

Dokumentacja Użytkownika - Symulator Sieci Sensorowej

Niniejsza dokumentacja użytkownika ma na celu zapoznanie z podstawowymi funkcjami aplikacji Symulatora Sieci Sensorowej oraz przedstawienie sposobu jej obsługi.

Jak korzystać z aplikacji?

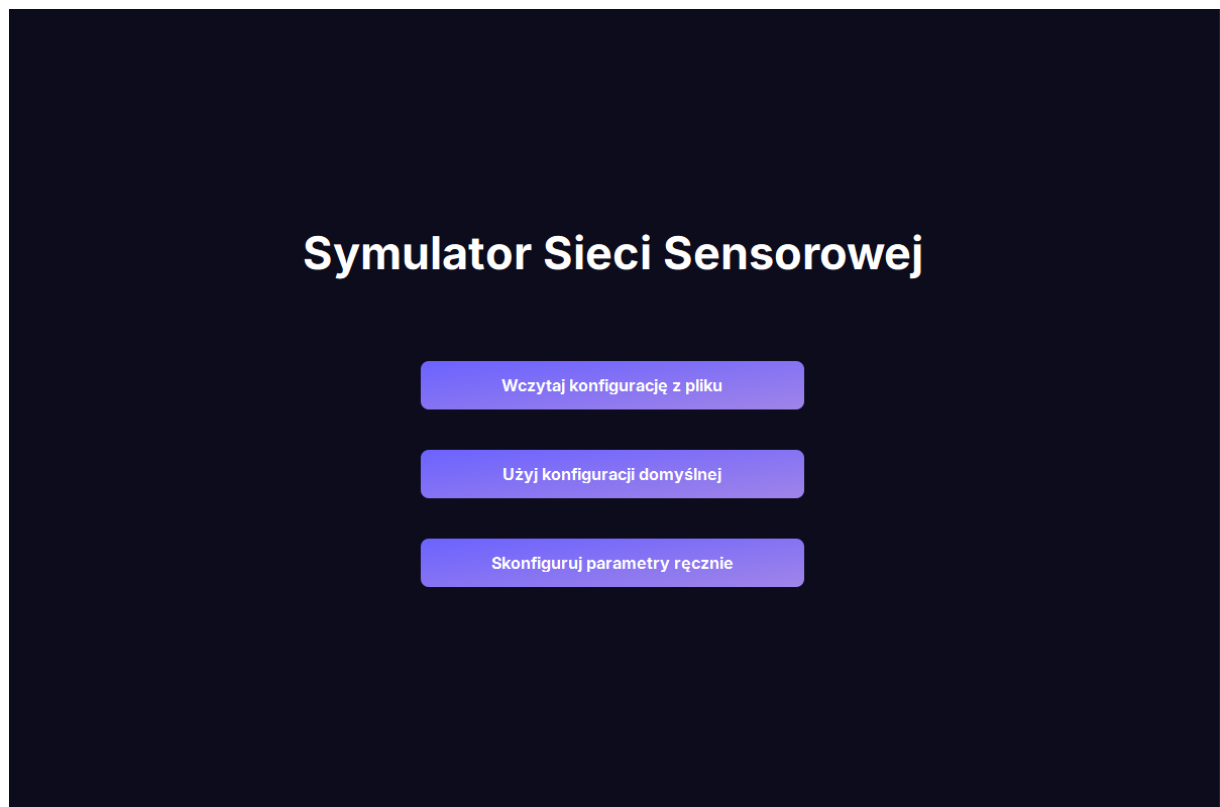
Aplikacja Symulatora Sieci Sensorowej umożliwia modelowanie i analizę działania bezprzewodowych sieci sensorowych. Poniżej przedstawiono kroki korzystania z aplikacji:

1. Uruchomienie Aplikacji:

- Aplikacja uruchamiana jest poprzez wykonanie głównego pliku main.py.
- Po uruchomieniu pojawi się okno startowe aplikacji.

