Team Jalayaan SIH – 2024

Problem Statement

Development of a versatile and fast algorithm for the optimal ship routing.

Aims

We Aim to take these points into utmost consideration.

1. Least fuel consumption.
2. **Travel time, passenger comfort and route safety** to avoid any damage to the ship, cargo, crew and passengers.
3. To avoid loss of life and property, **route weather safety** needs to be considered
4. There are no applications available publicly which can be customized for the **Indian Ocean region**
5. Complexity, computation time, versatility, etc. Various factors, such as, the forcings (surface winds, currents and waves), design of the ship and ship drift characteristics, impact the ship's motion at sea.
6. **The optimal route must be continually evolving** because the weather conditions keep changing as a ship proceeds on its voyage
7. **A suitable optimization method that can optimize several parameters for a range of ships** (with varying type, dimensions, drift characteristics of a ship) and develop an algorithm to return an optimal route within a **reasonable computational time**
8. The algorithm can optimize for the voyage time and safety to begin with but with a scope for **addition of more optimization parameters.**
9. The Project should be developed in an **open-source language such as Python.**

Objectives

1. **Develop Adaptive Ship Routing**: Create an algorithm optimized for the Indian Ocean, adjusting routes based on real-time data.
2. **Minimize Fuel and Travel Time**: Prioritize fuel efficiency and reduced travel time.
3. **Ensure Safety and Comfort**: Factor in passenger comfort and ship safety by accounting for weather and ocean conditions.
4. **Real-Time Weather Integration**: Continuously update routes based on real-time weather data.
5. **Support Multiple Ship Types**: Optimize routes for various ships with different designs and drift characteristics.
6. **Efficient and Feasible Optimization**: Balance multiple factors like safety and fuel within reasonable computation time.
7. **User-Friendly Web Interface**: Develop a simple, customizable web interface for the INCOIS website.
8. **Open-Source and Cost-Effective**: Use open-source technologies like Python to keep development budget-friendly and scalable.

Previous Work

In the domain of optimal ship routing algorithms, several research papers have contributed different approaches, each aimed at optimizing factors like fuel consumption, navigation time, and safety. After comparing four key research papers, the Particle Swarm Optimization (PSO) approach stands out as the most optimal and comprehensive algorithm for modern ship routing needs. This section provides a comparison of previous work, highlighting why PSO is the best and most suitable for our project with proposed improvements.

1. Ship Routing and Scheduling (2004)[1]

This paper focuses on foundational optimization techniques for ship routing, using deterministic models such as Mixed Integer Programming (MIP) and Dynamic Programming (DP). While these methods provide a solid basis for fleet utilization and cost minimization, they lack flexibility in real-time operations. The absence of uncertainty handling makes it less suited for modern shipping, where factors like weather changes and fuel price fluctuations play a crucial role.

2. Uncertainty in Maritime Ship Routing (2023)[2]

The 2023 paper introduces advancements by focusing on stochastic programming and robust optimization. These methods adapt to real-time uncertainties, such as fuel costs and travel time variations. While these models handle dynamic conditions well, they are computationally intensive, which may require further optimization for practical implementation in large-scale scenarios.

3. IIT KGP Paper: Modified Dijkstra's Algorithm[3]

The IIT KGP paper presents a modified Dijkstra's algorithm that minimizes travel time by accounting for weather conditions. While effective in specific regions like the North Indian Ocean, it is a single-criterion optimization model. It lacks the flexibility to address modern requirements like multi-objective optimization for fuel efficiency and safety, making it less adaptable to varying sea conditions.

4. Particle Swarm Optimization (PSO) Paper[4]

The PSO paper employs a hybrid of Particle Swarm Optimization and Genetic Algorithm (GA) to optimize multiple objectives: navigation time, fuel consumption, and safety. It dynamically adapts to real-time weather and sea conditions, offering diverse route choices. Its fast convergence and solution diversity make it suitable for complex maritime challenges. Validated under different weather conditions, the PSO algorithm is robust, flexible, and highly applicable to real-world operations.

Conclusion

The Particle Swarm Optimization (PSO) approach is the most comprehensive and effective algorithm for ship routing. Its ability to handle multi-objective optimization, real-time conditions, and diverse sea scenarios makes it superior to other methods. By improving the PSO algorithm, it can become even more efficient and adaptable, ensuring that it remains the best solution for modern maritime challenges.

