Uniwersytet Kardynała Stefana Wyszyńskiego w Warszawie

Wydział Matematyczno-Przyrodniczy

Szkoła Nauk Ścisłych

Jakub Koterba

Symulator Bezprzewodowej Sieci Sensorowej:
Wpływ algorytmu PSO na czas życia sieci

Projekt Zespołowy

Praca wykonana pod kierunkiem

Dr. Jakuba Gąsiora

*

Słowa kluczowe

Optymizacja Rojem Cząstek, Wizualizacja, Aplikacja

Klasyfikacja tematyczna

Bezprzewodowe Sieci Sensorowe

English Title

Wireless Sensor Network Simulator:
Influence of Particle-Swarm-Optimisation on lifetime expectancy

*

Streszczenie

Niniejsza praca dotyczy Symulatora sieci sensorowej. We wstępie poruszone zostają kluczowe kwestie dotyczące Bezprzewodowych Sieci Sensorowych oraz ich aspektów, co do których ciągle prowadzone są badania i próby optymalizacji. Dostarczona zostaje tym samym podstawowa wiedza, aby zrozumieć poleconą literaturę. W dalszej części poruszona zostaje tematyka specyfikacji wymagań oraz struktury programu. Omówione są obiekty i założenia, na których zbudowano modele symulacji. Następnie, poruszona jest kwestia badania, które ma na celu udowodnić, że wyniki osiągane w programie mają pokrycie z rzeczywistością. Praca zostaje zakończona omówieniem wyników doświadczenia.

Spis treści

1 Wstęp Teoretyczny

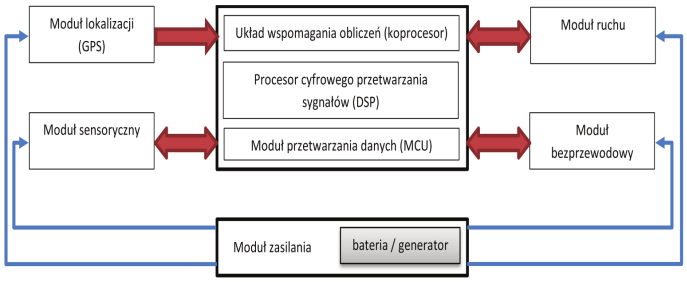
1.1 Sieć Sensoryczna

1.1.1 Informacje ogólne

Oryginalnie, sensory były elektromechanicznymi detektorami, służącymi do pomiaru wielkości fizycznych. Po raz pierwszy, komercyjnie użyto ich w 1933 roku do pomiaru temperatury pomieszczeń [9]. Wczesne mikroelektromechaniczne systemy(MEMS) składały się z kilku osobnych części, na których osobno znajdowały się: sensor, elektronika i mechanika. Taka budowa, czyniła te urządzenia znacznie mniej opłacalnymi w porównaniu do dzisiejszych [10]. Postęp w dziedzinie urządzeń MEMS i układów scalonych umożliwił znaczne zmniejszenie wymaganych komponentów na tyle, że obecnie sensor potrafi być zintegrowany z jednostką obliczeniową, na jednej, małej, płytce drukowanej. Małe wymiary urządzenia niosą za sobą przede wszystkim niższe koszty produkcji, ale także łatwość rozmieszczania. Obecnie sensory mogą być rozmieszczane za pomocą dronów i nawet insektów [11]. Niskie koszty sensorów, łatwość rozmieszczania w przestrzeni i nowoczesne technologie bezprzewodowe powodują, że bezprzewodowa sieć sensoryczna (BSS, ang. WSN) jest dobrym rozwiązaniem do monitorowania wielkości fizycznych, które następnie używane są do przeprowadzenia badań w wielu dziedzinach.

