Multi-objective Optimization for PATHfinding With Intelligent System Evolution (MO-PATHWISE)

Ashwin Raju

Abstract—**This research addresses the challenge of path planning in autonomous systems operating within random environments, focusing on Multi-Objective Optimization (MOO) to balance criteria such as safety, distance, and efficiency. Unlike traditional algorithms that optimize a singular objective, MOO provides a spectrum of optimal solutions, each representing a unique trade-off between the objectives. Drawing from N. Hohmann et al.'s "Hybrid Evolutionary Approach to Multi-objective Path Planning for UAVs" [5], this study explores the application of a Hybrid Evolutionary Strategy (HES) integrated with Non-Uniform Rational B-Splines (NURBS) for path representation. The research analyzes the complexities of MOO, emphasizing the flexibility of interchangeable objective functions and their results. The effectiveness of the HES in generating diverse, Pareto-optimal paths is demonstrated across simulated environments, indicating the method's proficiency over conventional path planning algorithms. The study paves the way for further exploration in 3D path planning and higher dimensional multi-objective scenarios, aiming to enhance the decision-making capabilities of autonomous systems for practical, real-world applications.**

I. Introduction

In the domain of autonomous systems and robotics, path planning stands as a fundamental challenge, especially in unpredictable environments. Traditionally, algorithms like Dijkstra's and Rapidly-exploring Random Trees (RRT) have been pivotal in shaping the landscape of motion planning, focusing primarily on achieving optimal paths based on singular objectives like shortest distance or feasibility. However, real-world scenarios often demand a more nuanced approach, where multiple objectives must be balanced simultaneously. This is where Multi-Objective Optimization (MOO) becomes critical.

MOO in path planning allows for a comprehensive consideration of various factors such as path distance, smoothness, safety, and feasibility. Unlike single-objective algorithms, MOO does not seek a singular 'best' path; instead, it aims to provide a set of optimal solutions, each representing a different trade-off among the objectives. This approach is particularly beneficial for autonomous systems operating in complex environments, where decision-making involves weighing multiple factors against each other. For example, in a search and rescue operation, an autonomous drone might need to balance the urgency of reaching a target quickly (minimizing path distance) against the need to navigate safely around obstacles (maximizing path safety).

The integration of MOO into autonomous systems is not without its challenges. The primary hurdle lies in developing a method that can effectively generate feasible and optimal paths. To tackle these challenges, this research embarks on a methodical approach, starting with a thorough review of existing literature and applications of MOO in robotics. This foundation will aid in the development of algorithms tailored for multi-objective optimization. Subsequent steps involve implementing these algorithms in simulated environments, fine-tuning the system, and rigorously evaluating its performance against existing methods.

The research's implications extend beyond academic interest, offering benefits in various industries where autonomous systems are employed. By advancing the capabilities of these systems in dynamic environments, the research aims to enhance their efficiency, adaptability, and overall utility in real-world applications. The ultimate goal is to elevate the decision-making processes of autonomous systems, enabling them to navigate complex environments with greater intelligence and autonomy.

II. Related Work

The field of robotic path planning has seen considerable advancements with the integration of Multi-Objective Optimization (MOO), particularly in static and semi-dynamic environments [1]. This section highlights the current trends and applications of MOO in path planning, emphasizing the static or predictable dynamic nature of the environments considered [2].

Recent innovations in MOO for path planning have been characterized by hybrid optimization strategies. Ajith and Jolly [2] introduced a distinctive framework for Unmanned Aerial Vehicles (UAVs), merging Particle Swarm Optimization (PSO) with Grey Wolf Optimizer (GWO), emphasizing the benefits of combining different algorithms to enhance convergence rates and solution diversity. This concept parallels our project's integration of MOO with evolutionary algorithms.

Tailoring MOO models to specific scenarios is essential for effective path planning. Ren, Shi, and Qiao [1] demonstrated this by adapting optimization models to static environments with various challenges, such as fixed obstacles. Their work aligns with our project's focus on MOO in predictable environments, showcasing MOO's effectiveness in environments where changes are either absent or occur in a predictable manner.

Bio-inspired algorithms have also gained traction in MOO for path planning. Jing, Zhang, Shen, and Zhang [3] enhanced the Ant Colony Optimization (ACO) method for path planning, proving the potential of bio-inspired algorithms in complex scenarios. These insights are valuable to our project as we navigate defining our MOO problem in similar contexts.

Comprehensive frameworks that merge various optimization techniques are also significant. Cortés et al. [4] introduced a framework combining Genetic Algorithms (GA) with Simulated Annealing (SA), skillfully addressing the integration of different optimization methods. Although done for different applications, this approach is reflective of our project's aim to explore core technical questions in MOO for path planning in static or predictable environments.

III. Technical Questions

Despite the seemingly abundant approaches to approaching MOO in path planning, the question on how to implement the framework still remains. Splitting this question into parts, we have three main blocks of questions to be addressed: representing the path, combining path characteristics, and real-world applications.

A. Defining Path Characteristics

- **Identifying Suitable Design Variables:**
  1) What characteristics of a path can be effectively leveraged as design variables in multi-objective optimization for path planning?
  2) How do general attributes, like the geometric layout, trajectory curvature, or potential energy costs, impact the overall objectives like efficiency, safety, and navigability?
- **Metrics for Path Efficiency, Safety, Navigability, etc.:**
  1) How can we quantitatively define and measure path characteristics like efficiency, safety, and navigability in environments with obstacles?
  2) What metrics can accurately reflect the impact of chosen design variables on path optimization objectives in complex scenarios?
- **Trade-offs in Multi-Objective Optimization:**
  1) How can we effectively assess and balance competing priorities in path planning, such as minimizing path length and maximizing path smoothness?
  2) What role do the chosen design variables play in navigating these trade-offs?