2. Konfiguracja Symulacji:

- Na stronie startowej użytkownik ma do wyboru trzy opcje konfiguracji:
 - **Wczytaj konfigurację z pliku:** Pozwala na załadowanie wcześniej przygotowanego pliku konfiguracyjnego (w formacie .txt lub .ini). Umożliwia to szybkie wczytanie złożonych scenariuszy symulacji.
 - **Użyj konfiguracji domyślnej:** Ładuje predefiniowany zestaw parametrów symulacji, który znajduje się w pliku config/default_simulation_config.txt.
 - **Skonfiguruj parametry ręcznie:** Przenosi do strony, gdzie użytkownik może interaktywnie modyfikować wszystkie parametry symulacji poprzez interfejs graficzny zorganizowany w zakładkach.

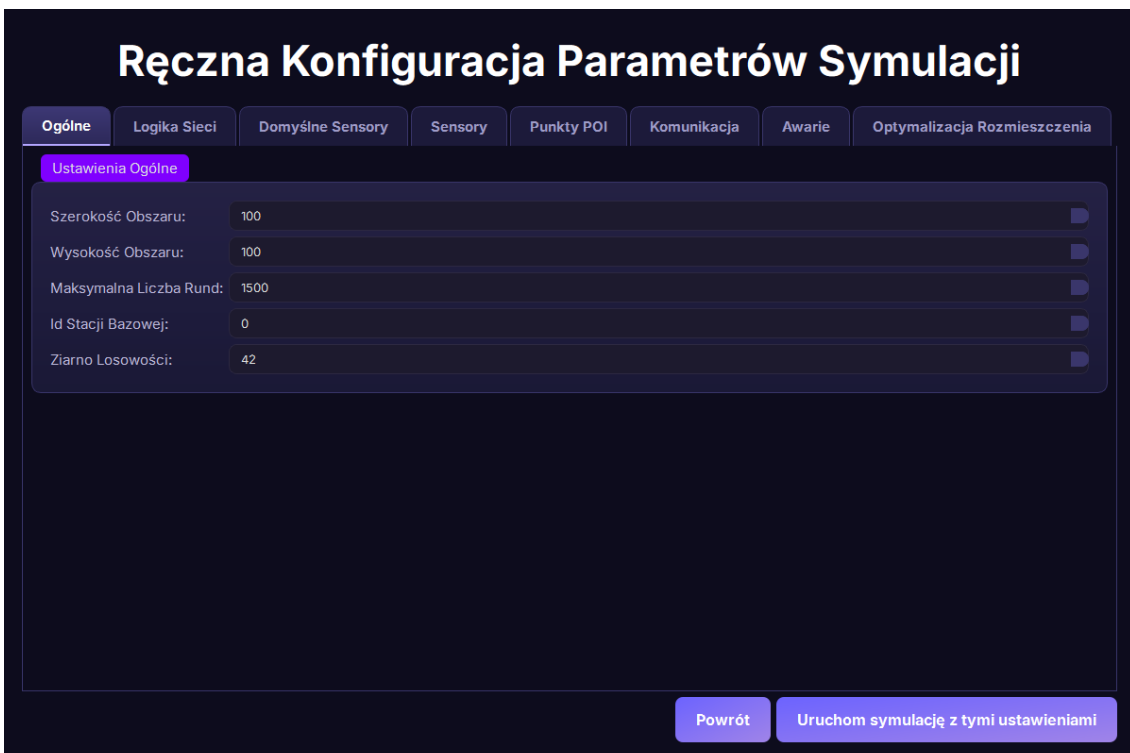


3. Modyfikacja Parametrów (jeśli wybrano wczytanie z pliku lub ręczną konfigurację):

- **Edytor Pliku Konfiguracyjnego:** Jeśli wybrano wczytanie z pliku, jego zawartość pojawi się w edytorze tekstowym. Użytkownik może bezpośrednio modyfikować parametry w edytorze, zapisać zmiany lub zapisać jako nowy plik.



