Aleksander Wawerka 160624

Adrian Zieliński 160420

**Projekt z przedmiotu Roboty Mobilne**

1. **Temat projektu**

Realizowany przez nas temat projektu: ***"Grafowe metody planowania ścieżki"****,* a opis projektu to *"Porównanie trzech wybranych grafowych metod planowania ścieżki. Początek trasy u wybrzeży Japonii na wysokości Tokio, koniec u wschodnim wybrzeżu Anglii na wysokości Londynu. Wolną przestrzeń konfiguracyjną stanowią wszystkie zbiorniki wodne, kontynenty i wyspy to przeszkody."*

1. **Opis wykorzystanych algorytmów**

W projekcie wykorzystaliśmy następujące metody przeszukiwania grafu:

* Algorytm A\*

Algorytm A\* jest heurystycznym algorytmem służącym do znajdowania najkrótszej ścieżki w grafie. Jest to algorytm zupełny i optymalny, co oznacza, że zawsze zostanie znalezione najlepsze rozwiązanie. W metodzie tej istnieje zorganizowany model pamięciowy, który gwarantuje że każdy punkt może zostać odwiedzony. A\* jest przykładem metody „najpierw najlepszy”. Algorytm A\* działa najlepiej, gdy przestrzeń przeszukiwań jest przestrzenią drzewiastą.

* Algorytm Djikstry

Jest to algorytm wyznaczający najkrótsze ścieżki z jednego wierzchołka (zwanego wierzchołkiem źródłowym) do pozostałych wierzchołków. Algorytm wymaga, aby wagi krawędzi grafu nie były ujemne. Autorem algorytmu jest holenderski naukowiec Edsger Dijkstra.

* Algorytm Best-first - podejście zachłanne

Wersja zachłanna algorytmu (ang. greedy best-first) próbuje zmniejszyć koszt dojścia do punktu końcowego nie sprawdzając przy tym czy w dalszej perspektywie jest to opłacalny zabieg.

Ta wersja metody best-first nie jest algorytmem optymalnym ani zupełnym, nie zawsze znajdzie rozwiązanie problemu, a jeśli wyznaczy rozwiązanie, to nie musi być ono najkrótszą drogą.

1. **Realizacja projektu**

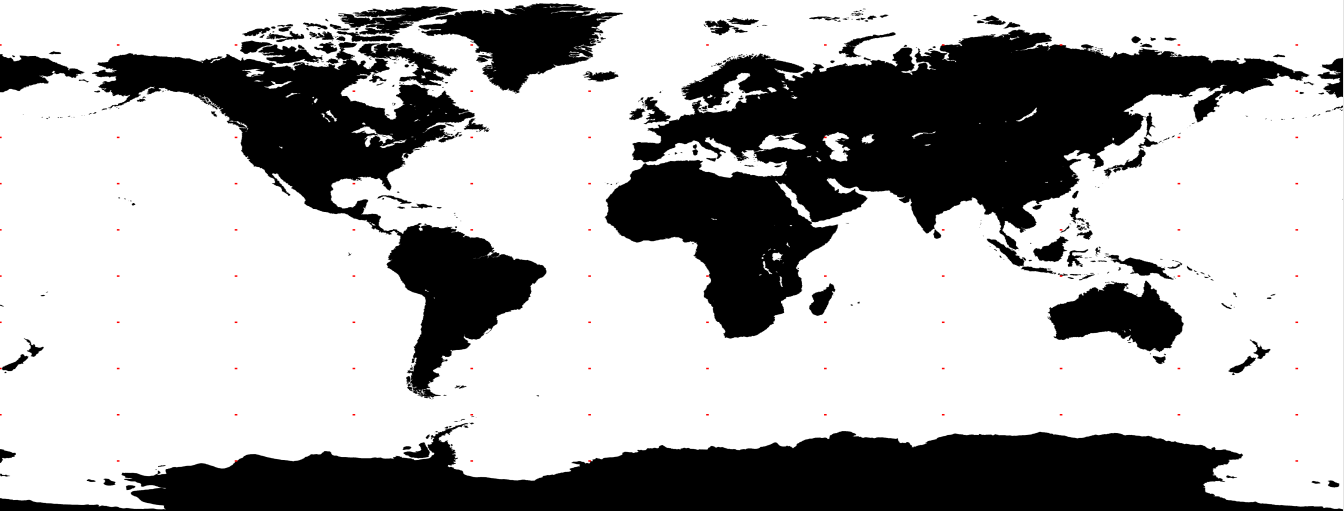
Na strukturę projektu składają się następujące elementy:

Implementacja klasy "Wierzchołek", w której definiujemy jego pozycję, nr identyfikacyjny, listę sąsiadów, odległości do poszczególnych sąsiadów, parametr informujący o tym czy dany wierzchołek znajduje się na obszarach lądowych czy wodnych. Dodatkowo na potrzeby algorytmu A\* oraz Djikstry definiowane są parametry takie jak: wyliczony koszt dojścia, informacja o tym czy wierzchołek został już wcześniej odwiedzony (A\*), informacja o poprzedniku (Djikstra).

