**Specyfikacja Wymagań**

1. Opis Projektu

1.1. Cel Projektu

Stworzenie aplikacji symulującej sieć sensorową, która maksymalizuje czas życia sieci poprzez efektywne harmonogramowanie aktywności sensorów, wykorzystując algorytmy heurystyczne i metaheurystyczne (np. algorytmy genetyczne).

1.2. Zakres Projektu

Aplikacja z prostym interfejsem graficznym (GUI), umożliwiająca definiowanie parametrów symulacji, monitorowanie stanu sensorów i analizowanie wyników.

2. Wymagania Funkcjonalne

2.1. Interfejs Graficzny (GUI)

- Obszar wizualizacji siatki z sensorami i punktami monitorowania.

- Panel kontrolny z możliwością ustawienia parametrów symulacji:

- Liczba sensorów.

- Liczba punktów do monitorowania.

- Poziom pokrycia.

- Rozmiar obszaru.

- Zasięg sensora.

- Przycisk do uruchomienia symulacji.

- Etykieta wyświetlająca wynik symulacji.

2.2. Funkcjonalność Symulacyjna

- Losowe rozrzucenie sensorów i punktów monitorowania na siatce.

- Iteracyjny proces optymalizacji:

- Minimalizowanie funkcji celu poprzez zmiany stanu (np. aktywacja/dezaktywacja sensorów).

- Aktualizacja harmonogramu na podstawie wyników iteracji.

2.3. Monitorowanie i Wizualizacja Wyników

- Wyświetlanie bieżącego stanu pokrycia punktów monitorowania w czasie.

- Liczba aktywnych sensorów w danym momencie.

- Wykresy dotyczące liczby aktywnych sensorów i pokrycia punktów monitorowania w czasie.

2.4. Dokumentacja

- Opis rozwiązania problemu optymalizacji czasu życia sieci.

- Wyniki symulacji dla różnych konfiguracji sieci.

- Wykresy i analizy porównawcze.

3. Wymagania Niefunkcjonalne

- Intuicyjny i prosty w obsłudze interfejs graficzny.

- Efektywny algorytm optymalizacyjny, który może obsługiwać duże liczby sensorów i punktów do monitorowania.

4. Wymagania Techniczne

- Język Programowania: Python

- Biblioteki: Tkinter (GUI), Matplotlib, NumPy

- Systemy operacyjne: Windows

5. Scenariusze Użytkownika

5.1. Uruchomienie Symulacji

1. Użytkownik otwiera aplikację.

2. Ustawia parametry wejściowe.

3. Uruchamia symulację.

4. Obserwuje wizualizację i monitoruje wyniki na bieżąco.

6. Przykład Wersji Zbioru Inputów dla Algorytmu

1. Liczba sensorów: 50

2. Liczba punktów do monitorowania: 20

3. Poziom pokrycia: 75%

4. Rozmiar obszaru: 300x300 jednostek

5. Zasięg detekcji sensorów: 30 jednostek

7. Harmonogramowanie Algorytmu

1. Inicjalizacja

- Losowe rozrzucenie sensorów i punktów monitorowania na siatce.

- Ustawienie początkowych parametrów zasięgu dla każdego sensora.

2. Symulacja

- Iteracyjny proces optymalizacji.

- Minimalizowanie funkcji celu poprzez zmiany stanu (np. aktywacja/dezaktywacja sensorów).

- Aktualizacja harmonogramu na podstawie wyników iteracji.

3. Generowanie Wyników

- Wyświetlanie czasu życia sieci dla różnych parametrów.

- Generowanie wykresów i analiz.

8. Wnioski

Program umożliwia łatwe monitorowanie i optymalizację sieci sensorowej, z możliwością dostosowywania parametrów oraz analizą wyników za pomocą wykresów i statystyk. Efektywne algorytmy optymalizacyjne zapewniają maksymalny czas życia sieci sensorowej w różnych warunkach operacyjnych.