Algorytmy Ewolucyjne

Jan Pogłód

May 2024

1 Wstęp

Algorytmy ewolucyjne i genetyczne to zaawansowane techniki optymalizacyjne inspirowane procesami ewolucji biologicznej. Celem tych algorytmów jest znalezienie rozwiązań problemów optymalizacyjnych poprzez symulację mechanizmów selekcji naturalnej, krzyżowania i mutacji. W naszej pracy skupimy się na implementacji algorytmów genetycznych do problemu szuakania minimum funkcji oraz problemu "Cutting Stock".

2 Algorytmy genetyczne

Algorytmy ewolucyjne (AE) stanowią ogólną klasę metod optymalizacyjnych, które opierają się na zasadach ewolucji Darwina. Kluczowe elementy AE to tworzenie populacji kandydatów na rozwiązania, mechanizm selekcji wybierający najlepsze jednostki, procesy krzyżowania i mutacji generujące nowe rozwiązania oraz strategia zastępowania słabszych osobników nowymi. Algorytmy te znalazły zastosowanie w wielu dziedzinach, od optymalizacji funkcji matematycznych po złożone problemy inżynieryjne i naukowe. Algorytmy genetyczne (AG) są specyficznym przypadkiem algorytmów ewolucyjnych, które także wykorzystują operatory genetyczne takie jak krzyżowanie i mutacje, lecz ich geny przechowywane są w postaci binarnej. AG i AE modelują proces reprodukcji genetycznej, tworząc nowe populacje na podstawie istniejących, co umożliwia eksplorację przestrzeni rozwiązań.

3 Szukanie ekstremów funkcji

Sprawdzimy czy algorytmy genetyczne są skutecznym narzędziem do szukania minimów globalnych funkcji w przestrzeniach R^2 i R^3. Początkowo musimy zdefinować pojęcia w naszej implementacji, które są niezbędne w każdym algorytmie genetycznym.

- **Osobnik** jest możliwością rozwiązania problemu, podlega ewoulucji, najlepszy z nich będzie potencjalnym rozwiązaniem, a więc punktem w przestrzeni będącym minimum funkcji.
- **Populacja** jest zbiorem osobników o określonej liczności (na przykład 100, 1000). Inicjalizację populacji można przeprowadzić w sposób losowy, na przykład z rozkładu normalnego.
- **Genotyp**: W kontekście algorytmów genetycznych, genotyp reprezentuje całkowity zestaw informacji genetycznej osobnika.

- **Chromosom** - przechowuje informacje wewnątrz genotypu o danym rozwiązaniu.
- **Gen** to pojedyńcza jednostka informacji na chromosomie, która odpowiada za konkretną cechę. W naszym przypadku genem będzie współrzędna w przestrzeni $\mathbf{R^2}$ (np. x, y) lub $\mathbf{R^3}$ (np. x, y, z).
- **Fenotyp** to rzeczywista reprezentacja genotypu, czyli punkt w przestrzeni $\mathbf{R^2}$ lub $\mathbf{R^3}$ określony przez wartości współrzędnych.
- **Funkcja celu** w naszym problemie to funkcja, której minimum szukamy. Przykładowo funkcja f(x,y) w $\mathbf{R^2}$ lub f(x,y,z) w $\mathbf{R^3}$. Natomiast **Funkcja przystosowania** ocenia jakość przystosowania każdego osobnika, a więc ocenę, czy jest to minimum funkcji.