1.1.2 Budowa węzła

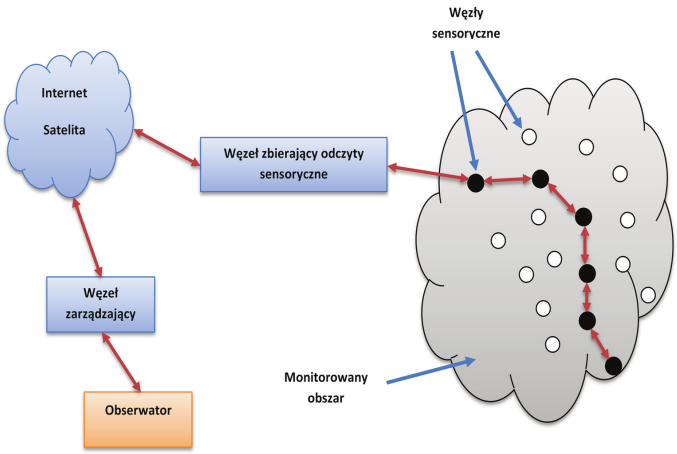
Przeciętny węzeł bezprzewodowej sieci sensorowej składa się z: modułu sensorycznego(odpowiadającego za zbieranie danych), modułu bezprzewodowego(odpowiadającego za komunikację między innymi węzłami sieci), modułu ruchu(nieobowiązkowy, potrzebny, jeżeli węzeł sensoryczny ma poruszać się o własnych siłach), modułu lokalizacji(GPS), układu wspomagania obliczeń, procesora cyfrowego przetwarzania sygnałów, modułu przetwarzania danych oraz modułu zasilania, rys. 1[12], [13].



Rysunek 1: Budowa węzła bezprzewodowej sieci sensorowej[12]

1.1.3 Sposób funkcjonowania sieci

Węzły sensoryczne zazwyczaj są rozmieszczone są w przestrzeni w sposób losowy, rys 2 [12],[14]. Każdy z sensorów może zbierać dane i przekazywać dane. W przypadku dużych sieci, korzystających z klastrowania, mogą istnieć węzły typowo służące do przekazywania danych. Po ponownym ustaleniu zawartości klastrów, taki węzeł może wrócić do roli zbierającego dane.



Rysunek 2: Schemat komunikacji sieci[12], [14]

1.2 Aspekty sieci podlegające optymalizacji

1.2.1 Pokrycie

Sensory podczas inicjalizacji sieci są zazwyczaj zrzucane w dużych ilościach, z pokładów samolotów. Z racji, że obecnie koszt stworzenia funkcjonującej sieci przewyższa wielokrotnie koszt samych sensorów, często zrzuca się ich więcej niż jest to wymagane. Taka sytuacja powoduje, że rozrzucona infrastruktura jest upakowana gęsto i występuje w niej zjawisko pokrywania się rejonów badanych [**?**, i2] Może to przynieść dobre pole do przedłużenia czasu życia sieci, bez tracenia procentażu pokrycia. Gdzie procent pokrycia, to stosunek ilości aktywnych sensorów do wszystkich rozrzuconych sensorów, lub stosunek obszaru pokrywanego zasięgiem wykrywania do całego obszaru, który miał zostać pokryty. W dziedzinie optymalizacji algorytmów trasowania ważniejszym stosunkiem jest ten pierwszy.

1.2.2 Czas życia sieci

W literaturze często pojawia się definicja, mówiąca, że czasem życia sieci jest taki czas który upłynie zanim ilość aktywnych węzłów przekroczy pewien ustalony procent wszystkich węzłów. Czyli czas, kiedy przynajmniej k z n węzłów jest aktywna. Taki model jest używany w większości literatury specjalistycznej.

2 Przeglad Literatury

W literaturze, problematyka bezprzewodowej sieci sensorowej jest rozlegle poruszana od wielu lat. Nie jest to dziwne, ponieważ nadal jest to jedna, z najbardziej efektywnych metod monitorowania dużych areałów. W podanych pracach, najczęściej spotyka się heurystyki i metaheurystyki, które podlegają badaniom, celem udowodnienia sensowności danej implementacji. Bardzo często rozwiązaniem mającym przedłużyć czas życia sieci jest technika zwana klastrowaniem. Czyli dzieleniem całej grupy sensorów na podgrupy. Taki zabieg ma na celu wyrównanie zużycia energii na przestrzeni całej sieci. Taka tematyka poruszana jest na przykład w [1][3][6]

3 Specyfikacja Wymagań

3.1 Wstęp

3.1.1 Cel

Celem tej części dokumentu jest przedstawienie projektu symulatora bezprzewodowej sieci sensorowej, który ma umożliwić wygodne i szybkie emulowanie sieci, bez względu na użyte techiniki trasowania i optymalizacji czasu życia.

