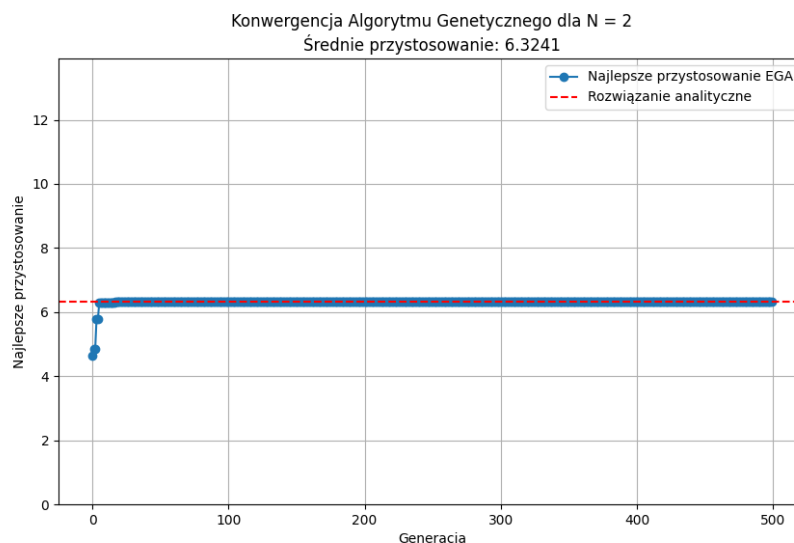


Figure 1: Konwergencja algorytmu dla $N = 2$

Dla $N = 2$ algorytm bardzo szybko osiąga wartość funkcji celu zbliżoną do optymalnej. Już po kilkudziesięciu generacjach następuje stabilizacja, co świadczy o skuteczności algorytmu dla małych rozmiarów problemu.

W przypadku $N = 4$ obserwujemy podobną tendencję. Algorytm konverguje nieco wolniej niż dla $N = 2$, jednak nadal szybko osiąga wartość funkcji celu bliska optymalnej.

Dla $N = 10$ konwergencja jest wolniejsza. Wartość funkcji celu rośnie stopniowo przez wiele generacji. Algorytm potrzebuje więcej czasu, aby zbliżyć się do wartości optymalnej, co wynika z większej złożoności problemu.

Przy $N = 20$ algorytm wykazuje jeszcze wolniejszą konwergencję. Wartość funkcji celu nadal rośnie pod koniec procesu ewolucji, co sugeruje, że zwiększenie liczby generacji lub dostrojenie parametrów mogłoby poprawić wyniki.