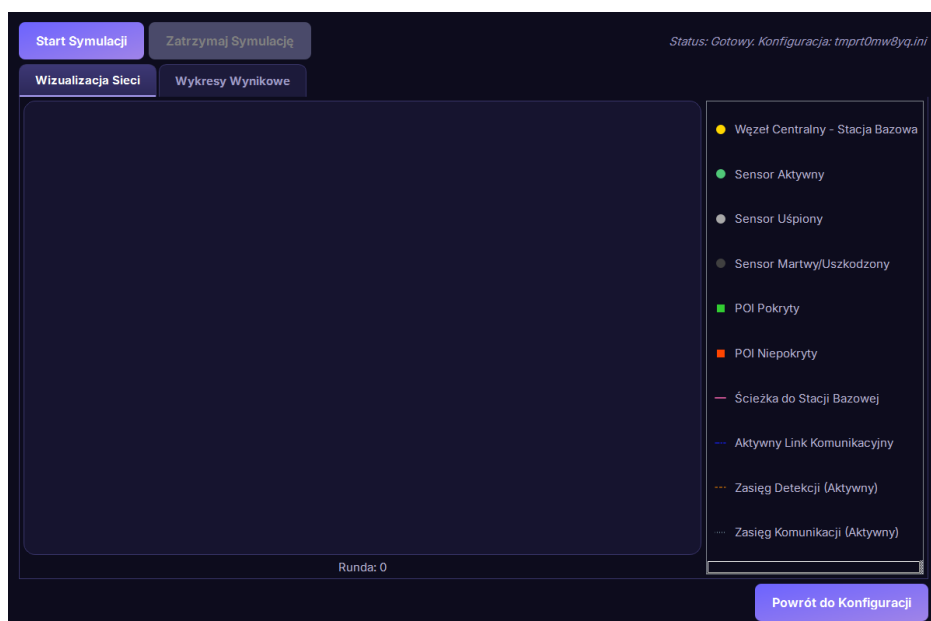
- **Ręczna Konfiguracja Parametrów:** Jeśli wybrano konfigurację ręczną, użytkownik ma dostęp do szeregu zakładek, w których może modyfikować poszczególne aspekty symulacji:



- **Ogólne:** Szerokość i wysokość obszaru, maksymalna liczba rund, ID stacji bazowej, ziarno losowości.
- **Logika Sieci:** Metoda nagradzania, czas pracy zbioru pokrycia, warunek zakończenia symulacji.
- **Domyślne Sensory:** Początkowa energia, zasięg komunikacji, zasięg sensora, parametr 'a' dla automatów uczących.
- **Sensory:** Możliwość ustawienia ogólnej liczby sensorów. Dokładne pozycje i indywidualne parametry sensorów (nadpisujące domyślne) konfiguruje się poprzez wczytanie pliku konfiguracyjnego.
- **Punkty POI:** Możliwość ustawienia ogólnej liczby Punktów Zainteresowania (POI). Dokładne pozycje POI konfiguruje się poprzez wczytanie pliku konfiguracyjnego.
- **Komunikacja:** Prawdopodobieństwo utraty pakietu, opóźnienie transmisji na skok, maksymalny rozmiar kolejki, interwał broadcastu informacji o POI.
- **Awarie:** Współczynnik awarii sensora na rundę.
- **Optymalizacja Rozmieszczenia:** Umożliwia włączenie i konfigurację parametrów Algorytmu Genetycznego (GA) do optymalizacji rozmieszczenia sensorów (rozmiar populacji, liczba generacji, współczynniki mutacji i krzyżowania itp.).

4. Uruchomienie Symulacji:

- Po skonfigurowaniu parametrów (poprzez edytor pliku lub stronę ręcznej konfiguracji), należy nacisnąć przycisk "Uruchom symulację z tym plikiem" (na stronie edytora) lub "Uruchom symulację z tymi ustawieniami" (na stronie ręcznej konfiguracji).
- Aplikacja przejdzie do strony symulacji



5. Przebieg Symulacji:

- Na stronie symulacji, po kliknięciu przycisku "Start Symulacji", rozpocznie się symulacja.
- **Wizualizacja Sieci:** Domyślnie wyświetlana jest wizualizacja sieci sensorowej w czasie rzeczywistym.
 - Pokazuje pozycje sensorów i POI, ich stany (aktywny, uśpiony, martwy/uszkodzony), zasięgi (detekcji i komunikacji, jeśli włączone w konfiguracji), ścieżki do stacji bazowej oraz aktywne linki komunikacyjne.
 - Legenda po prawej stronie wyjaśnia znaczenie poszczególnych elementów.
 - Wyświetlany jest aktualny numer rundy.