W algorytmie ewolucyjnym symuulujemy ewolucję poprzez modyfikacje wybranych osobników w każdej generacji. W każdej z generacji osobniki potrafią się krzyżować, mutować a na koniec obliczać swoją funkcję przystosowania i na tej podstawie metoda selekcji może przepisywać najlepsze chromosomy - rozwiązania do kolejnej generacji. Oto w jaki sposób będziemy korzystać z powyższych metod w problemie optymalizacji:

- Metody **selekcji** to np. selekcja ruletkowa, gdzie prawdopodobieństwo wyboru danego chromosomu jest proporcjonalne do jego przystosowania. My w naszym algorytmie dla każdego rodzica wybierać będziemy lepszego, pod względem funkcji przystosowania, osobnika z dwóch losowo wybranych dostępnych w obecnej populacji.
- **Krzyżowanie** polega na wymianie fragmentów chromosomów między rodzicami, co pozwala na tworzenie nowych kombinacji genów. W naszym problemie wykorzystamy kryżowanie jednopuntkowe, a więc wybierające część informacji od jednego rodzica i część od drugiego na podstawie dokładnie jednego wylosowanego punktu krzyżowania. Krzyżowanie będziemy przeprowadzać na podstawie prawdopodobieństwa 70%. Wówczas na przykładzie przestrzeni R^3 losować będzimy punkt przecięcia we współrzędnych (x, y, z) i tak wybierzemy na przykład punkty (x_1, y_1, z_2) od rodziców 1. i 2, co będzie ich zmutowanym dzieckiem - nowym osobnikiem.
- **Mutacja** wprowadza losowe zmiany do chromosomów potomstwa, co pomaga utrzymać różnorodność genetyczną w populacji i uniknąć zbieżności do lokalnego minimum. W implementacji zastosujemy mutacje Gaussowską z prawdopodobieństwem zajścia 20%, a więc dla tych osobników dodawać będziemy losową wartość z rokładu normalnego $\epsilon \sim N(0, 1)$, co da nam $(x + \epsilon, y + \epsilon, z + \epsilon)$.

Ostatecznie taką procedurę przeprowadzać będziemy dla każdego osobnika w populacji powtarzając algorytm dla liczby generacji równej n. Każdy osobnik w bieżącej generacji jest wówczas rodzicem, który na podstawie krzyżowania z innym rodzicem będzie w stanie stworzyć nowego osobnika rozszerzając w ten sposób przestrzeń poszukiwań, przy założeniu, że populacja będzie cały czas stała (nie powiększa się ani nie zmniejsza się liczba osobników). Mutacje dadzą nam możliwość wyjścia z ewentualnego utknięcia w minimum lokalnym.

4 Wizualizacja wyników dla problemu szukania minimum globalnego

Jako pierwsze testowaliśmy zachowanie algorytmu ewolucyjnego dla funkcji $f : R^3 \to R$ postaci $f(x, y, z) = x^2 + y^2 + 2z^2$. Przy 1000 generacji i liczbie populacji 100 dostaliśmy, że minimum znajduje się w punkcie $(0.0495, 0.0239, 0.1718)$. Natomiast dla 10000 generacji otrzymaliśmy wynik bardzo bliski zeru: $(-3.389 * 10^{-2}, -3.999 * 10^{-2}, -5.800 * 10^{-6})$