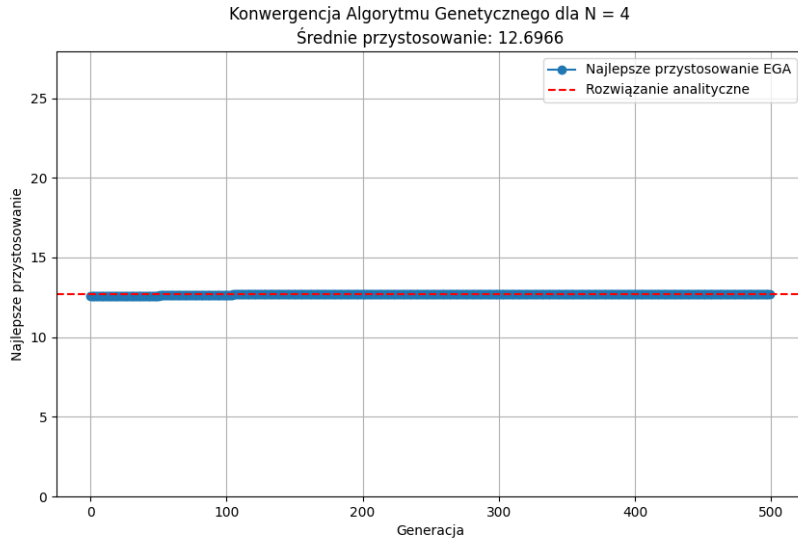


Figure 2: Konwergencja algorytmu dla $N = 4$

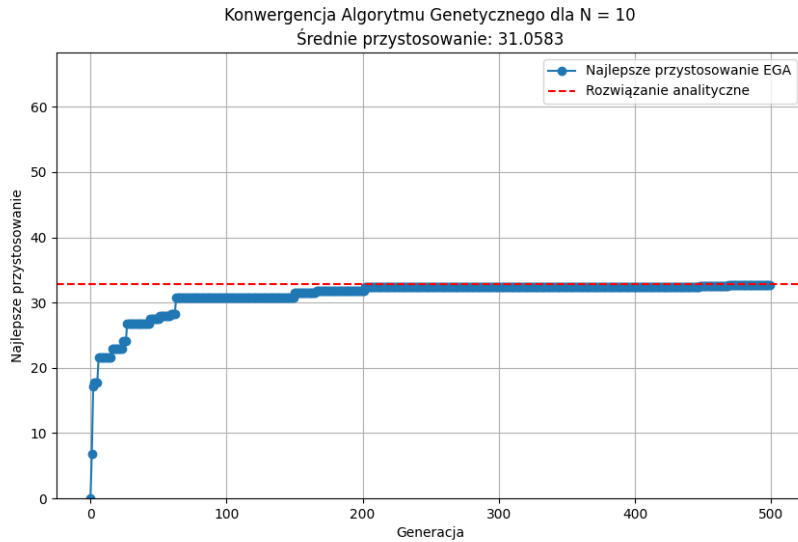


Figure 3: Konwergencja algorytmu dla $N = 10$

Dla największej wartości $N = 45$ konwergencja jest najwolniejsza. Algorytm osiąga wartość funkcji celu znacznie niższą niż wartość optymalna. Jest to spowodowane dużą złożonością problemu i koniecznością zastosowania większej liczby generacji lub zaawansowanych technik optymalizacji.

7.2 Porównanie stanów x_k uzyskanych przez EGA i rozwiązanie analityczne

Wykresy przedstawiające stany x_k w kolejnych okresach pozwalają na wizualne porównanie rozwiązań uzyskanych przez algorytm EGA z rozwiązaniem analitycznym.

Dla $N = 2$ stany uzyskane przez EGA niemal idealnie pokrywają się z rozwiązaniem analitycznym. Świadczy to o wysokiej dokładności algorytmu dla prostych problemów.

W przypadku $N = 4$ również obserwujemy bardzo dobrą zgodność między EGA a rozwiązaniem analitycznym. Niewielkie różnice mogą wynikać z losowości algorytmu genetycznego.

Dla $N = 10$ różnice między stanami są już bardziej widoczne, jednak ogólny trend pozostaje podobny. EGA zbliża się do rozwiązania analitycznego, ale nie osiąga go dokładnie.

Przy $N = 20$ różnice są jeszcze większe. Stany uzyskane przez EGA odchylają się od tych z rozwiąza-