- **Kontrola Symulacji:**
 - **Start Symulacji:** Rozpoczyna lub wznawia symulację.
 - **Zatrzymaj Symulację:** Zatrzymuje bieżącą symulację.
 - **Powrót do Konfiguracji:** Umożliwia powrót do strony startowej w celu zmiany konfiguracji po zakończeniu lub zatrzymaniu symulacji.
- **Status:** Wyświetla aktualny status symulacji (np. "Oczekuje na konfigurację", "Symulacja w toku...", "Symulacja zakończona").
- **Postęp:** W trakcie działania symulacji wyświetlane jest okno dialogowe z postępem, pokazujące przetwarzaną rundę i maksymalną liczbę rund.

6. Wyniki Symulacji:

- Po zakończeniu symulacji (lub jej zatrzymaniu), wyniki są dostępne w zakładce "Wykresy Wynikowe".
- Prezentowane są wykresy ilustrujące kluczowe metryki, takie jak:
 - Stany Sensorów (liczba sensorów aktywnych, uśpionych, martwych/uszkodzonych w czasie).
 - Średnia Energia (średni poziom energii żywych sensorów).
 - Pokrycie POI (Q) (wskaźnik Q-pokrycia sieci).
 - PDR (Packet Delivery Ratio).
 - Średnie Opóźnienie (średnie opóźnienie pakietów dostarczonych do stacji bazowej).



- Dodatkowo, szczegółowy log symulacji, zawierający m.in. statystyki z każdej rundy, zapisywany jest do pliku tekstowego (domyślnie results/simulation_log.txt).

Opis kluczowych funkcjonalności

Aplikacja symuluje działanie bezprzewodowej sieci sensorowej, koncentrując się na **wykorzystaniu automatów uczących (Learning Automata) do maksymalizacji czasu życia sieci** poprzez inteligentne zarządzanie aktywnością sensorów. Główne funkcjonalności obejmują:

- **Symulator Sieci Sensorowej zorientowany na Czas Życia Sieci:**
 - **Centralny Mechanizm Oparty na Automatach Uczących (Learning Automata):**
 - Każdy sensor wyposażony jest w automat uczący, który dynamicznie podejmuje decyzje o przejściu w stan aktywny (monitorowanie i komunikacja) lub uśpiony.
 - Decyzje te są podejmowane na podstawie mechanizmu nagród i kar, który odzwierciedla wkład sensora w pokrycie Punktów Zainteresowania (POI) oraz efektywność energetyczną.
 - Celem automatów uczących jest wypracowanie takiej strategii działania sensorów, która zapewnia wymagane pokrycie POI przy jednoczesnym minimalizowaniu zużycia energii, co bezpośrednio przekłada się na maksymalizację czasu życia całej sieci. W pliku `simulation_core/sensor.py` znajduje się implementacja klasy `LearningAutomata`, a logika jej wykorzystania w sieci (np. do wyboru zbiorów pokrywających) jest zaimplementowana w `simulation_core/network.py`.
 - **Model Komunikacji:**
 - Umożliwia sensorom wykrywanie POI w ich zasięgu.
 - Zaimplementowano mechanizmy przekazywania informacji o wykrytych POI do sensora centralnego (stacji bazowej). Odbywa się to poprzez wieloskokowy routing.
 - Sensory komunikują się z sąsiadami w celu ustalenia ścieżek do stacji bazowej oraz potencjalnie w celu koordynacji (choć główna koordynacja wynika z działania automatów uczących i decyzji podejmowanych przez obiekt `Network`). Pliki `simulation_core/communication_model.py` oraz `simulation_core/routing.py` odpowiadają za te aspekty.
 - **Model Energii:**
 - Każdy sensor posiada monitorowany poziom baterii.
 - Zużycie energii jest zróżnicowane w zależności od stanu sensora:
 - **Sen:** minimalne zużycie energii.
 - **Monitoring/Aktywność:** zwiększone zużycie energii związane z działaniem modułu sensorycznego i przetwarzaniem.
 - **Komunikacja (transmisja/odbior):** największe zużycie energii.