3.1.2 Przewidywani adresaci

Osobami, do których skierowany jest ten program mogą być zarówno użytkownicy komercyjni, jak i prywatni oraz programiści. Szczególnie zaleca się użycie go jako pomoc dla osób badających sieci sensorowe, lub studentów

3.1.3 Zastosowanie oprogramowania

Proponowany symulator, to narzędzie, które zostało zaprojektowane tak, aby ułatwić przeprowadzenie skomplikowanych badań dotyczących czasu życia sieci sensorowej. Jego działanie jest intuicyjne i proste. Pozwala na łatwe wybranie wszystkich kluczowych parametrów pracy sieci oraz na zebranie wyników symulacji do wykresu.

3.1.4 Źródła

Repozytorium projektu: https://github.com/M4KIF/Sensoric-Network-Research.git

3.2 Opis

3.2.1 Zamysł produktu

Poprzez interfejs graficzny, mamy wgląd do wyników symulacji przeprowadzonej na uproszczonym modelu sieci sensorowej, który składa się z obiektów zwanych węzłami. W tym modelu, każdy węzeł posiada w sobie centralną jednostkę obliczeniową, jednostkę zarządzania energią i baterię. Każdy parametr przedstawionych elementów jest możliwy do zmodyfikowania. Istnieje też możliwość łatwego zapisania wyników symulacji do wykresu.

3.2.2 Funkcje produktu

Do funkcji tego oprogramowania należą przede wszystkim: - generacja sieci sensorycznej,
- modyfikacja parametrów pracy,
- symulacja pracy sieci sensorycznej,
- wyświetlanie efektów działania na ekranie,
- zapisywanie efektów symulacji do wykresu.

3.3 Użyte techniki

Do użytych technik należą:
- programowanie obiektowe,
- wielowątkowość,
- algorytmy inspirowane naturą.

3.4 Użyte moduły

3.4.1 pyqt

Odpowiada za utworzenie czytelnego interfejsu graficznego do obsługi aplikacji oraz do wprowadzenia obsługi wielowątkowości za pomocą wewnętrznej implementacji, bez użycia składni języka Python.

3.4.2 shapely

To moduł, który ułatwia operacje na punktach, obliczanie zależności między punktami i figurami. Upraszcza obliczenia związane z lokalizacją obiektów programu.

3.4.3 matplotlib

Daje możliwość rysowania wykresów w intuicyjny sposób. W tym przypadku, używany do rysowania położenia indywidualnych węzłów i wyników symulacji.

3.4.4 math

Moduł do szybkich obliczeń matematycznych

3.4.5 scipy

Moduł z którego wzięty jest pseudo-losowy generator oparty na rozkładzie równomiernym

3.5 Środowisko operacji

Program jest możliwy do uruchomienia na każdym systemie operacyjnym wspierającym język Python w wersji min. 3.10, tj.
- Windows,
- MacOS,
- Linux.

4 Struktura programu

4.1 Obiekty

Obiekt zbierający dane z wykonań symulacji sieci. Umożliwia zapisanie zbieranych danych do prostego wykresu.

```
class DataCollector()
```

Obiekt emulujący działanie baterii, przechowuje ładunek oraz umożliwia korzystanie ze swojej energii

```
class Battery()
```

Obiekt emulujący działanie jednostki zarządzania energią urządzenia, komunikuje się z baterią i oblicza wartości zużycia energii

```
class EMU(Battery)
```

Obiekt emulujący działanie centralnej jednostki obliczeniowej w węźle sieci

```
class SOC()
```

Obiekt emulujący działanie pojedyńczego węzła sieci sensorowej, zawiera w sobie obiekty: EMU i SOC.

```
class Node()
```

Obiekt przechowujący wszystkie węzły należące do stworzonej sieci. Odpowiada za przeprowadzanie emulacji działania

```
class SensoricNetwork(QObject)
```

Obiekt emitujący pojedyńczą cząsteczkę, punkt w przestrzeni. Posiadaja wszystkie paramtery określające jego położenie w czasie i przestrzeni, tj. prędkość i położenie. Wykorzystywany do algorytmu optymizacji rojem cząstek

```
class Particle()
```