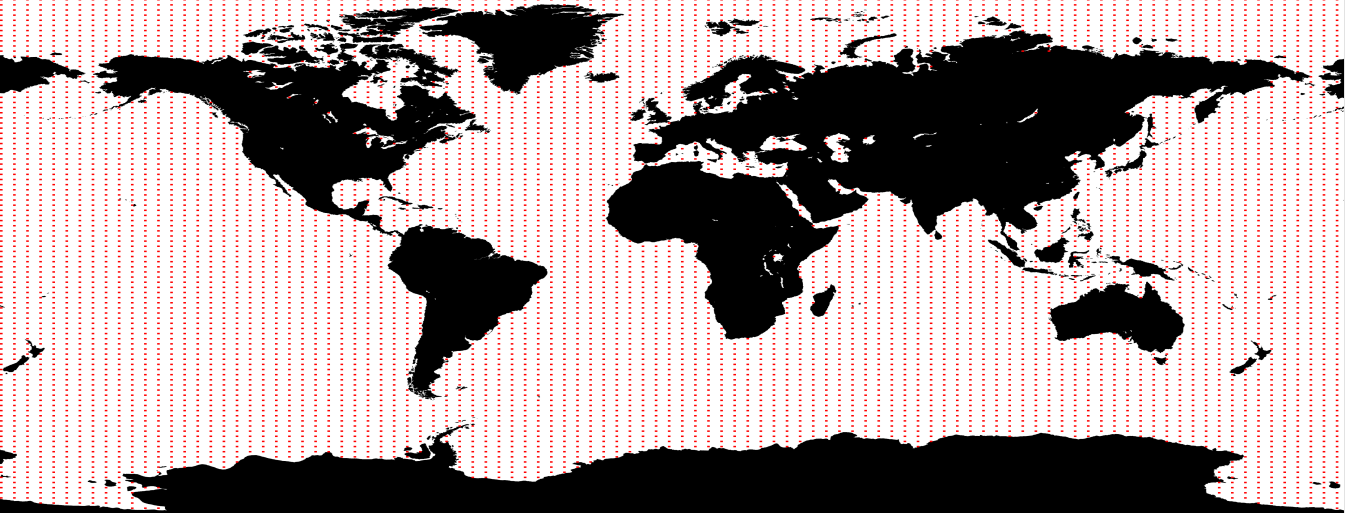
Klasa "Świat" umożliwiająca dobór dowolnego punktu początkowego oraz końcowego, które użytkownik może zmienić poprzez kliknięcie lewym przyciskiem myszy w wybrane miejsce na mapie. W tej klasie również odbywa się generowanie oraz naniesienie na mapę wierzchołków z wykorzystaniem pomocniczej klasy "Rysowanie", w której zawarta jest charakterystyka wyświetlanych punktów. Rozmieszczenie punktów zależne jest od stopnia wypełnienia, który użytkownik ustala za pomocą suwaka znajdującego się w górnej części interfejsu.



*Rys. 1 Suwak odpowiadający za gęstość rozmieszczenia punktów*

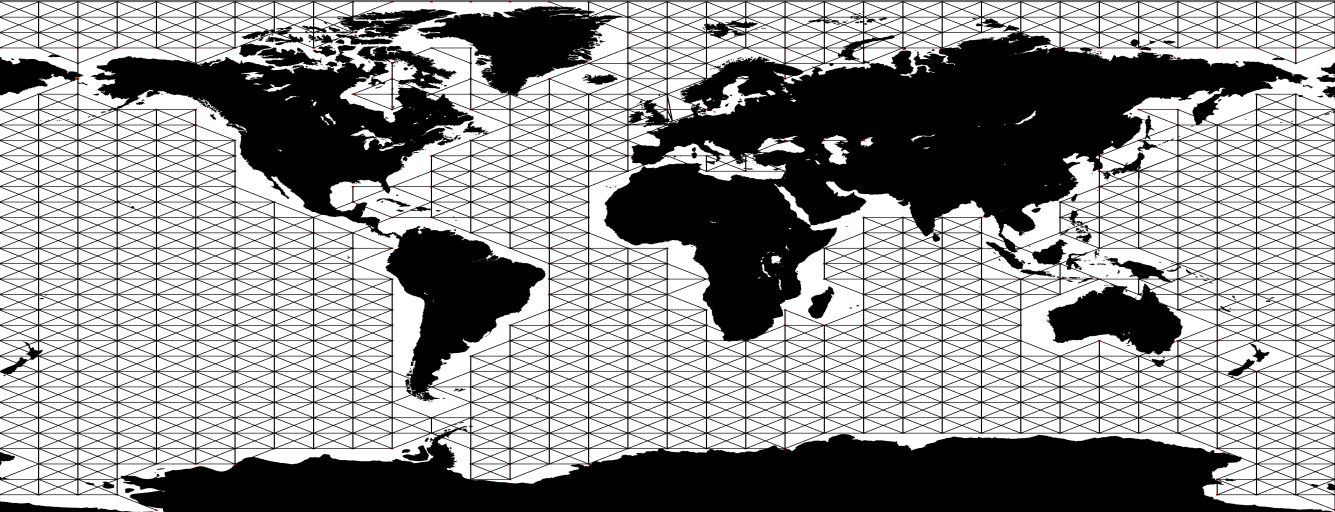
**

*Rys. 2 Rozmieszczenie punktów o małym stopniu wypełnienia*

**

*Rys. 3 Rozmieszczenie punktów o maksymalnym stopniu wypełnienia*

W klasie tej zawarte jest również połączenie poprzednio wygenerowanych punktów w graf.