- Ten model (zdefiniowany w `simulation_core/energy_model.py`) jest kluczowy dla oceny efektywności strategii zarządzania energią i maksymalizacji czasu życia sieci.
- **Algorytmy Optymalizacyjne:**
 - **Główny Algorytm Optymalizacyjny (Maksymalizacja Czasu Życia Sieci):** Realizowany poprzez wspomniane wcześniej automaty uczące (LA) wbudowane w sensory. Decydują one o aktywności sensorów w celu zapewnienia pokrycia POI przy minimalnym zużyciu energii. Aplikacja implementuje mechanizm wyboru zbioru pokrycia (Cover Set - CS), który aktywuje minimalną liczbę sensorów niezbędną do pokrycia wszystkich krytycznych POI, a także sensorów mostowych (Bridge Sensors) zapewniających łączność ze stacją bazową. Ten proces jest sterowany przez logikę w `simulation_core/network.py` i wykorzystuje LA.
 - **Optymalizacja Rozmieszczenia Sensorów (GA):** Opcjonalnie, przed rozpoczęciem głównej symulacji, możliwe jest uruchomienie Algorytmu Genetycznego (GA) w celu znalezienia optymalnego początkowego rozmieszczenia sensorów. Celem GA jest maksymalizacja pokrycia POI i zapewnienie łączności ze stacją bazową. Konfiguracja GA (np. rozmiar populacji, liczba generacji) jest dostępna w interfejsie użytkownika lub pliku konfiguracyjnym. Implementacja znajduje się w `simulation_core/deployment_optimizer.py`.
- **Obsługa Awarii i Problemów Komunikacyjnych:**
 - Symulowana jest możliwość losowej **awarii sensora** w każdej rundzie symulacji (parametr `failure_rate` w konfiguracji). Awaryjny sensor staje się nieaktywny (martwy).
 - Model komunikacji uwzględnia możliwość **zgubienia pakietu** podczas transmisji (parametr `packet_loss_probability`).
 - Występują także opóźnienia w komunikacji.
- **Interfejs Użytkownika (GUI):**
 - **Elastyczna Konfiguracja Danych Wejściowych:** Umożliwia użytkownikowi modyfikację szerokiego zakresu parametrów symulacji, w tym:
 - Liczba i rozmieszczenie sensorów oraz POI (szczegółowe rozmieszczenie głównie przez plik konfiguracyjny).
 - Początkowy poziom baterii sensorów.
 - Zasięg "widoczności" (sensora) oraz zasięg komunikacji.
 - Parametry modelu komunikacji (np. prawdopodobieństwo utraty pakietu).
 - Wybór i konfiguracja wykorzystywanych algorytmów optymalizacji (np. parametry LA, włączenie/konfiguracja GA dla rozmieszczenia).