Obiekt służący do łatwego zarządzania rysowanym wykresem położenia węzłów sieci

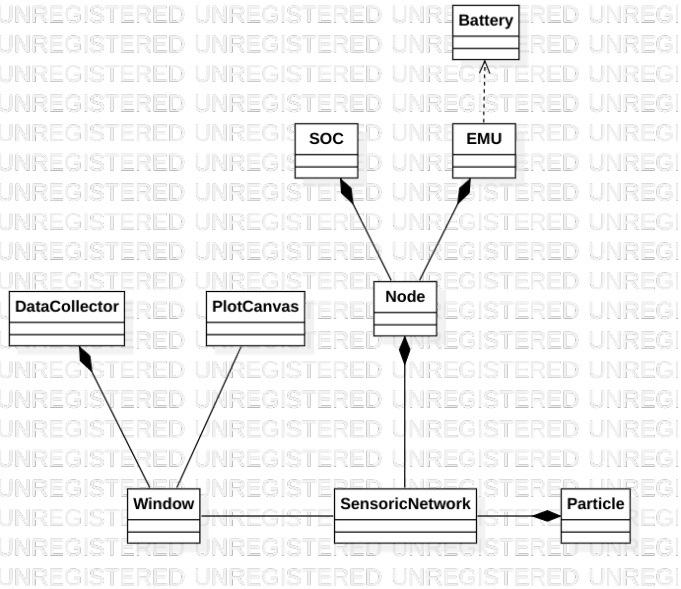
```
class PlotCanvas(FigureCanvasQTAgg)
```

Główny obiekt odpowiedzialny za funkcjonowanie interfejsu graficznego tej aplikacji, z niego przekazywane są polecenia do sieci sensorowej.

```
class Window(QMainWindow)
```

4.2 Diagram Klas

Na poniższym diagramie, rys. 3, znajdują się zależności między obiektami, z których składa się symulator



Rysunek 3: Diagram Klas

4.3 Założenia

4.3.1 Model Sieci

W tym dokumencie, sieć jest formowana przez losowe rozrzucenie n ilości sensorów na powierzchni prostokąta o wymiarach $X * Z$. Pole obszaru działania sieci jest reprezentowane jako A. Założenia są następujące[1]:

- Wszystkie sensory są jednakowe i posiadają taką samą ilość energii początkowej

- Każdy z sensorów jest świadomy swojego położenia w przestrzenie dzięki użyciu systemu GPS

- Wszystkie sensory są nieruchome

- Wszystkie sensory są świadome ilości pozostałej energii w swoim ogniwie i posiadają taki sam zasięg transmisji

- Węzły mogą same dostosowywać ilość pobieranej energii na podstawie drogi transmisji

- Stacja Bazowa sieci jest nieruchoma, oraz posiada nielimitowaną ilość energii

4.3.2 Model Energii

Użyty model energii jest taki sam jak w [15], [16] i [1]. Czyli użycie energii w węzłach zachodzi głównie w obwodach nadajnika, wzmacniacza sygnału i odbiornika. Model obiera poziom niskiego użycia energi i wysokiego, w zależności od dystansu pomiędzy nadajnikiem, a odbiornikiem. Model poboru energii jest proporcjonalny do d^2 jeżeli dystans transmisji$d$ jest mniejszy od dystansu progu d_0, w innym wypadku jest proporcjonalne do d^4. Całkowita energia potrzebna do dostarczenia n-bitowej wiadomości na dystansie d wynosi:

$$E_T(n,d)=\begin{cases} n*E_{nad}+n*\epsilon_{low}*d^2 & dla\ d<d_0 \\ n*E_{nad}+n*\epsilon_{high}*d^4 & dla\ d>d_0 \end{cases}.$$

$GdzieE_{nad}$ to energia potrzebna dla nadajnika, do wysłania 1 bitu danych, a $\epsilon_{low}, \epsilon_{high}$ to dwie wartości, odpowiednio niska i wysoka, poboru mocy wzmacniacza dla 1 wysyłanego bitu. Wartość d_0 jest obliczana za pomocą wzoru:

$$d_0 = \sqrt{\epsilon_{low}/\epsilon_{high}}$$

4.4 Pokrycie

W przypadku tego programu, pokryciem określa się stosunek aktywnych(nie rozładowanych) węzłów sensorycznych do całkowitej liczby węzłów.

4.5 Czas życia

Jako czas życia sieci określa się czas (ilość rund), które upływają od 100% pokrycia aż do osiągnięcia przez sieć wartości pokrycia równej α

4.5.1 Algorytm Naiwny

Jako algorytm naiwny, w tym dokumencie brany jest pod uwagę prosty algorytm iteracyjny, który wywołuje elementy sieci węzeł po węźle, aż do przekroczenia progu pokrycia.