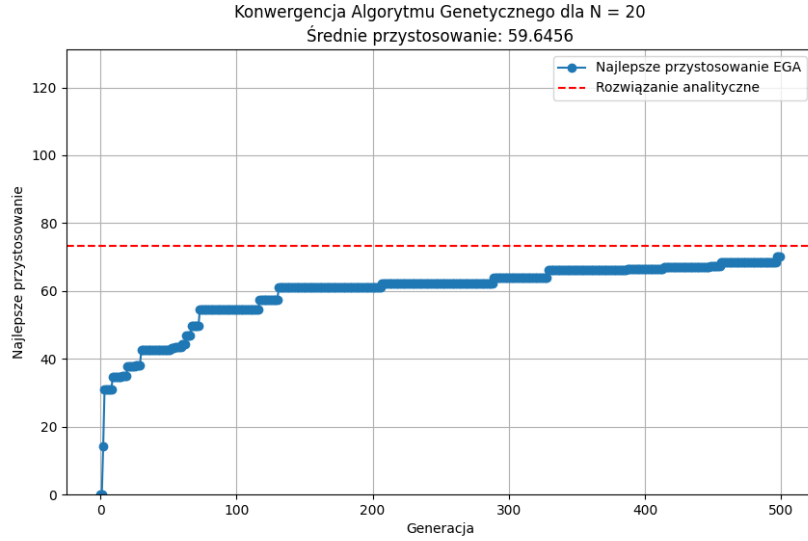


Figure 4: Konwergencja algorytmu dla $N = 20$

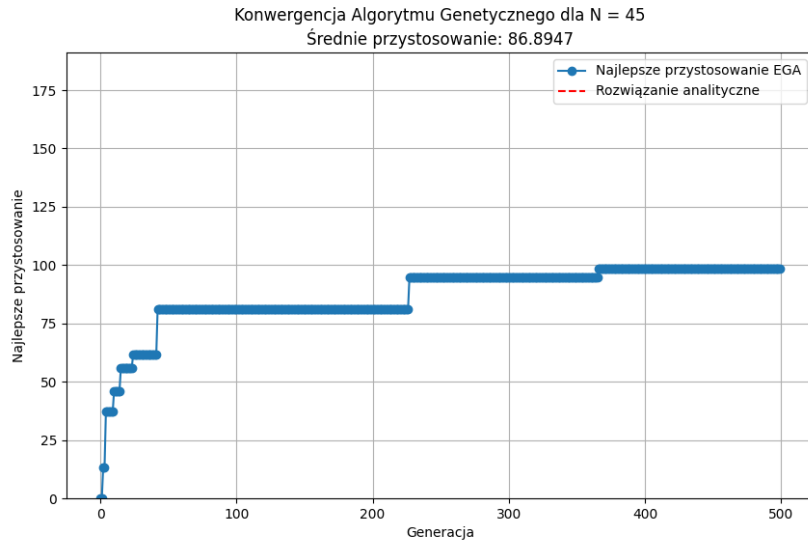


Figure 5: Konwergencja algorytmu dla $N = 45$

nia analitycznego, co sugeruje, że algorytm ma trudności z dokładnym odwzorowaniem optymalnej trajektorii przy większych N .

Dla $N = 45$ stany uzyskane przez EGA znacznie odbiegają od rozwiązania analitycznego. Algorytm nie jest w stanie znaleźć odpowiednio dobrego rozwiązania w zadanej liczbie generacji, co potwierdza potrzebe dalszej optymalizacji parametrów lub zastosowania innych metod.

7.3 Porównanie sterowań u_k uzyskanych przez EGA i rozwiązanie analityczne

Analiza sterowań u_k pozwala zrozumieć, jak algorytm EGA alokuje zasoby w czasie w porównaniu do optymalnego rozwiązania.

Dla $N = 2$ sterowania uzyskane przez EGA są bardzo zbliżone do optymalnych. Algorytm prawidłowo rozkłada zbiór płonów na dwa okresy, maksymalizując funkcje celu.

W przypadku $N = 4$ sterowania EGA również są bliskie optymalnym, choć występują niewielkie różnice w poszczególnych okresach. Algorytm dobrze radzi sobie z alokacją zasobów.

Dla $N = 10$ sterowania uzyskane przez EGA zaczynają odbiegać od rozwiązania analitycznego.

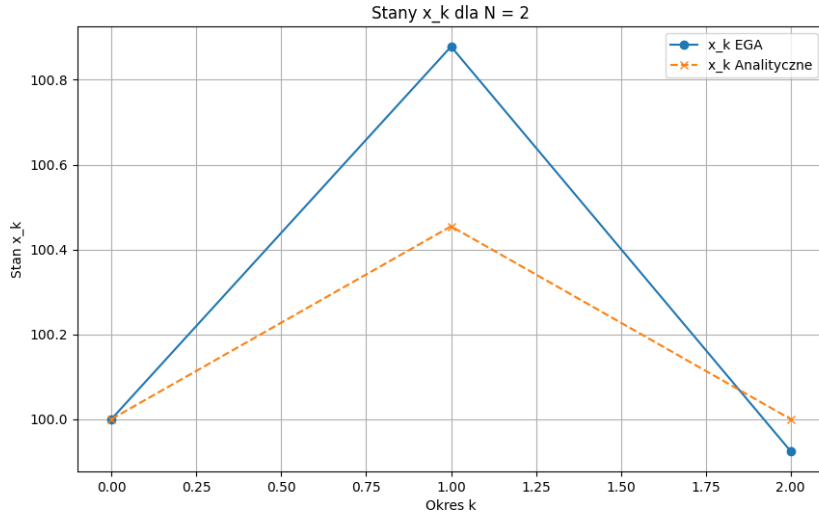


Figure 6: Porównanie stanów x_k dla $N = 2$

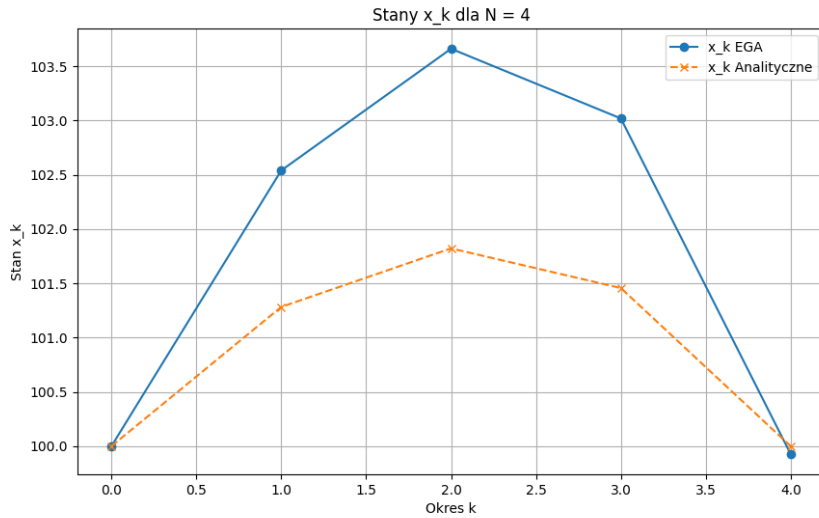


Figure 7: Porównanie stanów x_k dla $N = 4$

Możemy zauważyć większe wahania i mniej regularny rozkład zbiorów.