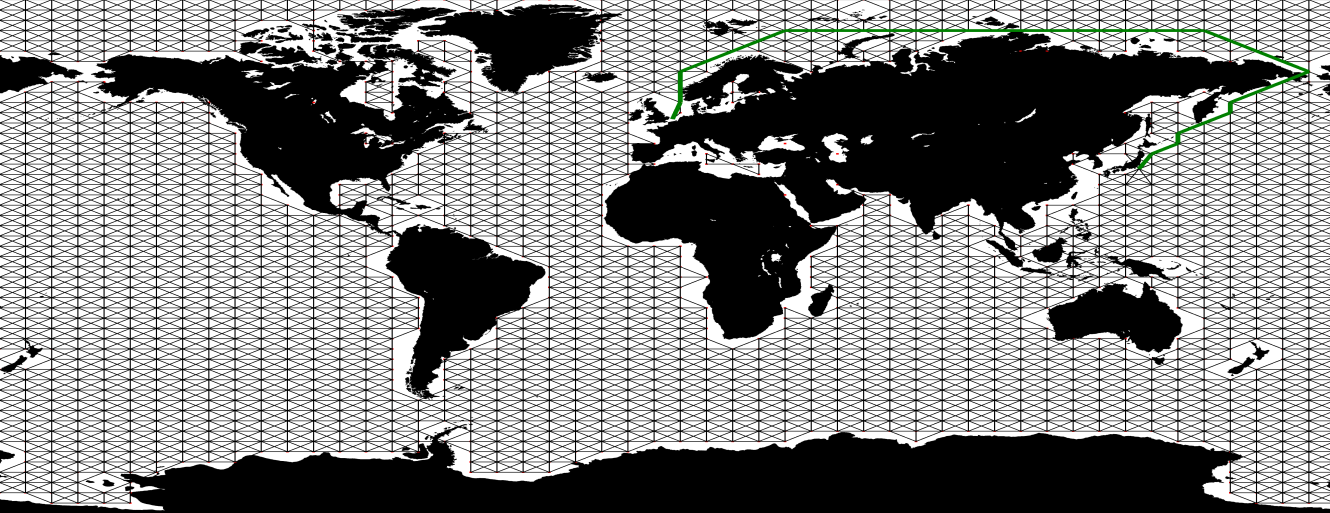


*Rys. 4 Graf powstały z połączenia sąsiadujących ze sobą punktów*

Po wygenerowaniu wierzchołków oraz grafu, a także określeniu punktów początkowego i końcowego pozostaje nam wyznaczenie najkrótszej ścieżki między nimi, które możliwe jest przy pomocy jednego z trzech algorytmów:

Algorytm A\*

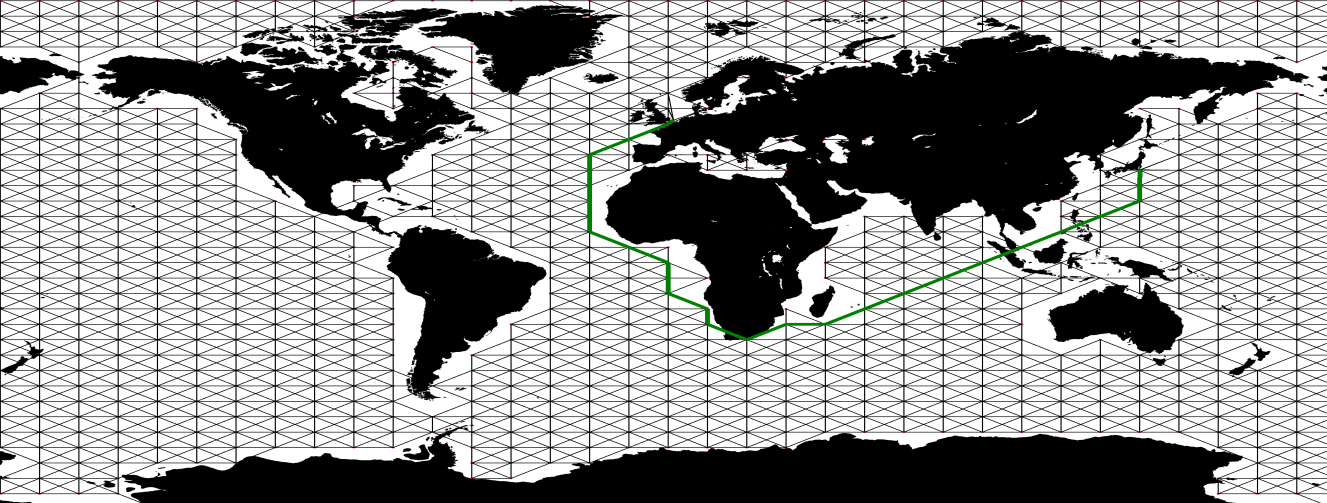
1. Dodajemy nieodwiedzonych sąsiadów aktualnego wierzchołka do listy "Otwarte".
2. Dla każdego sąsiada wyliczamy drogę dojścia, która równa jest sumie drogi dojścia do aktualnie badanego wierzchołka powiększonej o odległość między tym wierzchołkiem, a analizowanym sąsiadem.
3. Jeżeli analizowany sąsiad nie był dotąd odwiedzony bądź przypisana do niego wartość drogi dojścia jest większa od aktualnie wyznaczonej, zapamiętujemy aktualną wartość drogi dojścia jako najkrótszą.
4. Weryfikujemy czy suma odległości sąsiad - punkt końcowy oraz drogi dojścia do sąsiada jest mniejsza od odległości dotychczas uznawanej za najkrótszą ścieżkę i ewentualnie dokonujemy aktualizacji. Jeżeli zaś droga ta nie jest najkrótsza lub nie istnieje możliwość dotarcia do punktu końcowego z danego sąsiada, to aktualnie badany sąsiad zostaje usunięty z listy "Otwarte".
5. Rekonstruujemy ścieżkę dojścia na podstawie wyznaczonych kosztów dojścia.