4.5.2 Algorytm Optymizacji Cząsteczkami Roju

W ramach algorytmu optymizacji rojem cząstek, używane są funkcje dopasowania, obliczania iou, pokrycia i wag

- Na początku symulacji, przeprowadzana jest selekcja potencjalnych miejsc, w których mogą znajdować się klastry węzłów

- Po selekcji, w wybranych miejscach znajdowani są odpowiedni kandydaci na głowy klastrów(Cluster Head)

5 Badanie

5.1 Cel badania

Celem badania, jest sprawdzenie, czy zaproponowana aplikacja do symulowania bezprzewodowej sieci sensorowej poprawnie emituje zachowanie się faktycznej sieci sensorowej.

5.2 Metoda badania

Zostaną przeprowadzone symulacje czasu życia sieci sensorowej, przy użyciu algorytmu naiwnego, oraz algorytmu PSO. Wyniki zostaną porównane z wynikami, które dostępne są w literaturze traktującej o zastosowaniu algorytmu optymizacji rojem cząstek w bezprzewodowych sieciach sensorowych.

5.3 Założenia ogólne

- Pobór energii zachodzi jedynie w przypadku trasnmisji danych od węzła

5.4 Założenia dla algorytmu naiwnego

- Każdy węzeł sieci, poza stacją bazową, komunikuje się w systemie pojedyńczych skoków. To znaczy, że aby dotrzeć do stacji bazowej, wykonuje pojedyńczy ruch na drodze węzeł-stacja.
- Węzły są wywoływane iteracyjnie, jeden po drugim, aż do przekroczenia progu minimalnego pokrycia

5.5 Założenia dla algorytmu optymizacji rojem cząstek

-Jest wywoływany na początku symulacji i za każdym razem, gdy rozładuje się węzeł będący głową klastra. Dokonana jest wtedy reselecja klastrów dla pomniejszonej populacji sieci
-Jeżeli głowa klastra znajduje się w odległości mniejszej niż d_0, to komunikuje się on bezpośrednio ze stacją bazową, w przeciwnym wypadku, szuka drogi poprzez inne głowy klastrów.

5.6 Ustawienia dla poszczególnych przypadków

5.6.1 Przypadek 1: mało czujników

W tej sytuacji, badanie będzie miało na celu sprawdzenie skrajnego przypadku.

- Sieć o wymiarach: 200x200
- Ilość węzłów: 50
- Pojemność ogniwa w węzłach[J]: 1
- Minimalny Procent Pokrycia: 70
- Ilość powtórzeń symulacji: 10

5.6.2 Przypadek 2: dużo czujników

- Sieć o wymiarach: 200x200
- Ilość węzłów: 200
- Pojemność ogniwa w węzłach[J]: 1
- Minimalny Procent Pokrycia: 70
- Ilość powtórzeń symulacji: 10

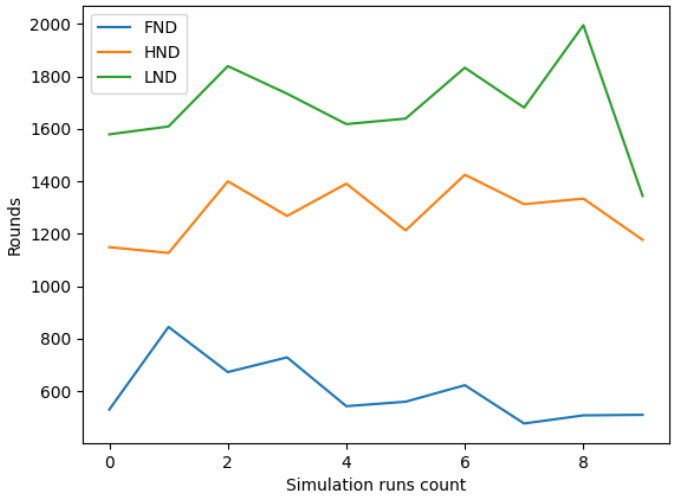
6 Wyniki Badania

6.1 Objaśnienie pojęć

W następujących badaniach użyte zostaną określenia:
- First Node Dead (FND) - Numer rundy, w którym wyłączył się pierwszy węzeł sensoryczny, czyli moment, w którym pokrycie spadło ze 100%,
- Half Nodes Dead (HND) - Numer rundy, w którym pokrycie spadło poniżej $100\% - \alpha/2$. Gdzie α to procent minimalnego pokrycia,
- Last Node Died (LND) - Numer rundy, w którym wyłącza się ostatni węzeł, po którym pokrycie spada poniżej wyznaczonego progu.