DRAWBACKS

1. This algorithm does not focus on the Indian Ocean which opens us to fine tune it specifically for our desired proximity.
2. The algorithm is fast but it gets slowed down when multiple ships as inputs are taken into account and because of heavy computations and usage of a programming language like python will cause increase in run-time.
3. There is no scope for adding more parameters in future in this algorithm.

Methodology

1. Why modified PSO and GA are Better for Ship Routing:

1. **Multi-Objective Optimization:**
    - **PSO and GA** can handle multiple conflicting objectives (e.g., minimizing fuel, maximizing safety, and optimizing travel time) more efficiently.
    - **Dijkstra and A*** are typically used for single-objective shortest-path problems, which might not adapt well to balancing various factors like fuel, safety, and weather conditions.
2. **Handling Dynamic Environments:**
    - **PSO and GA** adapt well to changing environments, such as evolving weather, currents, and wave conditions. They can continually update the route as the ship progresses.
    - **Dijkstra and A*** are static and would need to re-run every time conditions change, leading to inefficiency in real-time applications.
3. **Global vs Local Optimization:**
    - **PSO and GA** are global optimization techniques, meaning they can explore the entire solution space to find a better overall route. This helps avoid getting stuck in local minima.
    - **Dijkstra and A*** tend to find locally optimal paths based on current conditions, which may not always be globally optimal in a dynamic environment.
4. **Incorporating Complex Constraints:**
    - **PSO and GA** can easily handle complex constraints such as weather conditions, ship drift, fuel efficiency, and varying ship types.
    - **Dijkstra and A*** have limitations when it comes to integrating non-linear or multi-dimensional constraints efficiently.
5. **Scalability and Versatility:**

- **PSO and GA** can scale to large, complex routing problems involving multiple ships, varying ship dimensions, and ocean currents. They are versatile in optimizing multiple parameters simultaneously.
- **Dijkstra and A***, though efficient for certain types of graph-based problems, struggle with scalability and complexity in large-scale, real-world applications with numerous variables.

6. **Non-Grid-Based Search:**
   - **PSO and GA** work in continuous space, which is better suited for real-world ship routing where the environment is not limited to a pre-defined grid.
   - **Dijkstra and A*** operate on a graph or grid, which can oversimplify real-world geography and constraints.

2. Main Methodology

Multicriteria Route Planning Framework:

Describes the overall framework used to solve the ship route planning problem specifically for the Indian Ocean by fine tuning our algorithm taking necessary variables into account, including the key elements like optimization criteria (time, fuel, risk), speed analysis, and route evaluation.

Optimization Criteria:

1. Navigation Time: Mathematical model to calculate total travel time based on ship speed and distance between waypoints.

2. Meteorological Risk: Model for calculating risk based on factors like wind and wave effects on the ship.

3. Fuel Consumption: Formula to estimate fuel usage based on ship speed and environmental factors.

4. A control variable: A variable let's say X which can be replaced by future parameters whenever required. As of now, X = 0.

Ship Speed Loss:

Explains how wind and waves affect the ship's speed and the formula used to calculate the actual speed of the ship under these conditions.

Population Model:

1. How the initial population of possible ship routes is generated and represented in the algorithm.

2. Defines the boundaries for generating random routes (waypoints) and ensures they respect static (e.g., land) and dynamic (e.g., weather) constraints.

Modified Particle Swarm Optimization (PSO) and Genetic Algorithm (GA):

PSO: Describes how ship routes (particles) update based on their best-known routes and the best routes found by others in the population.

GA: Describes how the genetic algorithm enhances route exploration with operations like crossover (combining routes) and mutation (randomly altering waypoints).

Fixing the time complexity by analysing and reducing the total computation especially while utilizing an open-source language like Python.

Multicriteria Modified PSO-GA Algorithm:

A detailed explanation of how PSO and GA are combined to form a hybrid algorithm.

Steps involved in how the algorithm evolves routes over multiple iterations to optimize for time, fuel, and safety, including:

1. Particle Cooperative Operation: Updating routes based on group cooperation.
2. Crossover Operation: Combining parts of two routes to form new ones.
3. Mutation Operation: Adding small random changes to routes to explore new solutions.
4. Multigroup Elite Selection: Selecting the best-performing routes to carry forward to the next generation.