- *Zrzut ekranu: Strona ręcznej konfiguracji parametrów z zakładkami "Ogólne", "Logika Sieci", "Domyślne Sensory", "Sensory", "Punkty POI", "Komunikacja", "Awarie", "Optymalizacja Rozmieszczenia".*
- **Wczytywanie Konfiguracji z Pliku:** Aplikacja obsługuje wczytywanie wszystkich parametrów symulacji z odpowiednio sformatowanego pliku tekstowego (format .txt interpretowany jako INI). Pozwala to na łatwe definiowanie i odtwarzanie złożonych scenariuszy.
 - *Zrzut ekranu: Strona "Edytor Pliku Konfiguracyjnego" z załadowanym przykładowym plikiem konfiguracyjnym.*
- **Wizualizacja Działania Sieci w Trakcie Symulacji:**
 - Graficzna prezentacja stanu sieci w czasie rzeczywistym (lub z zadaną prędkością).
 - Zaznaczane są:
 - Sensory: aktywne (np. kolor zielony), uśpione (np. kolor szary), martwe/uszkodzone (np. kolor czarny/czerwony).
 - Zasięg "widoczności" (detekcji) poszczególnych sensorów (jako okręgi).
 - Zasięg komunikacji sensorów (jeśli włączone w opcjach wizualizacji).
 - Wybrane ścieżki komunikacji z węzłem centralnym (stacją bazową).
 - Punkty POI (zaznaczenie, czy są aktualnie pokrywane).



- **Wyniki Symulacji:**
 - **Generowanie Pliku Wynikowego:** Po zakończeniu symulacji (lub w jej trakcie, w zależności od konfiguracji logowania) tworzony jest plik tekstowy (domyślnie results/simulation_log.txt). Zawiera on szczegółowe wyniki, takie jak:
 - Liczba sensorów aktywnych, uśpionych i martwych w każdej rundzie.

- Aktualny poziom baterii poszczególnych sensorów.
 - Liczba oraz lista widocznych sąsiadów dla sensorów.
 - Aktualny poziom pokrycia POI (metryka Q-Coverage).
 - Packet Delivery Ratio (PDR) – stosunek dostarczonych pakietów do wysłanych.
 - Latencja (opóźnienie) pakietów.
 - Inne statystyki istotne dla analizy działania sieci i efektywności algorytmów.
- **Wizualizacja Danych Wyjściowych (Wykresy):** Aplikacja prezentuje zagregowane wyniki w postaci wykresów, które ilustrują zmiany kluczowych parametrów w czasie trwania symulacji. Przykładowe wykresy to:
- Liczba aktywnych/uśpionych/martwych sensorów w funkcji czasu (rund).
 - Średni poziom energii w sieci.
 - Poziom pokrycia Q w funkcji czasu.
 - PDR i średnia latencja w funkcji czasu.



Spis błędów

Podczas korzystania z aplikacji mogą wystąpić następujące komunikaty o błędach:

- **Błąd wczytywania pliku konfiguracyjnego:**
 - "Config file not found: [ścieżka_do_pliku]" - Wyświetlany, gdy podany plik konfiguracyjny nie istnieje.
 - "Nie można otworzyć pliku [ścieżka_do_pliku]: [szczegóły_błędu]" - Wyświetlany w GUI, gdy wystąpi problem z odczytem zawartości pliku.
 - "Nie można utworzyć domyślnego pliku konfiguracyjnego: [szczegóły_błędu]" - Gdy aplikacja nie może automatycznie utworzyć domyślnego pliku konfiguracyjnego.
- **Błąd zapisu pliku konfiguracyjnego:**
 - "Najpierw załaduj lub 'Zapisz jako...' nowy plik." - Gdy użytkownik próbuje zapisać zmiany bez aktywnego pliku.
 - "Nie można zapisać pliku [ścieżka_do_pliku]: [szczegóły_błędu]" - Gdy wystąpi problem podczas zapisu pliku.
- **Błąd przygotowania symulacji:**
 - "Plik konfiguracyjny nie istnieje: [ścieżka_do_pliku_konfiguracyjnego]" - Wyświetlany na stronie symulacji, jeśli przekazany plik konfiguracyjny jest nieprawidłowy.
 - "Nie można odczytać wymiarów sieci z konfiguracji: [szczegóły_błędu]" - Jeśli w pliku konfiguracyjnym brakuje lub są nieprawidłowe dane dotyczące wymiarów obszaru symulacji.
 - "Nie można przygotować pliku konfiguracyjnego: [szczegóły_błędu]" - Ogólny błąd podczas generowania tymczasowego pliku konfiguracyjnego na podstawie ustawień ręcznych.
- **Błąd w trakcie symulacji:**
 - "Błąd Symulacji: [komunikat_błędu]" - Ogólny komunikat o błędzie krytycznym, który przerwał działanie symulacji. Szczegóły błędu mogą być widoczne w konsoli.
 - "SIM CRITICAL ERROR: network.run_one_round() returned None..." - Komunikat w konsoli wskazujący na wewnętrzny błąd logiki sieci, który uniemożliwia kontynuację.
 - "Coverage lost at round X as reported by Network object." - Symulacja może zostać przerwana, jeśli sieć straci zdolność do pokrywania POI (zależnie od konfiguracji warunku zakończenia).
 - "MONITORING CRITICAL - Sensor X still DISCONNECTED..." - Komunikat w logach wskazujący, że sensor należący do zbioru pokrywającego nie mógł uzyskać połączenia ze stacją bazową, co może prowadzić do utraty pokrycia.

