Zadanie 2. **Dokumentacja dla algorytmu ewolucyjnego**Artem Kukushkin,317140

**1. Cel projektu**

Celem projektu jest zaprojektowanie i zaimplementowanie algorytmu ewolucyjnego, który będzie w stanie zoptymalizować funkcje celu z benchmarku CEC2017 (F3 i F19) przy wymiarowości n = 10. Algorytm ewolucyjny zostanie porównany pod względem zbieżności z implementacją algorytmu gradientu prostego, a wyniki będą uśredniane z wielu uruchomień.

**2. Opis algorytmu ewolucyjnego**

Algorytm ewolucyjny jest techniką optymalizacji inspirowaną procesami naturalnymi, takimi jak selekcja, krzyżowanie i mutacja. Celem jest znalezienie najlepszego rozwiązania problemu optymalizacji w danym środowisku.

**Kroki algorytmu ewolucyjnego:**

1. **Inicjalizacja populacji**: Tworzymy początkową populację rozwiązań (wektory liczb) w określonym zakresie.
2. **Ocena dopasowania**: Każde rozwiązanie w populacji jest oceniane na podstawie wartości funkcji celu.
3. **Selekcja**: Populacja jest selekcjonowana metodą turniejową. Wybierani są najlepsi osobnicy do dalszej reprodukcji.
4. **Krzyżowanie**: Z wybranych rodziców tworzymy nowe rozwiązania (potomków) przez krzyżowanie (np. krzyżowanie jednopunktowe).
5. **Mutacja**: Wprowadzamy losowe zmiany w niektórych rozwiązaniach, aby wprowadzić różnorodność do populacji.
6. **Zakończenie**: Algorytm kończy się po określonej liczbie pokoleń. Najlepsze rozwiązanie zostaje wybrane jako wynik.

**3. Szczegóły implementacji**

* **Funkcje pomocnicze**:
  + inicjalizuj\_populacje: Generuje początkową populację.
  + ocena\_dopasowania: Ocena każdej jednostki w populacji na podstawie funkcji celu.
  + selekcja\_turniejowa: Selekcja osobników z populacji do dalszej reprodukcji.
  + krzyzowanie\_jednopunktowe: Operacja krzyżowania dwóch rodziców.
  + mutacja: Mutacja losowa, która wprowadza zmiany do rozwiązania.
* **Algorytm główny**: Funkcja algorytm\_ewolucyjny implementuje cały algorytm ewolucyjny, w tym inicjalizację populacji, selekcję, krzyżowanie, mutację oraz aktualizację populacji na przestrzeni wielu pokoleń.
* **Parametry algorytmu**:
  + rozmiar\_populacji: Liczba jednostek w populacji.
  + wymiar: Wymiar przestrzeni poszukiwań (n = 10).
  + zakres: Zakres dla zmiennych (np. (-100, 100)).
  + liczba\_pokolen: Liczba pokoleń w algorytmie.
  + rozmiar\_turnieju: Liczba uczestników w selekcji turniejowej.
  + prawdopodobieństwo\_mutacji: Prawdopodobieństwo mutacji dla każdego rozwiązania.

**4. Porównanie algorytmu ewolucyjnego z gradientem prostym**

Zbadano zbieżność algorytmu ewolucyjnego na dwóch funkcjach celu (F3 i F19 z benchmarku CEC2017) oraz porównano go z prostym algorytmem gradientu.

**Wyniki:**

* **Funkcja F3**:

Średnia wartość funkcji celu: 33226.54

* **Funkcja F19**:

Średnia wartość funkcji celu: 24811.16

**5. Parametry eksperymentu**

* Wymiar przestrzeni: n = 10
* Zakres dla zmiennych: (-100, 100)
* Liczba powtórzeń: 10
* Liczba pokoleń: 200
* Rozmiar populacji: 50
* Rozmiar turnieju: 3
* Prawdopodobieństwo mutacji: 0.1

**6.Wnioski**

Na podstawie przeprowadzonych eksperymentów algorytm ewolucyjny skutecznie optymalizuje funkcje F3 oraz F19 w przestrzeni wymiarowej n = 10. Wyniki są stabilne i uśrednione z wielu uruchomień, co pozwala uzyskać rzetelną miarę zbieżności algorytmu.