Przy $N = 20$ różnice są jeszcze bardziej widoczne. Sterowania EGA nie pokrywają się z trendem optymalnym, co wpływa na niższą wartość funkcji celu.

Dla $N = 45$ sterowania uzyskane przez EGA są bardzo nieregularne i znacznie odbiegają od optymalnych. Algorytm ma trudności z wyznaczeniem właściwej strategii zbioru plonów dla tak dużej liczby okresów.

7.4 Porównanie wartości funkcji celu

Wykresy przedstawiające wartości funkcji celu uzyskane przez EGA oraz rozwiązanie analityczne pozwalają na ocenę efektywności algorytmu.

Dla $N = 2$ wartość funkcji celu uzyskana przez EGA niemal pokrywa się z wartością analityczną. Algorytm jest bardzo skuteczny dla prostych problemów.

W przypadku $N = 4$ sytuacja jest podobna. Różnica między wartościami jest minimalna, co potwierdza dobrą jakość rozwiązań EGA.

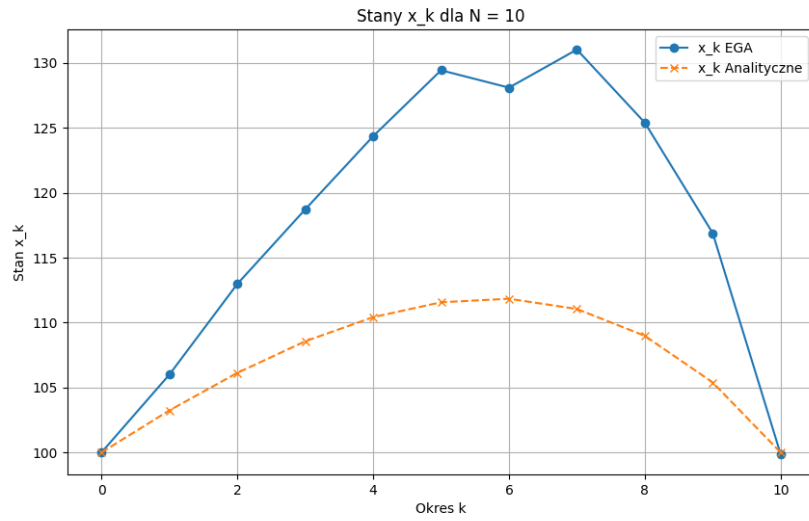


Figure 8: Porównanie stanów x_k dla $N = 10$

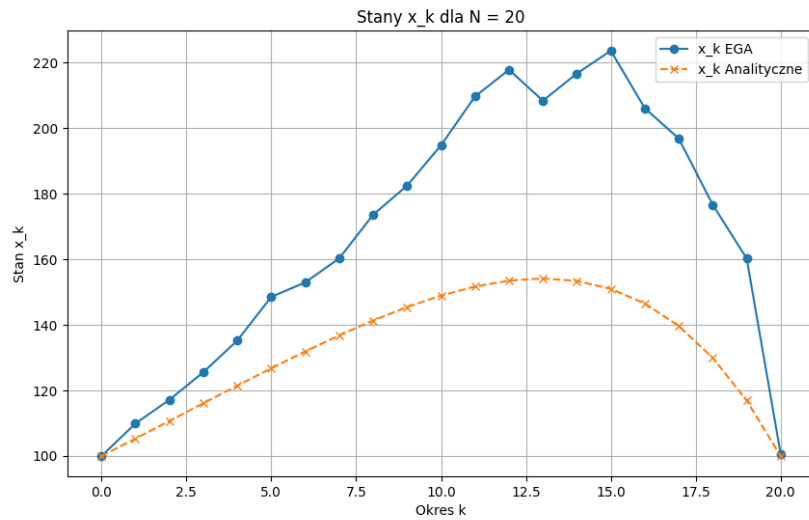


Figure 9: Porównanie stanów x_k dla $N = 20$

Dla $N = 10$ różnica między EGA a rozwiązaniem analitycznym jest większa, ale nadal akceptowalna. Algorytm zbliża się do wartości optymalnej, choć jej nie osiąga.

Przy $N = 20$ różnica jest już wyraźna. Wartość funkcji celu uzyskana przez EGA jest niższa niż wartość analityczna, co wskazuje na potrzebę dalszej optymalizacji algorytmu.