Lista nietypowych zachowań systemu

- **Brak aktualizacji wizualizacji, jeśli okno zostało zamknięte:** Jeśli użytkownik zamknie okno wizualizacji Matplotlib w trakcie działania symulacji, wizualizacja przestanie być aktualizowana, a symulacja może zostać zatrzymana (komunikat "Visualization window closed. Stopping simulation." pojawi się w konsoli).
- **Brak wykresów wynikowych przy braku danych:** Jeśli symulacja zostanie zatrzymana bardzo wcześnie (przed zebraniem jakichkolwiek danych statystycznych) lub wystąpi błąd uniemożliwiający zebranie danych, zakładka "Wykresy Wynikowe" może nie wyświetlić żadnych wykresów lub pokazać komunikat o braku danych.
- **Ograniczenie $P(A)$ w Algorytmie CS:** W algorytmie wyboru sensorów do zbioru pokrycia (CS), sensory z prawdopodobieństwem akcji ACTIVE ($P(A)$) poniżej 0.5 są początkowo pomijane. Jeśli nie znajdzie się odpowiednich kandydatów, kryterium $P(A)$ jest luzowane. Może to prowadzić do wyboru mniej "chętnych" sensorów w pewnych scenariuszach.
- **Losowość przy braku zoptymalizowanych pozycji:** Jeśli optymalizacja rozmieszczenia jest wyłączona lub nie uda się znaleźć zoptymalizowanych pozycji dla wszystkich sensorów, brakujące pozycje są losowane. Może to prowadzić do różnych początkowych topologii sieci przy tych samych pozostałych parametrach, jeśli nie ustawiono ziarna losowości.
- **Domyślne wartości dla brakujących parametrów POI/Sensorów w pliku:** Jeśli w pliku konfiguracyjnym brakuje niektórych szczegółowych parametrów dla poszczególnych sensorów lub POI (np. indywidualna energia, poziom krytyczności POI), aplikacja użyje wartości domyślnych z sekcji [SensorDefaults] lub standardowych wartości (np. 1 dla critical_level POI).

Opis niezgodności projektu z założonymi Kryteriami Akceptacji

- **Interfejs użytkownika - "dowolne modyfikowanie zbioru danych wejściowych":** Interfejs graficzny pozwala na modyfikację wielu kluczowych parametrów poprzez stronę ręcznej konfiguracji (podzieloną na zakładki) oraz edytor plików konfiguracyjnych. Jednakże, "dowolne" modyfikowanie każdego możliwego atrybutu każdego indywidualnego sensora lub POI bezpośrednio w GUI (poza ogólną liczbą, jeśli nie ładuje się pliku) może być ograniczone na rzecz edycji pliku konfiguracyjnego dla bardziej szczegółowych scenariuszy.
 - *Uzasadnienie:* Pełna modyfikacja wszystkich parametrów dla dużej liczby sensorów/POI bezpośrednio w GUI byłaby bardzo nieporęczna i trudna w implementacji. GUI oferuje wygodny sposób na modyfikację ogólnych parametrów i edycję samego pliku, a szczegółowa konfiguracja może być wczytywana z pliku.