6.2 Przypadek 1

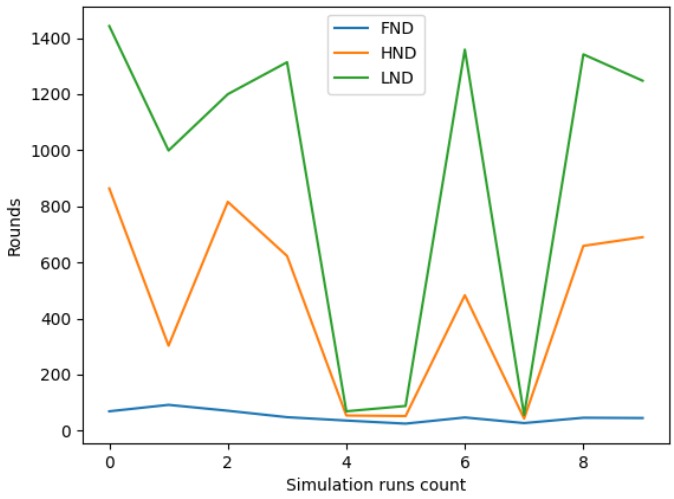
Z wykresu zależności stanu sieci od ilości wykonanych rund, dla algorytmu naiwnego, przy 50 węzłach, Rys 4. można wywnioskować, że wyniki, które dostarcza algorytm naiwny są spójne



Rysunek 4: Wykres progów pokrycia sieci. Źródło, badania własne

Z kolei wykres zależności stanu sieci od ilości wykonanych rund, dla algorytmu optymizacji rojem cząstek, przy 50 węzłach, Rys 5. wskazuje na losową niską efektywnością

algorytmu. Jest to powiązane z losowością ruchu roju cząstek.



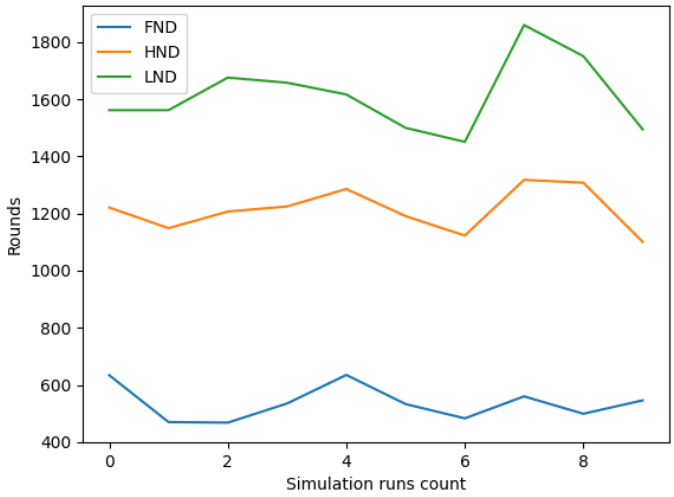
Rysunek 5: Wykres progów pokrycia sieci. Źródło, badania własne

W poniższej tabeli znajdują się uśrednione wyniki badania przypadku 1.

Algorytm	FND	HND	LND
Naive	571	1279	1687
PSO 50	458	912	

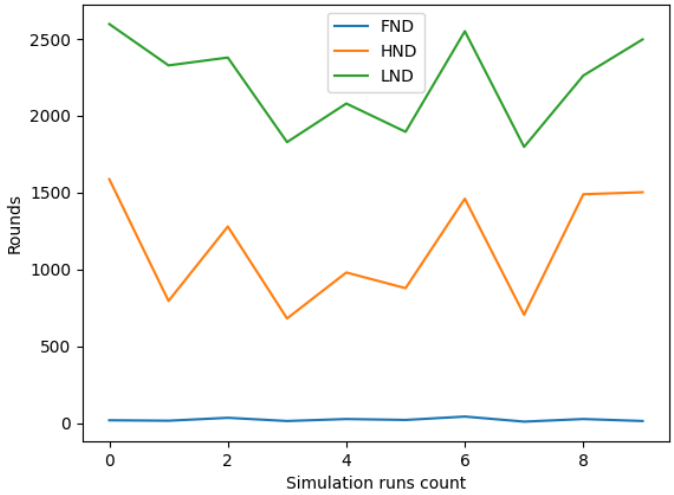
6.3 Przypadek 2

Wykres zależności stanu sieci od ilości wykonanych rund, dla algorytmu naiwnego, przy 200 węzłach, Rys 4. prezentuje spójne wyniki



Rysunek 6: Wykres progów pokrycia sieci. Źródło, badania własne

W przypadku wykresu zależności stanu sieci od ilości wykonanych rund, dla algorytmu optymizacji rojem cząstek, przy 200 węzłach, Rys 5. problem losowych krytycznie niskich wyników zdaje się być zażegnany. Rozwiązanie to dostarcza teraz wyników o wyższym wskaźniku LND.