*Rys. 5 Wyznaczenie najkrótszej ścieżki przy pomocy algorytmu A\**

Algorytm Djikstry:

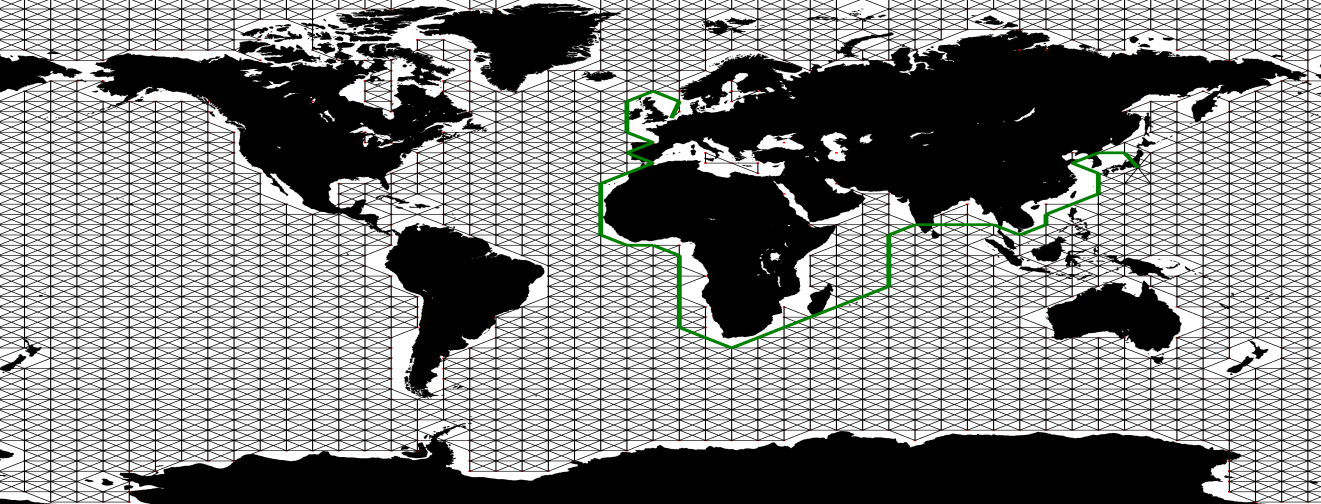
1. Zapisanie wszystkich wygenerowanych wierzchołków do listy nieprzeanalizowanych wierzchołków - Q.
2. Ze wszystkich wierzchołków listy Q wybieramy ten o najmniejszej drodze dojścia.
3. Wybrany wierzchołek usuwamy z listy Q i dodajemy do listy analizowanych wierzchołków - S.
4. Dla każdego z jego sąsiadów wpisanych do listy Q aktualizujemy drogę dojścia, jeżeli odległość od sąsiada powiększona o koszt własny jest mniejsza od dotychczasowej drogi dojścia bądź sąsiad był do tej pory nieosiągalny. Zapamiętujemy również jego poprzednika.
5. Po przeanalizowaniu wszystkich wierzchołków na liście Q odtwarzamy najkrótszą ścieżkę dojścia od punktu końcowego do początkowego po zapamiętanych poprzednikach.



*Rys. 6 Wyznaczenie najkrótszej ścieżki przy pomocy algorytmu Djistry*

Algorytm Best-first:

Ostatni z zastosowanych przez nas algorytmów jest bardzo zbliżony do algorytmu A\*, gdyż A\* należy do grupy metod "najpierw najlepszy". Różnica polega na tym, że przy obliczaniu drogi dojścia uwzględniamy jedynie drogę dojścia z analizowanego wierzchołka do punktu końcowego.