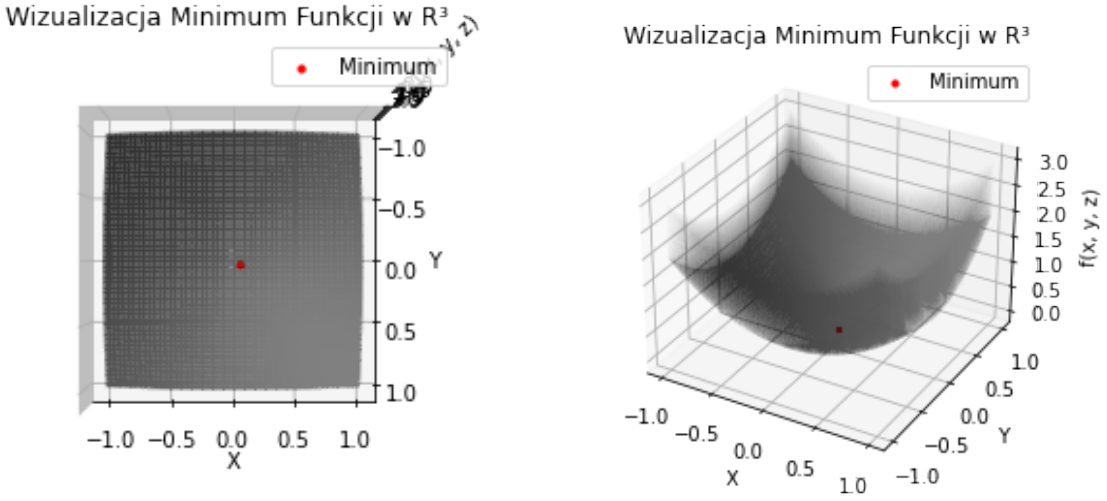

Figure 1: Minimum funkcji kwadratowej znalezione za pomocą algorytmu ewolucyjnego

Następnie testowaliśmy funkcję Rastrigana w n-wymiarowej przestrzeni daną wzorem.

$$f(\mathbf{x}) = A \cdot n + \sum_{i=1}^{n} \left[x_i^2 - A\cos(2\pi x_i)\right]$$

Przeprowadzając testy na przestrzeni 3-wymiarowej i 5-wymiarowej dostaliśmy następujące wyniki:

	n = 3 | 100 generacji	n = 5 | 100 generacji	n = 5 | 1000 generacji
x_1	-0.001	-0.050	-3.634 * 10^{-4}
x_2	0.008	-1.060	-4.989 * 10^{-4}
x_3	-0.001	0.051	7.362 * 10^{-4}
x_4		0.053	9.052 * 10^{-4}
x_5		0.967	1.184 * 10^{-4}

Table 1: Najlepiej przystosowany osobnik z podziałem na każdy z punktów w przestrzeni

Możemy zauważyć, że w problemie szukania minimum funkcji, dla dowolnej liczby wymiarów, większa liczba generacji wpływa korzystniej na końcowy wynik, ponieważ przeszukujemy wówczas większą przestrzeń, na co wpływają m.in. metody krzyżowania i mutacji. Jest to szczególnie

widoczne w przypadku funkcji Rastrigana, która posiada wiele minimum lokalnych. Jedyne 100 generacji w przypadku 5-wymiarowej funkcji okazało się być zbyt małe do uniknięcia minimum lokalnego.

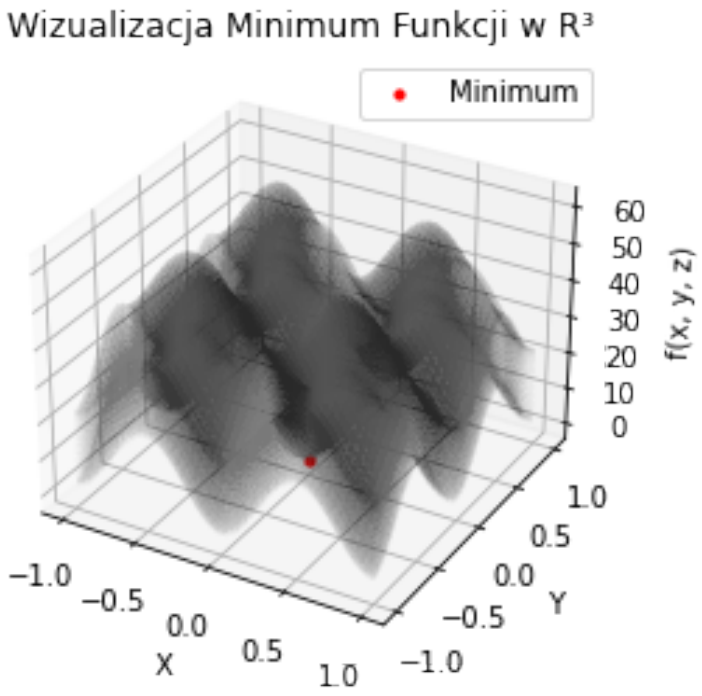

Figure 2: Minimum 3-wymiarowej funkcji Rastrigana znalezione za pomocą algorytmu ewolucyjnego