Pareto Optimal Solution:

Method for generating a Pareto optimal set of solutions, which provides a range of routes that balance the different criteria (time, fuel, and risk).

The Pareto front ensures that no route is strictly better than another across all objectives, allowing for a variety of recommended routes.

Math Behind Methodology

It presents a hybrid ship route planning algorithm designed to optimize ship routes by considering three key criteria: **navigation time, fuel consumption, and meteorological risk**. The algorithm combines **Particle Swarm Optimization (PSO)** with **Genetic Algorithm (GA)** to enhance route efficiency, safety, and environmental impact.

Mathematical expressions involved in PSO and GA research paper

1. Navigation Time:

$$T_{total} = \sum_{i=0}^{n-1} t_i;\ t_i = \frac{L_i}{V_a^i}$$

This sums up the time it takes for a ship to travel each segment of the route.

Li is the length of a route segment, and Via is the ship's speed on that segment.

2. Distance Between Two Points:

$$\frac{l_2 - l_1}{\tan \varphi_{rh}} = \ln\left[\tan\left(\frac{\pi}{4} + \frac{\lambda_2}{2}\right)\left(\frac{1 - e\sin\lambda_2}{1 + e\sin\lambda_2}\right)^{\frac{e}{2}}\right] - \ln\left[\tan\left(\frac{\pi}{4} + \frac{\lambda_1}{2}\right)\left(\frac{1 - e\sin\lambda_1}{1 + e\sin\lambda_1}\right)^{\frac{e}{2}}\right],$$

$$L_{rh} = (\lambda_2 - \lambda_1) \cdot \sec\varphi_{rh},$$

This calculates the distance between two points (latitude and longitude) on a map.

λ represents latitude, and φrh is the angle direction between two points.

3. Risk Due To Winds (Index Measuring Risk Due To Winds){Meteorological Risk}:

$$risk_{wind} = \begin{cases} \frac{u_{cross}}{u_{10\max}}, & if\ the\ value\ is\ less\ than\ 1 \\ 1 \quad , & else \end{cases} .$$

This measures how risky it is to sail in crosswinds.

It is the ratio of wind speed hitting side of the ship to max wind speed ship can tolerate.

4. Wave Risk (Index Measuring Risk of Waves){Meteorological Risk}:

This shows how risky the waves are for the ship.

{Why Wave Risk: Under severe weather conditions, rolling is an important factor that causes a ship to capsize}

$$T_E = \frac{\lambda}{\left|1.25 \cdot \sqrt{\lambda} + V \cdot \cos\mu\right|}, \qquad T_\theta = \frac{2 \cdot C \cdot B}{\sqrt{GM}},$$

$$risk_{wave} = \begin{cases} \frac{T_\theta}{T_E} \quad , & if\ 0 \le \frac{T_\theta}{T_E} < 1 \\ 2 - \frac{T_\theta}{T_E}, & if\ 1 \le \frac{T_\theta}{T_E} < 2 \\ 0 \quad , & \frac{T_\theta}{T_E} \ge 2 \end{cases} .$$

Tθ is the ship's natural rolling period (how long it takes to roll back), and TE is the time between the ship encountering waves.

Note: there is an absolute risk when riskwave > 0.7, and when it is less than 0.7, the risk gradually decreases.

5. Total Risk:

The total risk is a weighted sum of wind, wave, and seakeeping risks. Here, a1, a2, a3 are weights, typically set to 1/3 each, meaning equal importance is given to each factor.

$$risk^i = a_1 \cdot risk^i_{wind} + a_2 \cdot risk^i_{wave} + a_3 \cdot risk^i_{seakeeping}, \quad \sum_{j=1}^{3} a_j = 1,$$

6. Fuel Consumption (Qti):

This calculates the fuel consumption per time unit.

a and b are constants based on the ship's design, and v is the ship's speed.

$$Q_{ti} = a \cdot e^{b \cdot v},$$

7. Speed Loss Due To Winds And Waves (Va):

This calculates the actual speed of the ship, which is slower than the usual speed V0 because of wind and waves.