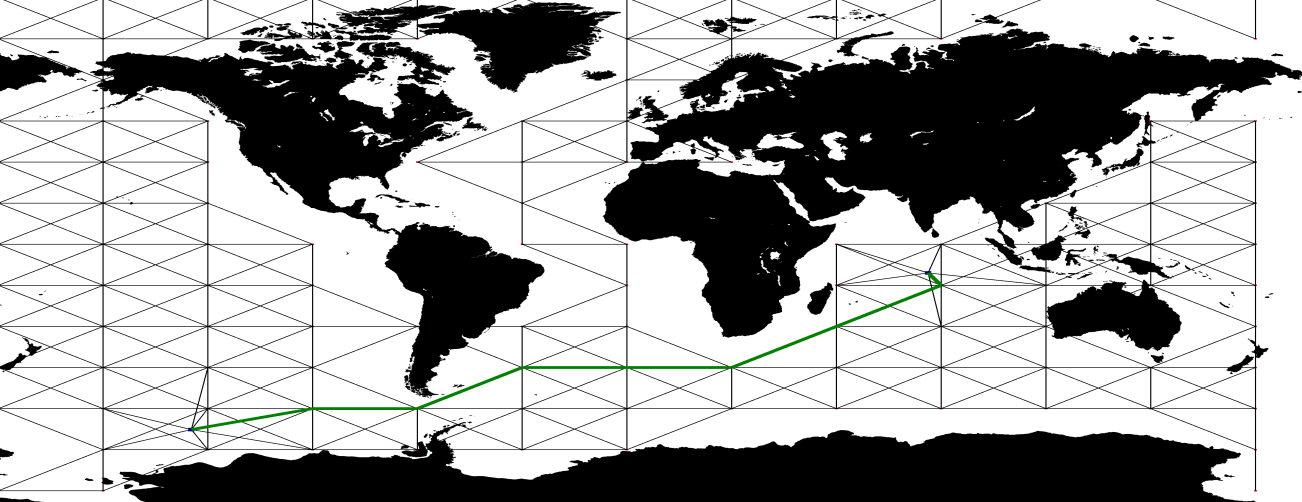


*Rys. 7 Wyznaczenie najkrótszej ścieżki przy pomocy algorytmu Best-first*

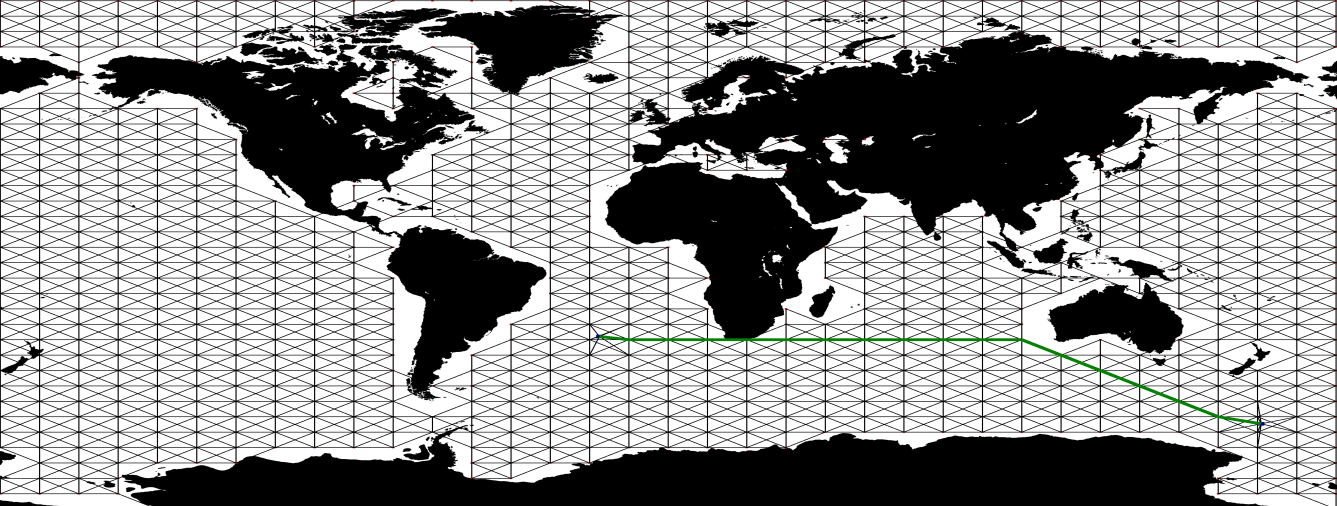
1. **Opis eksperymentów**

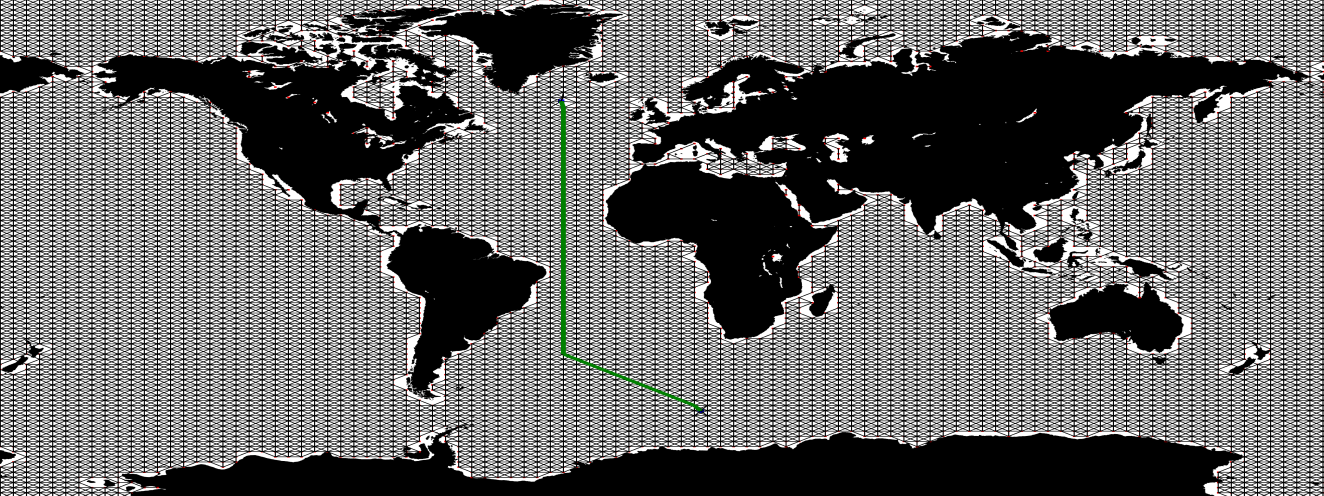
Pierwszy z wykonanych eksperymentów polegał na zbadaniu wpływu manipulacji stopniem zagęszczenia punktów na dokładność jej wypełnienia.