5 Cutting stock problem

Cutting Stock problem jest zagadnienem optymalizacyjnym dotyczącym maksymalnego wykorzystania powierzchni materiału, aby wyciąć z niego jak najwięcej określonych kształtów. W naszej implementacji skupimy się na problemie wycinania prostokątów o określonych wymiarach z koła o podanym promieniu, przy czym każdy prostokąt posiadać będzie swoją wartość. Naszym zadaniem będzie osiągnąć jak największą wartość sumaryczną wszystkich prostokątów, przy warunkach, aby żadne prostokąty na siebie nie nachodziły oraz żeby wszystkie prostokąty znajdowały się w obszarze koła. Rozwiązanie problemu "cutting stock" z użyciem algorytmów genetycznych wymaga odpowiedniego przedstawienia osobników, operatorów mutacji i krzyżowania.

- **Osobnik** Każdego osobnika kodować będzimy poprzez listę prostokątów, gdzie każdy prostokąt jest reprezentowany przez jego wymiary (wysokość i szerokość), wartość oraz pozycję w kole (x, y). Przyjmujemy, że każdy prostokąt może być umieszczony w dowolnym miejscu w obrębie koła o promieniu r, nie nachodząc przy tym na inne. W ten sposób zbudujemy

strukturę chromosomu - Każdy chromosom w populacji będzie listą prostokątów, gdzie każdy prostokąt ma następujące atrybuty:

  - height - wysokość prostokąta
  - width - szerokość prostokąta
  - value - wartość prostokąta
  - x - współrzędna x lewego górnego rogu prostokąta
  - y - współrzędna y lewego górnego rogu prostokąta

- **Ocena (Funkcja celu)** Funkcja celu będzie sumować wartości prostokątów, które mieszczą się w kole i nie nachodzą na siebie.
- **Mutacja** Operator mutacji będzie polegać na przesunięciu losowego prostokąta w nowe miejsce w obrębie koła lub zmianie jego wymiarów na losowe wymiary z rozkładu normalnego będące również w obrębie koła. Mutację będziemy również przeprowadzać z prawdopodobieństwem 20%.
- **Krzyżowanie** Krzyżowanie z prawdopodobieństwem 70% wśród dwójki rodziców będzie polegać na wymianie części prostokątów - podobnie jak w problemie szukania ekstremów, również zastosujemy wymianę jednopunktową, dzieląc przestrzeń znalezionych prostokątów na dwie części. Wówczas dziecko rodziców otrzymywać będzie część ustawienia prostokątów od jednego jak i drugiego rodzica.
- **Selekcja** Do selekcji wybierać będziemy losowo z połowy najlepszych osobników dla pierwszego i analogicznie drugiego rodzica. Dzięki temu upewniać się będziemy, że za każdym razem wybieramy dokładnie 50% najlepszych rozwiązań. Ten algorytm przeprowadzimy dla każdego osobnika w populacji i podobnie jak wcześniej powtarzać będziemy dla ustalonej liczby generacji.

Początkowo funkcja inicjalizuje populację losowych osobników, umieszczając prostokąty w losowych miejscach w obrębie koła. Wprowadzimy klasę Rectangle reprezentującą prostokąt z określoną wysokością, szerokością i wartością. Dodatkowo za pomocą odpowiedniej funkcji "isWithinCircle" sprawdzać będziemy, czy prostokąt mieści się w kole o zadanym promieniu oraz poprzez funkcję pomocniczą "rectanglesDoNotOverlap" sprawdzać będziemy, czy dwa prostokąty nie nachodzą na siebie. W każdej generacji obliczać będziemy wartość przystosowania dla danego osobnika, sumując wartości prostokątów, które spełniają warunki problemu. Algorytm ewolucyjny powtarzamy iterując przez określoną liczbę pokoleń, stosując selekcję, krzyżowanie i mutację. W ten sposób algorytm będzie starał się umieścić prostokąty w kole, maksymalizując ich sumaryczną wartość. Zaprezentowaliśmy wyniki poszczególnych parametrów takich jak promień koła, wymiary prostokątów, liczba generacji oraz ich wyniki - funkcję przystosowania najlepszego osobnika. Na ostatnich stronach zamieściliśmy wizualizacje, gdzie możemy obserwować najlepsze osobniki wybrane przez algorytm.