Rysunek 7: Wykres progów pokrycia sieci. Źródło, badania własne

Tabela z uśrednionnymi wynikami dla poszczególnych stanów:

Algorytm	FND	HND	LND
Naive	542	1212	1613
PSO	23	1136	2221

7 Wnioski

Dowiedzione zostało, że zaproponowany symulator Sieci Sensorowej jest w stanie emulować sieć w sposób conajmiej w jakimś stopniu odzwierciedlający rzeczywistość. Wyniki w [1] przynoszą podobne rezultaty, algorytm PSO w trasowaniu okazuje się być rozwiązaniem wydłużającym czas życia Sieci Sensorowej. Warto jednak zauważyć pewną zależność, algorytm ten, dla małych ilości sensorów, a co za tym idzie, małej ilości cząstek roju, wykazuje pewne losowe zachowania. Ta tendencja jest uwarunkowana założeniami, na których opiera się ów algorytm. Ruchy cząstek mają bowiem w sobie pewną losowość.

Literatura

[1] Danwei Ruan and Jianhua Huang, "A PSO-Based Uneven Dynamic Clustering Multi-Hop Routing Protocol for Wireless Sensor Networks"

[2] A. Dhawan, “Maximum Lifetime Scheduling in Wireless Sensor Networks”

[3] A. Singha, S. Sharmab, J. Singh, “Nature-Inspired Algorithms for Wireless Sensor Networks: A Comprehensive Survey”

[4] R. Muthukkumar, Lalit Garg, K. Maharajan, M. Jayalakshmi, Nz. Jhanjhi, S. Parthiban and G. Saritha, “A genetic algorithm-based energy-aware multi-hop clustering scheme for heterogeneous wireless sensor networks”

[5] Shaimaa Alrashed, Paulvanna Nayaki Marimuthu, and Sami J. Habib, "Optimal Deployment of Actors using Simulated Annealing within WSAN"

[6] Konstantinos P. Ferentinos *, Theodore A. Tsiligiridis, "Adaptive design optimization of wireless sensor networks using genetic algorithms"

[7] Ankur Choudhary, Arun Prakash Agrawal, Rajasvaran Logeswaran, Bhuvan Unhelkar, "Applications of Artificial Intelligence and Machine Learning"

[8] D. BenHaddou and A. Al-Fuqaha (Eds.), pp. 3-32, Springer, New York, 2015., "Wireless Sensor and Mobile Ad-Hoc Networks: Vehicular and Space Applications"

[9] D. Monk, Texas instruments: A history of innovation

[10] H. Weinberg, "How They Work: MEMS (Micro Electro-Mechanical Systems) Technology,"

[11] Vikram Iyer*, Maruchi Kim*, Shirley Xue*, Anran Wang, Shyamnath Gollakota, "Airdropping Sensor Networks from Drones and Insects"

[12] Paweł Piotr Czapski, "BEZPRZEWODOWE i MOBiLNE SiECi SENSORYCZNE – STAN WiEDZY"

[13] I. F. akyildiz, W. Su, y. Sankarasubramaniam, and e. Cayirci, Wireless Sensor Networks: a Survey”

[14] I. F. akyildiz, W. Su, y. Sankarasubramaniam, and e. Cayirci, „a Survey on Sensor Network", IEEE Communications Magazine, vol. 40, no. 8, pp. 102-114, aug. 2002.

[15] Hamida, E.B.; Chelius, G. A line-based data dissemination protocol for wireless sensor networks with mobile sink. In Proceedings of the 2008 IEEE International Conference on Communications, Beijing, China, 19–23 May 2008.

[16] Shen, J.; Wang, A.; Wang, C.; Hung, P.C.K.; Lai, C.F. An Efficient Centroid-Based Routing Protocol for Energy Management in WSN-Assisted IoT. IEEE Access 2017, 5, 18469–18479.