Sytuacje przedstawione na Rys. 8 - 10 pokazuje, że im więcej punktów naniesiemy na mapę, tym mniej jest "pustych" przestrzeni wokół lądów. Jednocześnie po ustawieniu najwyższego stopnia zagęszczenia punktów na mapie obserwujemy, że ścieżki połączeń między punktami przestają przecinać mniejsze wyspy - co również ma wpływ na dokładność wyznaczonej trasy i omijanie przeszkód.

****

*Rys. 8 Małe zagęszczenie punktów na mapie*

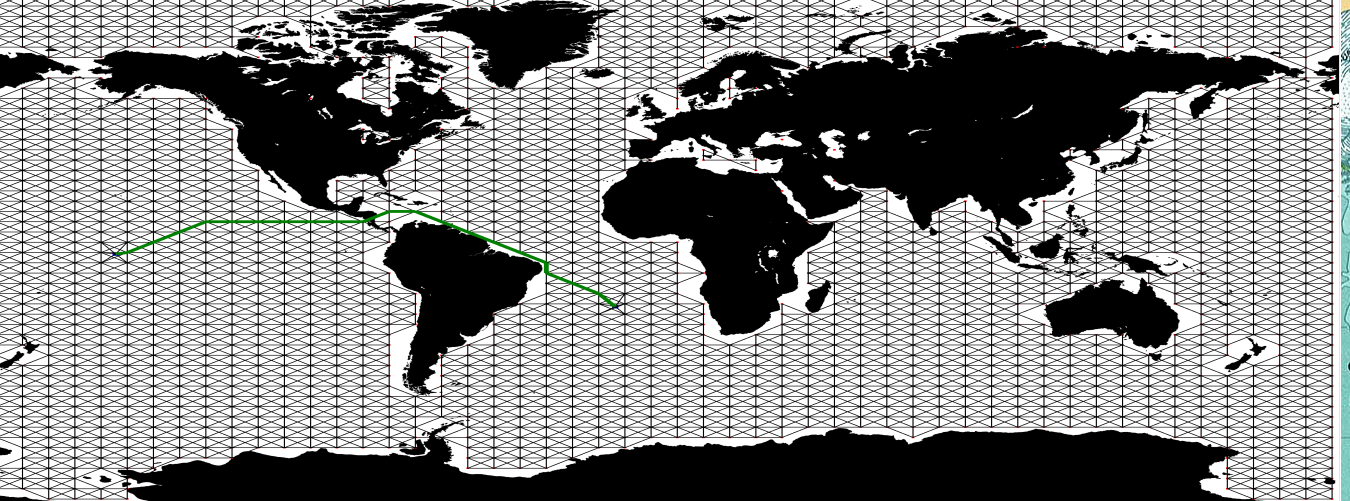
*****Rys. 9 Większe zagęszczenie punktów na mapie*

*****Rys. 10 Maksymalne zagęszczenie punktów na mapie*

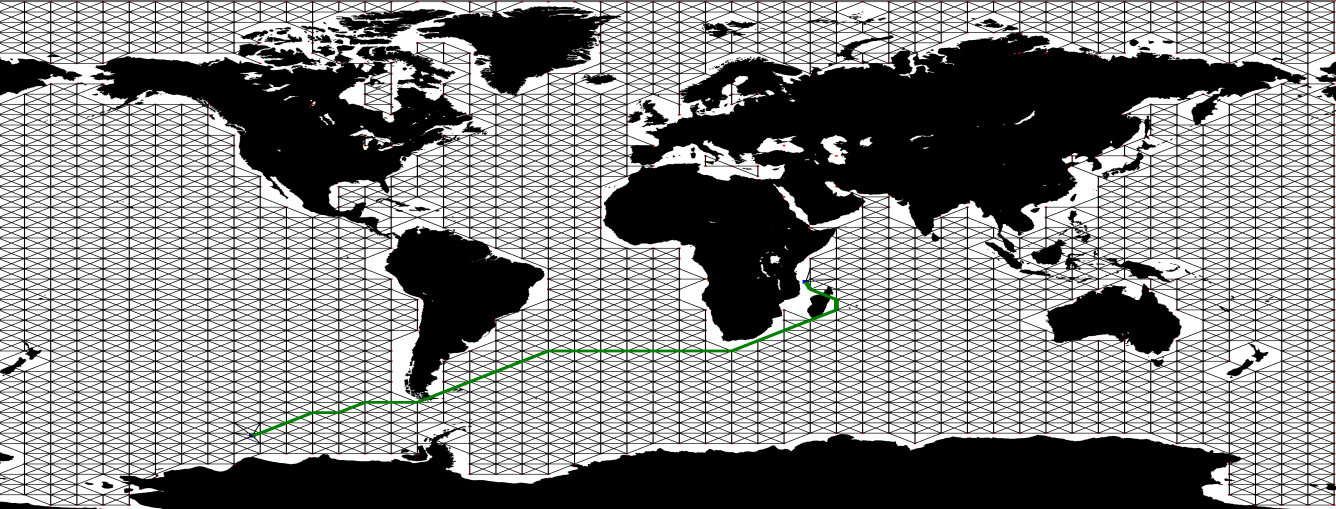
Ponadto na Rys. 8 - 10 przedstawiamy możliwość wyboru dowolnego punktu startowego i końcowego.