6 Podsumowanie

Algorytmy ewolucyjne są bardzo skutecznymi i ciekawymi pod względem inspiracji biologicznych metodami optymalizacyjnymi. Widzimy, że potrafią bardzo skutecznie przemierzać przestrzeń rozwiązań i za pomocą funkcji przystosowania wybierać najlepszego osobnika, który najlepiej spełnia zadanie. Wymagają one jedynie poprawnie skonstuowanych sposobów reprezentowania populacji, mutacji, krzyżowania, selekcji i potrafią być wówczas bardzo skuteczne. Uważam, że ich szerokie zastosowania zdecydowanie przyspieszą rozwój tych algorytmów na przestrzeni kolejnych lat.

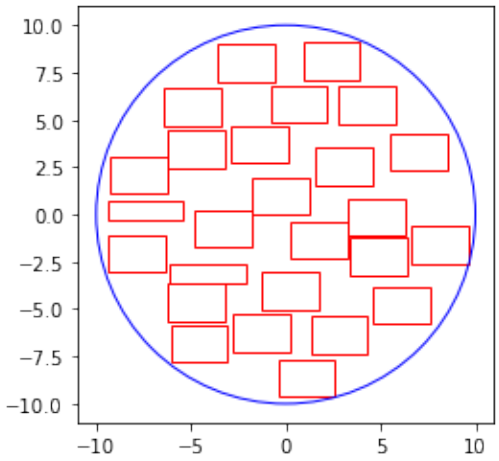

Figure 3: Rozwiązanie dla r=5 i 3 różne prostokąty. 100 generacji.