Dla $N = 45$ wartość funkcji celu uzyskana przez EGA jest znacznie niższa od wartości analitycznej. Algorytm nie jest w stanie znaleźć rozwiązania bliskiego optymalnemu w zadanej liczbie generacji.

7.5 Wnioski z analizy wykresów

Analiza wykresów pozwala na wyciągnięcie następujących wniosków:

- Algorytm EGA jest skuteczny dla małych i średnich wartości N , gdzie potrafi znaleźć rozwiązania bliskie optymalnym.
- Dla większych wartości N algorytm napotyka trudności i nie osiąga wartości funkcji celu zbliżonych do optymalnych, co wynika z większej złożoności problemu.

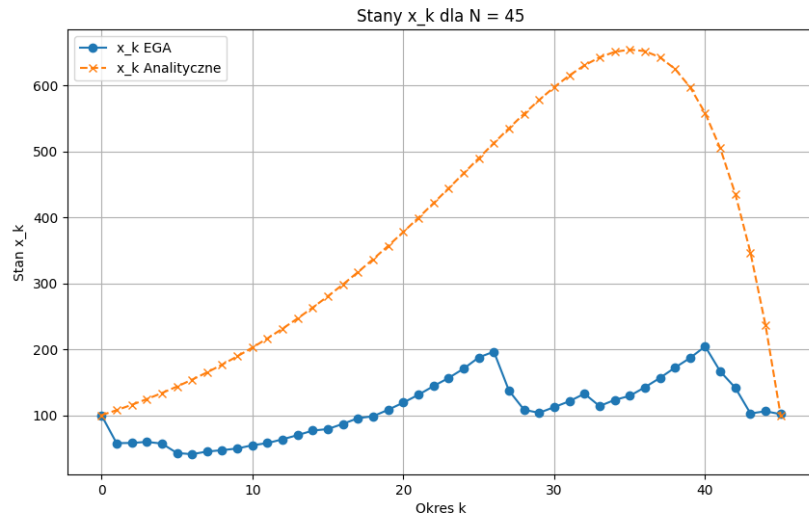


Figure 10: Porównanie stanów x_k dla $N = 45$

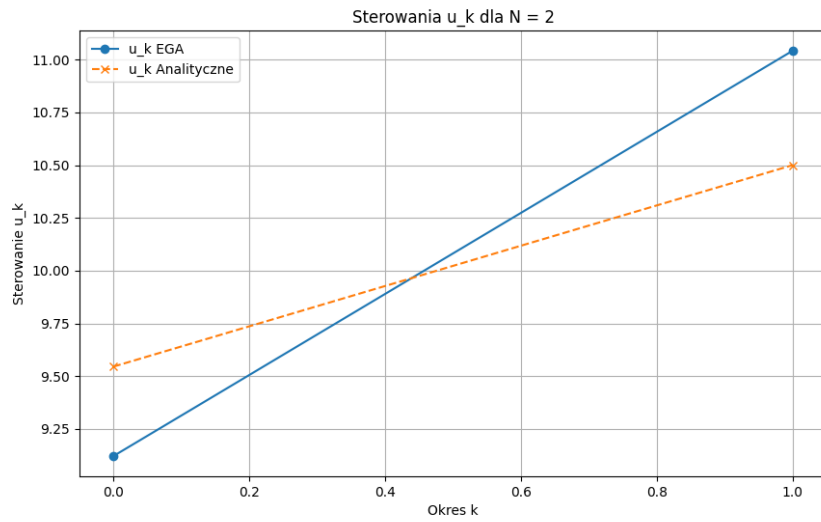


Figure 11: Porównanie sterowań u_k dla $N = 2$

- Konwergencja algorytmu jest wolniejsza dla większych N , co sugeruje potrzebę zwiększenia liczby generacji, dostrojenia parametrów lub zastosowania bardziej zaawansowanych technik optymalizacji.
- Porównanie sterowań i stanów wskazuje, że EGA ma trudności z odwzorowaniem optymalnej strategii zbioru pól przy dużej liczbie okresów.