Kolejny eksperyment miał na celu weryfikację dokładności omijania przeszkód (lądów) przez algorytm. Niestety ze względu na równomierne rozmieszczenie punktów w równych odstępach zarówno w osi x jak i w osi y, małe obszary lądowe jak np. wyspy były ignorowane i wskazywane jako możliwe połączenie miedzy punktami, natomiast wąskie zbiorniki wodne (cieśniny, jak również Morze Czerwone) nie zostały wskazane jako możliwe do przepłynięcia.

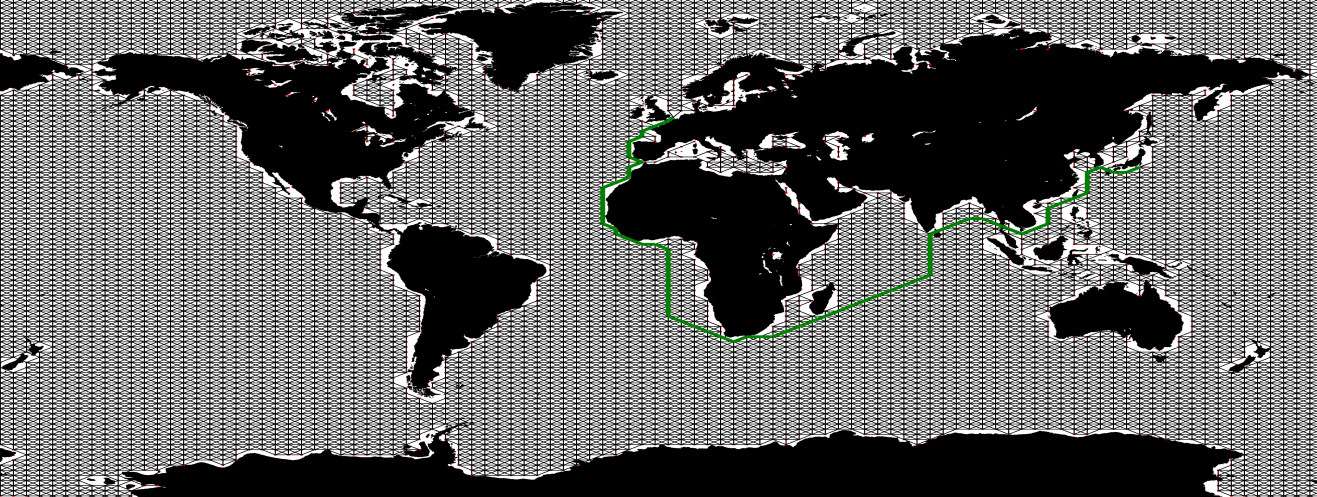
Rysunki 11 i 12 obrazują przykładowe błędy w wyznaczaniu ścieżki - poprowadzenie jej przez obszary niedostępne, ze względu na niepoprawne wyznaczenie połączeń między punktami lub jak w przypadku Madagaskaru na rysunku 12 - poprowadzenie ścieżki "na około" wyspy, zamiast optymalnie pomiędzy sąsiednim kontynentem a wyspą.