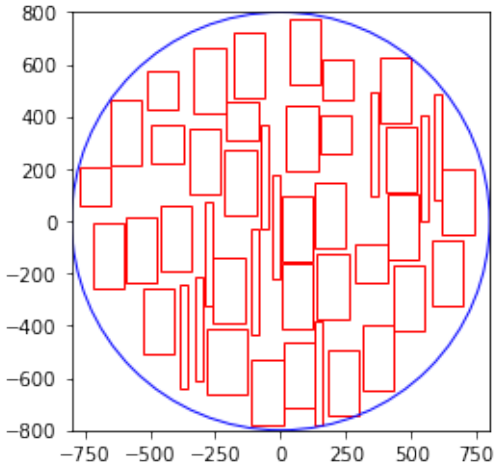

Figure 4: Rozwiązanie dla r=800. 100 generacji. **Wynik: 7480**

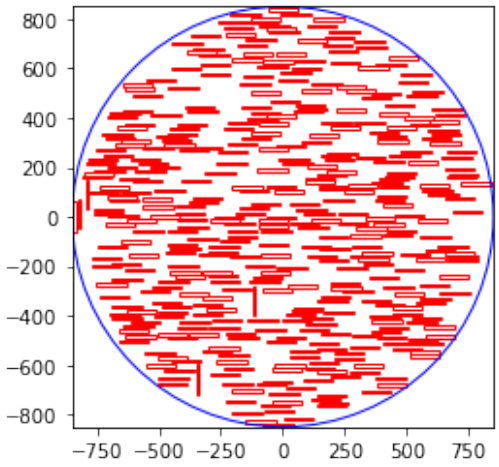

Figure 5: Rozwiązanie dla r=850. 100 generacji. **Wynik: 50190**

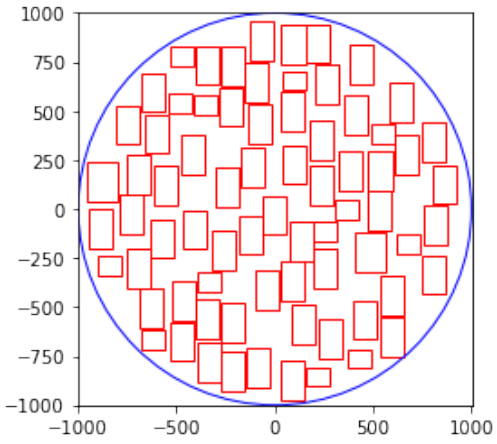

Figure 6: Rozwiązanie dla r=1000. 50 generacji. **Wynik: 12540**

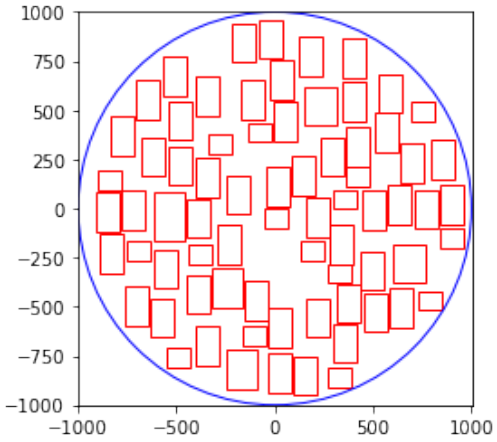

Figure 7: Rozwiązanie dla r=1000. 100 generacji. **Wynik: 10370**

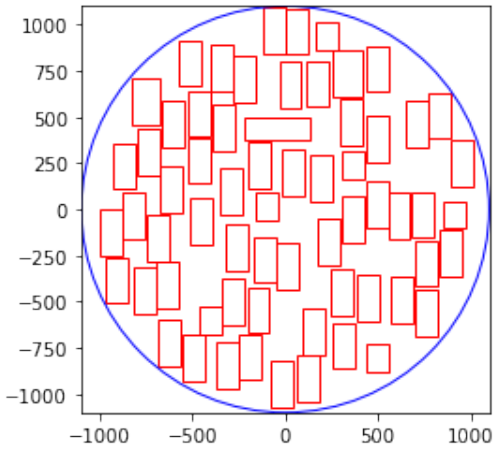

Figure 8: Rozwiązanie dla r=1100. 100 generacji. **Wynik: 7240**

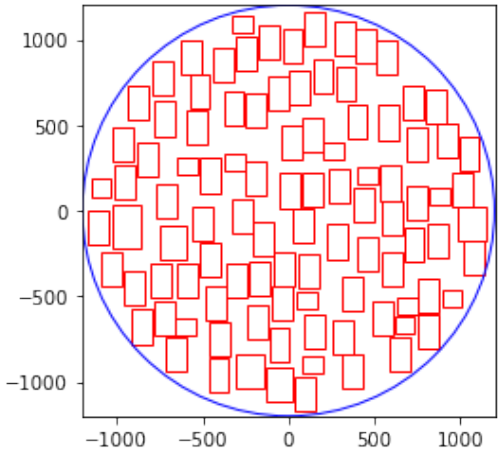

Figure 9: Rozwiązanie dla r=1200. 100 generacji. **Wynik: 18420**

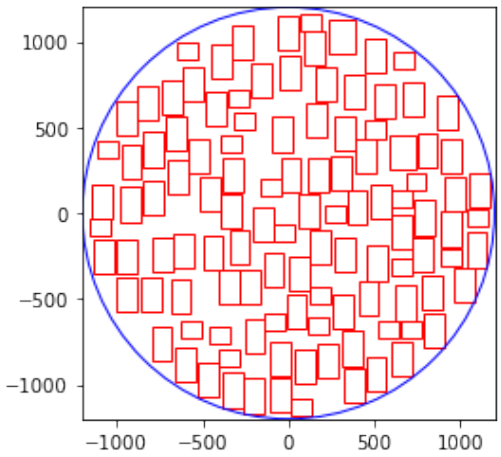

Figure 10: Rozwiązanie dla r=1200. 500 generacji. **Wynik: 18640**