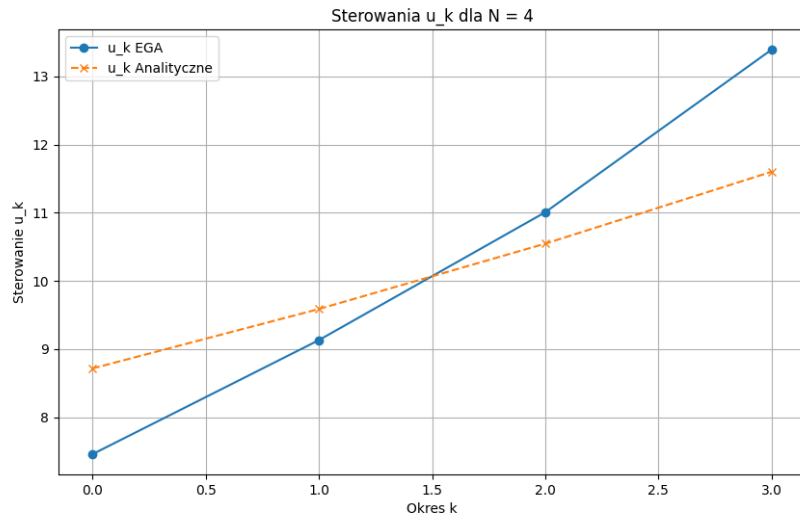


Figure 12: Porównanie sterowań u_k dla $N = 4$

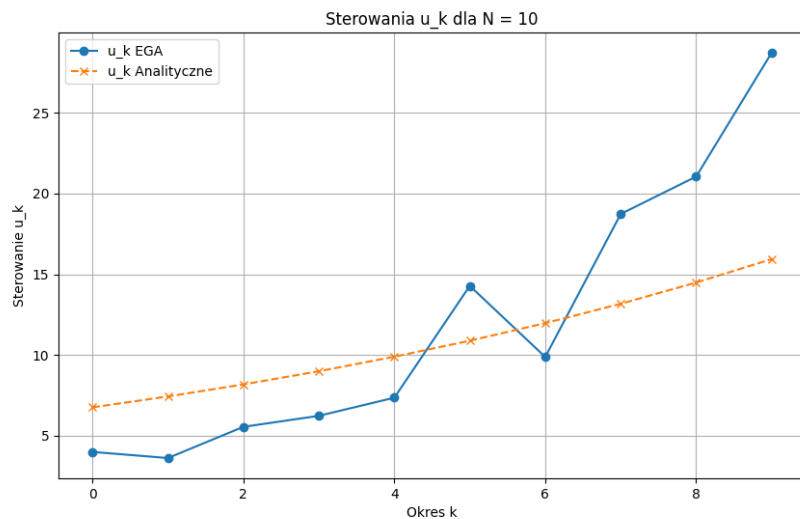


Figure 13: Porównanie sterowań u_k dla $N = 10$

Podsumowanie Sekcji

Wykresy i wizualizacje dostarczają cennych informacji na temat działania algorytmu EGA. Pokazują, że algorytm dobrze radzi sobie z prostszymi problemami, jednak dla bardziej złożonych wymaga dodatkowych usprawnień. Analiza ta pozwala na identyfikację obszarów wymagających optymalizacji i stanowi podstawę do dalszych badań nad poprawą efektywności algorytmu.

8 Podsumowanie

Algorytm EGA okazał się skuteczny w rozwiązywaniu problemu zbioru plonów dla małych i średnich wartości N . Dla większych N konieczne może być zwiększenie liczby generacji, dostrojenie parametrów lub zastosowanie dodatkowych technik optymalizacji, aby osiągnąć wyniki bliższe rozwiązaniu analitycznemu. Zastosowanie zaawansowanego krzyżowania BLX-Alpha oraz dużej populacji pozwoliło na efektywną eksplorację przestrzeni rozwiązań, co jest kluczowe w algorytmach genetycznych dla problemów z ograniczeniami.