****

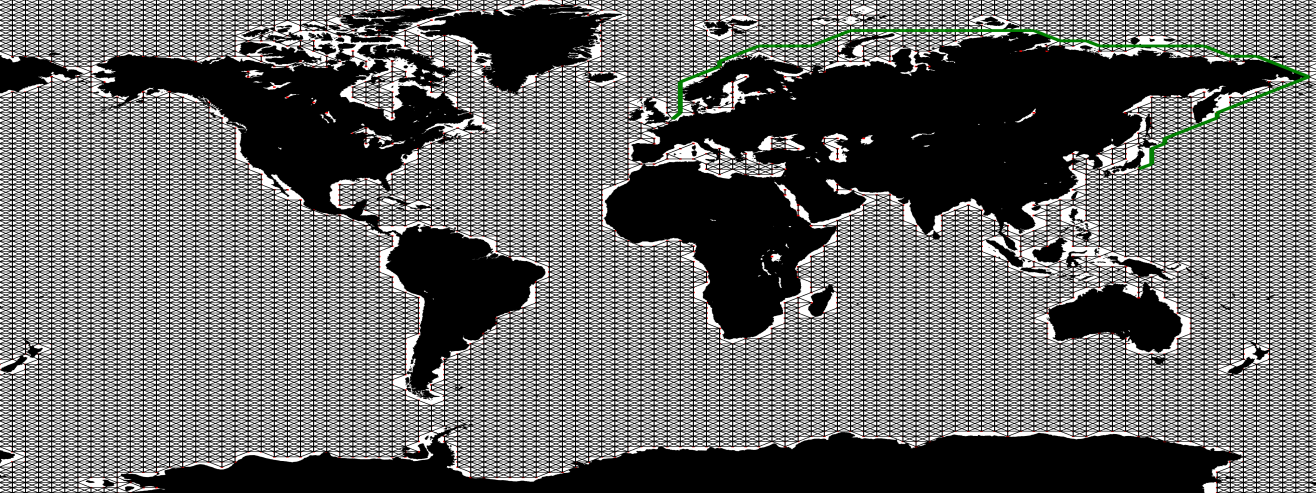
*Rys. 11 Błąd - zignorowanie wąskiego pasma lądu i przecięcie go ścieżką*

*****Rys. 12 Błąd - skrócenie ścieżki przez Madagaskar*

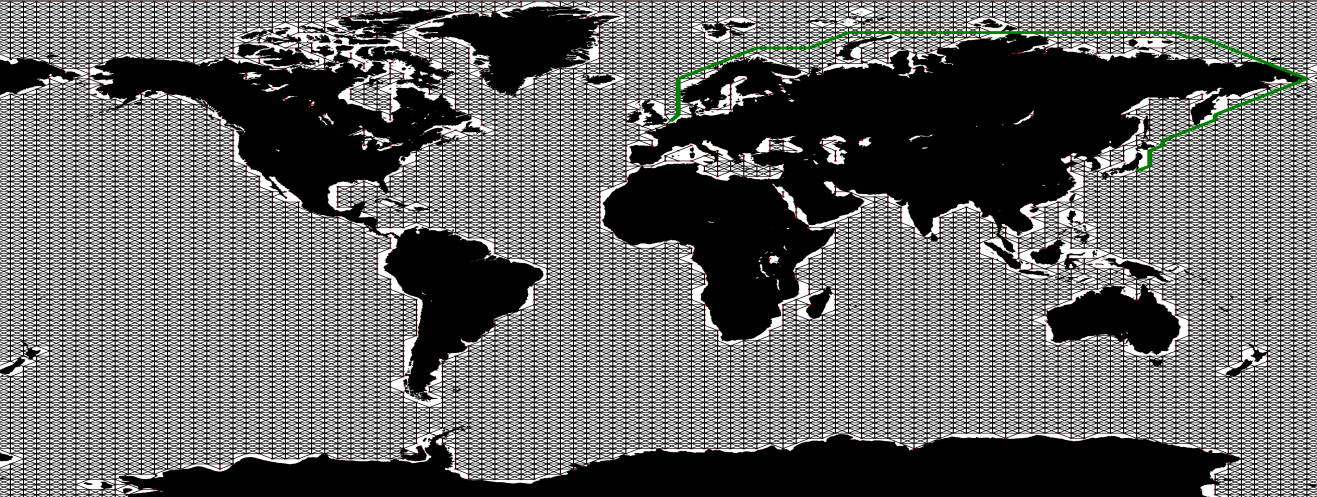
Ostatni eksperyment polegał na porównaniu sposobu działania trzech zaimplementowanych algorytmów. Na rysunkach 13 - 15 przedstawiamy efekty uruchomienia algorytmów na grafie o tym samym zagęszczeniu punktów. Jak widać ścieżki wyznaczone przez algorytmy Djikstry oraz A\* są podobne, choć ta wyznaczona metodą Djikstry ma krótsze proste odcinki i biegnie bliżej kontynentu. Metoda zachłanna najlepszy-pierwszy wskazała najmniej optymalną drogę, biegnącą wokół Afryki.

****

*Rys. 13 Działanie zachłannej wersji algorytmu najlepszy-pierwszy*

****

*Rys. 14 Działanie algorytmu Djikstry*

****

*Rys. 15 Działanie algorytmu A\**

1. **Wnioski**

Jako podsumowanie projektu można sformułować następujące punkty:

* Im większe zagęszczenie wierzchołków na mapie tym dokładniej pokryty zostaje obszar dostępny (zbiorniki wodne). Przesunięcie suwaka ku większej liczbie punktów sprawia jednak, że algorytm wyznaczający dostępne połączenia pomiędzy sąsiednimi punktami musi wykonać więcej operacji, a co za tym idzie wygenerowanie grafu zajmuje więcej czasu.
* Najkorzystniejszą metodą jest algorytm Djikstry. Pomimo tego, że algorytm A\* jest przykładem nie zachłannej metody Best-first, efekt jego działania jest bardziej zbliżony efektu uzyskanego przy pomocy algorytmu Djikstry, niż zachłannej odmiany Best-first.
* W dalszej pracy nad projektem należałoby zoptymalizować sposób weryfikacji dostępnych połączeń między sąsiednimi punktami - jednak opracowany przez nas program jest wystarczający na potrzeby realizacji zadania.