Va=V0 − {some terms involving wind, wave height, and angles}

8. Population Model:

A set of possible ship routes (each represented by waypoints) that evolve over time.

Individuals (Routes):

An individual is a potential ship route, represented by a series of waypoints, each having latitude and longitude coordinates. Xi = A waypoint with lat and long.

$$\mathbf{X} = [X_0, X_1, \cdots X_i \cdots, X_{n-1}, X_n].$$

This formula generates random initial routes within upper boundary (UB) and lower boundary (LB) constraints (to keep the routes within feasible areas like avoiding land).

rand: Random values used to create diverse initial solutions.

$$X' = rand \cdot \{UB - LB\} + LB.$$

9. Optimization (PSO-GA Formula):

The ship's route is optimized using Particle Swarm Optimization (PSO), which updates particle positions (routes) by considering the best routes found by individual particles and the group.

$$v_{id}^{k+1} = \left(1 - \left(\frac{0.4}{MaxGen}\right) \cdot i\right) \cdot v_{id}^{k} + c_1 \cdot rand \cdot \left(p_{pBest} - x_{id}^{k}\right) + c_2 \cdot rand \cdot \left(p_{gBest} - x_{id}^{k}\right).$$

= $w.v^k_{id}$ + c1.(best individual position) + c2.(best group position) {velocity update}

{position update}: $X^{(k+1)}_{id} = X_{id}^{k} + v_{id}^{(k+1)}$

10. Pareto Solution Set (disi):

This calculates the distance between solutions in the optimization process to keep diversity in the route options.

$$dis_i = \sqrt{\sum_{j=1}^{n} [f_i(j) - f_{i+1}(j)]^2},$$

11. Mutation Operation:

In Genetic Algorithms (GA), mutation introduces randomness to help explore new solutions and avoid getting stuck in local optima. The math here applies a Gaussian mutation (a small random change near a reference value).

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}},$$

f(x): Describes a Gaussian distribution, which is used to create small random changes.

μ: Mean value (in this case, the reference route point where the mutation happens).

σ: Standard deviation (controls how far the mutation can go from the reference route).

$$\sigma = \frac{|UB_i - X_i|}{3}.$$

This sets the standard deviation σ based on how far the route point Xi can move within the boundaries (UB and LB).

12. Multi Elite selection operation:

This step ensures that the best solutions (routes) survive and pass on their information to the next generation, based on nondominated sorting and crowding distance.

$$u = F(p') = min\{f_1(p'), f_2(p'), \ldots, f_{n_0}(p')\},$$

For each individual p′ (route), we calculate its performance on each objective (e.g., time, risk, fuel).

If one individual p' is better than another q′ in all objectives, it is considered nondominated.

$$Crowd(m) = \sum_{j=1}^{n_0} \frac{f_j(m+1) - f_j(m-1)}{f_{jMax} - f_{jMin}}, \quad m = 2, 3, \ldots, N' - 1,$$

This calculates the crowding distance for each individual m within a population.

It measures how "close" an individual is to its neighbors in terms of its performance on objectives. Individuals with larger distances are preferred (to keep diversity in the population).

fj(m+1) and fj(m−1): Neighboring objective values (e.g., risk or fuel) for individual m.

Execution, SRS – dhruv

Benefits and future scope – vivek

References

1. Christiansen, Marielle, Kjetil Fagerholt, and David Ronen. "Ship routing and scheduling: Status and perspectives." *Transportation science* 38, no. 1 (2004): 1-18.
2. Ksciuk, Jana, Stefan Kuhlemann, Kevin Tierney, and Achim Koberstein. "Uncertainty in maritime ship routing and scheduling: A Literature review." *European Journal of Operational Research* 308, no. 2 (2023): 499-524.
3. Sen, Debabrata, and Chinmaya P. Padhy. "An approach for development of a ship routing algorithm for application in the North Indian Ocean region." *Applied Ocean Research* 50 (2015): 173-191.
4. Zhao, Wei, Yan Wang, Zhanshuo Zhang, and Hongbo Wang. "Multicriteria ship route planning method based on improved particle swarm optimization–genetic algorithm." *Journal of Marine Science and Engineering* 9, no. 4 (2021): 357.