

# Realizacja SWD implementującego metody optymalizacji wielokryterialnej

## Optymalizacja Wielokryterialna – ćwiczenie 5

### Autorzy:

Julia Nowak

Adam Złocki

Jakub Szczypek

### Temat projektu: Optymalizacja przejazdu zespołu robotów przez przeszkodę

W ramach naszego projektu zdecydowaliśmy się zająć jednym z dostępnych problemów optymalizacyjnych: **przejazdem zespołu n robotów przez przeszkodę**. Problem ten zakłada, że:

- Ruch robotów na przeszkodzie podlega zakłóceniom, które mogą prowadzić do ich uszkodzenia.
- Prawdopodobieństwo uszkodzenia robotów wzrasta, gdy są one położone blisko siebie, ponieważ zakłócenia mają większy wpływ na obszary o wysokiej gęstości robotów.
- Im większy dystans między robotami, tym mniejsze ryzyko uszkodzenia, jednak wydłuża to czas przejazdu całego zespołu.

Celem optymalizacji jest znalezienie **najlepszego kompromisu** między dwoma przeciwstawnymi kryteriami:

1. Minimalizacją sumy uszkodzeń robotów,
2. Minimalizacją czasu przejazdu zespołu przez przeszkodę.

### Proponowane podejście

W projekcie zamierzamy wykorzystać **algorytm wielokryterialnego PSO (Particle Swarm Optimization)**. Jest to heurystyczna metoda optymalizacyjna, która naśladuje zachowanie grupowe (np. ławic ryb czy stad ptaków), umożliwiając jednocześnie badanie wielu potencjalnych rozwiązań i stopniowe zbliżanie się do optimum.

### **Kroki realizacji:**

#### **1. Definicja problemu optymalizacyjnego:**

- Zdefiniujemy funkcje celu dla dwóch kryteriów:
  - Pierwsze kryterium będzie odpowiadało za minimalizację **sumy uszkodzeń**, modelując zależność między odległością robotów a ryzykiem uszkodzenia.
  - Drugie kryterium będzie odpowiadało za minimalizację **czasu przejazdu zespołu**, uwzględniając większe opóźnienia przy zwiększonych odległościach między robotami.
- Wprowadzimy ograniczenia (np. maksymalną liczbę robotów, maksymalny czas przejazdu czy minimalną odległość od przeszkody).

#### **2. Modelowanie zakłóceń i ryzyka uszkodzeń:**

- Użyjemy metryk odwzorowujących oddziaływania między robotami, np.:
  - Odległości euklidesowej lub Manhattan między robotami.
  - Zależności nieliniowych, uwzględniających efekty kumulacyjne zakłóceń w strefach wysokiej gęstości.

#### **3. Wielokryterialna optymalizacja za pomocą PSO:**

- Rozwiniemy klasyczny algorytm PSO o mechanizm pozwalający na jednoczesną optymalizację dwóch kryteriów (np. za pomocą funkcji agregacyjnej lub metody Pareto).
- Ustalimy odpowiednie parametry symulacji, takie jak liczba cząstek (robotów), wagi dla funkcji celu czy ograniczenia przestrzeni poszukiwań.

#### **4. Weryfikacja i analiza wyników:**

- Zweryfikujemy skuteczność algorytmu na symulowanych danych, takich jak różne układy przeszkód czy różne liczebności zespołów robotów.
- Dokonamy analizy efektywności algorytmu, porównując czas obliczeń i jakość rozwiązania w zależności od złożoności problemu.

## Możliwości rozszerzenia problemu

Rozważamy rozszerzenie projektu o:

- **Uwzględnienie specyficznych metryk:**
  - Przykładowo, można uwzględnić metryki związane z zużyciem energii robotów w zależności od trasy.
  - Wprowadzenie parametrów środowiskowych, takich jak różne rodzaje przeszkód lub zakłóceń (np. dynamiczne zmiany w otoczeniu).
- **Dodanie ograniczeń rzeczywistych:**
  - Uwzględnienie ograniczonego zasięgu komunikacji między robotami.
  - Wprowadzenie modelu awarii, w którym roboty mogą przestać działać po przekroczeniu określonego poziomu uszkodzeń.

## Źródła danych do symulacji

Aby zapewnić realistyczne warunki symulacji, przewidujemy wykorzystanie następujących źródeł danych:

### 1. Generowane dane testowe:

Przygotujemy zestawy danych reprezentujące różne konfiguracje przeszkód i zakłóceń, np. za pomocą generatorów map 2D/3D. Uwzględnimy różne liczby robotów i schematy przeszkód (proste przeszkody liniowe, labirynty, dynamiczne przeszkody).

### 2. Dane z literatury naukowej:

Przegląd badań nad optymalizacją ruchu robotów, np. na podstawie prac dotyczących swarm robotics lub optymalizacji trajektorii.

### 3. Publiczne bazy symulacyjne:

Dane z projektów takich jak Open Robotics czy ROS (Robot Operating System), które mogą dostarczyć realistycznych środowisk symulacyjnych dla robotów.

Przykładowymi źródłami danych mogą być:

- <https://library.fiveable.me/swarm-intelligence-and-robotics>
- [https://en.wikipedia.org/wiki/Particle\\_swarm\\_optimization](https://en.wikipedia.org/wiki/Particle_swarm_optimization)
- <https://www.sciencedirect.com/science/article/pii/S0304389424004114>

## Technologia

Symulacja zostanie zaimplementowana przy pomocy języka Python. GUI powstanie z wykorzystaniem biblioteki PyQT.