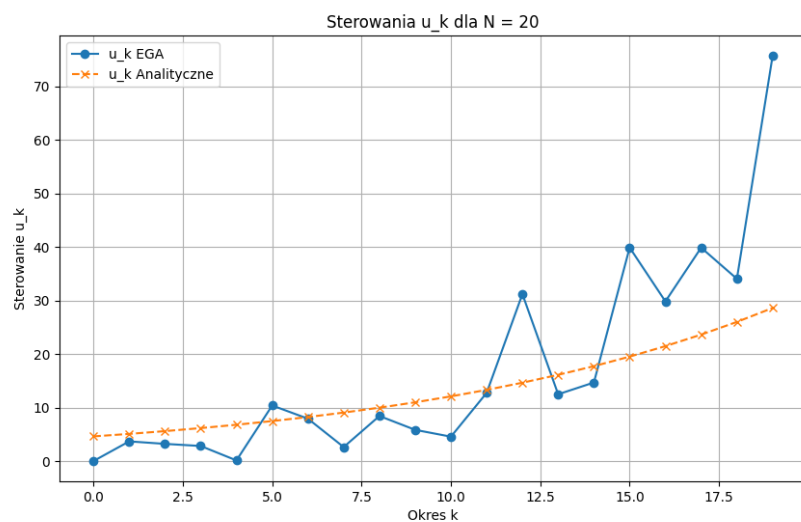


Figure 14: Porównanie sterowań u_k dla $N = 20$

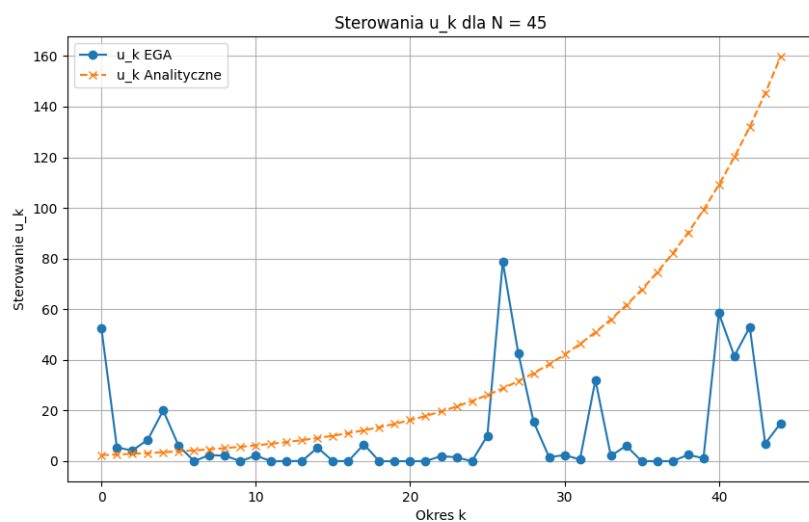


Figure 15: Porównanie sterowań u_k dla $N = 45$

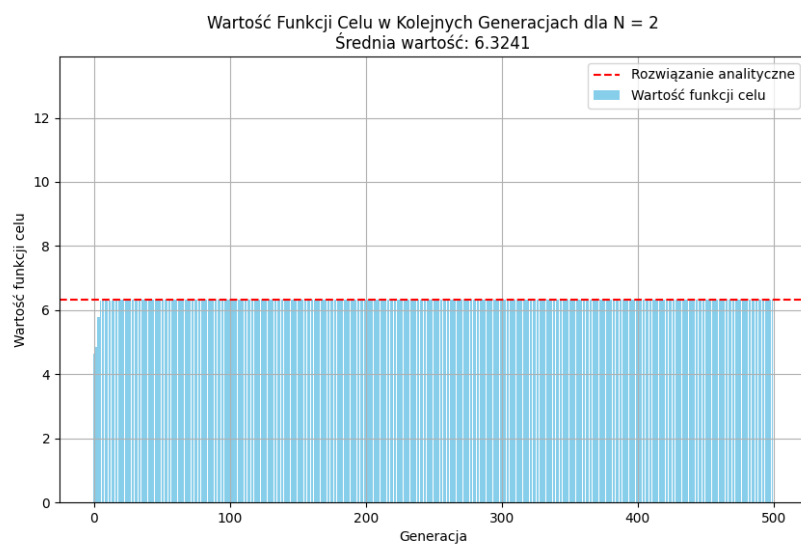


Figure 16: Wartości funkcji celu dla $N = 2$

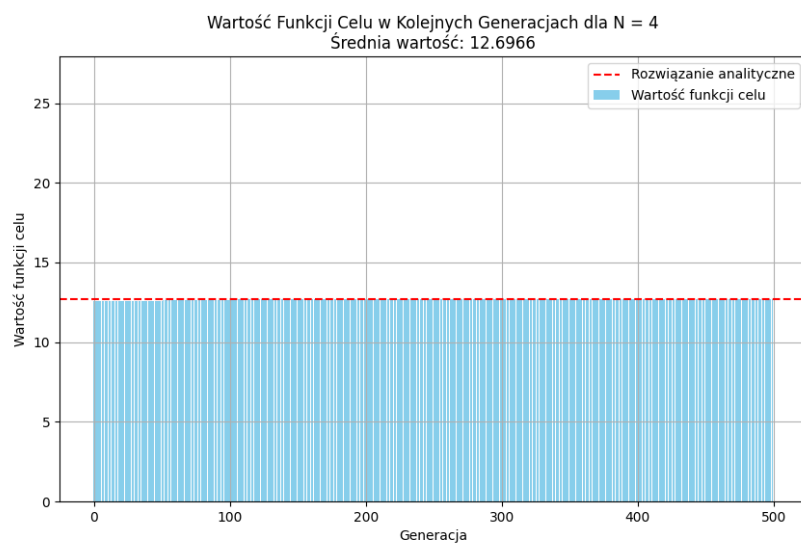


Figure 17: Wartości funkcji celu dla $N = 4$

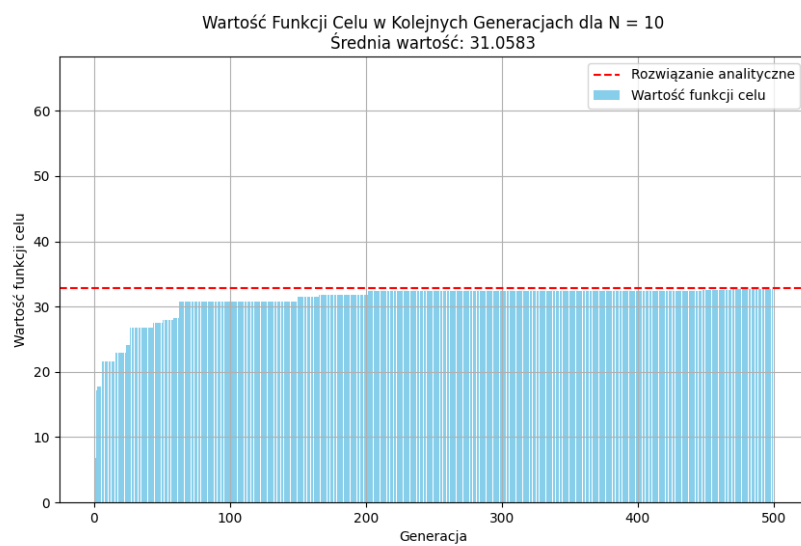


Figure 18: Wartości funkcji celu dla $N = 10$

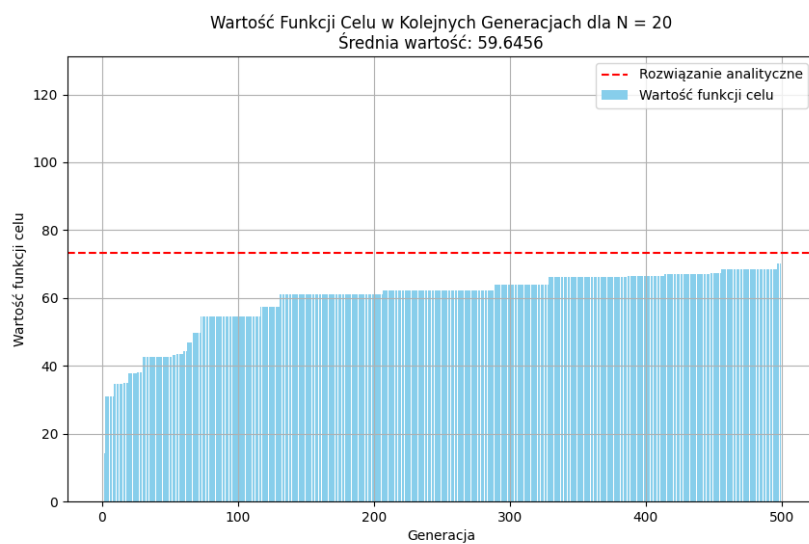


Figure 19: Wartości funkcji celu dla $N = 20$

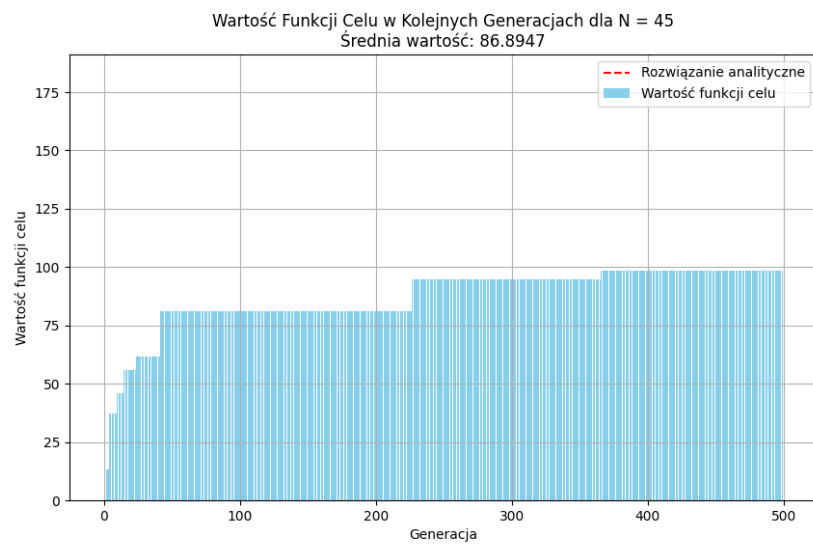


Figure 20: Wartości funkcji celu dla $N = 45$