B. Combining Path Characteristics for Optimization

- **Algorithmic Approaches for Synthesizing Path Characteristics:**
  1) What algorithms or methods effectively integrate characteristics like path length, smoothness, and safety into an optimal path solution?
  2) How can we adapt these algorithms to dynamic environments where real-time data and changing conditions are constant?
- **Evolving Path Characteristics Toward Optimal Solutions:**
  1) How can evolutionary algorithms or heuristic methods be utilized to refine path characteristics towards an optimal balance of objectives?
  2) What strategies can be implemented within these algorithms to avoid convergence to local optima and ensure progression towards globally optimal solutions?

C. Real-World Application and Adaptability

- **Adaptability to Real-World Scenarios:**
  1) Considering the current framework's limitations, such as the absence of dynamic obstacles, how can the algorithm be adapted for real-world applications?
  2) What are the potential implications of these limitations on the applicability of the algorithm in various real-world scenarios?
- **Integrating Real-Time Data:**
  1) How can the integration of real-time sensor data be managed within the MOO framework to enhance its responsiveness and accuracy in dynamic environments?
  2) What challenges might arise in this integration, and how can they be addressed to ensure effective path planning?

Addressing these questions is pivotal for advancing MOO in path planning. The resolution of these technical questions will contribute to a deeper understanding of MOO, ultimately leading to more efficient, adaptive, and robust path planning solutions in robotics and autonomous systems.

IV. Technical Approach

Although many avenues were explored to tackle these technical questions, it was decided that exploring a specific implementation would be most beneficial in investigating the application of MOO in path planning. This project delves into a specific implementation strategy from existing research in the field. The approach is outlined in the work by N. Hohmann et al. [5], ”Hybrid Evolutionary Approach to Multi-objective Path Planning for UAVs,” which provides a framework for integrating MOO methodologies into a path planning strategy.

This section outlines the core components of our implementation of the technical approach seen in [5] beginning with the representation of paths using Non-Uniform Rational B-Splines (NURBS). It then progresses to explain the concept of Pareto Optimality as a fundamental aspect of MOO, followed by the implementation of the NSGA-II (Non-dominated Sorting Genetic Algorithm II) algorithm. Finally, the section culminates with a description of the evolutionary strategy adopted, highlighting its application in effectively navigating the complexities of path planning.

A. Path Representation: Non-Uniform Rational B-Splines (NURBS)

In the realm of path planning for MOO, the representation of paths is a critical factor. Non-Uniform Rational B-Splines (NURBS) are employed in this project due to their versatility and precision. Originating from uses in computer graphics and geometric modeling, NURBS have been widely used for their ability to accurately represent complex shapes and curves. As explained in Piegl et al [6], NURBS curves are defined by a set of control points, weights, and a degree of the basis function, allowing for complex and precise path formation.

The NURBS curve $C(u)$ is mathematically expressed as:

$$C(u) = \frac{\sum_{i=0}^{np-1} N_{i,p}(u) w_i P_i}{\sum_{i=0}^{np-1} N_{i,p}(u) w_i} \tag{1}$$

where:

- $C(u)$ is a piecewise polynomial curve, representing the path in a continuous and smooth form.
- np denotes the number of control points. Control points are used in shaping the curve. They represent where on the path curves occur, influencing the overall geometry. The more control points used, the more complex and detailed the curve can be.
- p represents the polynomial degree of the basis function $N_{i,p}$. The degree of the basis function determines the smoothness and flexibility of the curve. As in a conventional polynomial function, a higher degree allows the curve to be smoother and more flexible, allowing for more intricate paths to be represented.
- $P_i = (x_i, y_i)$ is the ith control point (assuming a 2D curve). These points are the physical locations in space where the curve is manipulated to pass through or near.
- w_i is the weight associated with the ith control point. Weights adjust the influence of each control point on the curve. A higher weight makes the curve cling closer to the control point, thus giving more prominence to that section of the path.

The choice of NURBS for path representation in MOO offers several benefits:

- **Flexibility and Control:** The control points and weights provide a high degree of flexibility, enabling the curve to take various shapes and forms. This flexibility is pivotal in navigating through complex environments where path adaptability is key.
- **Precision in Path Planning:** NURBS allow for precise control over the curve’s shape, essential for ensuring accurate and feasible paths in constrained spaces.
- **Suitability as Design Variables:** In MOO, the ability to adjust control points and weights makes NURBS an ideal choice for design variables. These variables can be finely tuned to optimize multiple objectives, such as path length, smoothness, and obstacle avoidance.

B. Pareto Optimality

Pareto Optimality plays a central role in MOO, a method used to address complex problems involving multiple, ideally conflicting, objectives. As mentioned before, unlike traditional single-objective optimization, MOO doesn’t seek a singular optimal solution. Instead, it aims to find a set of solutions, each representing a different trade-off among the objectives.

1) Understanding Pareto Optimality: A solution is deemed Pareto optimal if none of its objectives can be improved without degrading at least one other objective. In other words, a Pareto optimal solution is not outperformed across all the objectives by any other solution. This makes Pareto optimality dependent on the set of available solutions.

2) The Pareto Front: The Pareto front is a boundary in the objective space that represents the set of all non-dominated, or Pareto optimal, solutions. Each point on this front corresponds to a solution that offers a unique balance of the objectives. In path planning, the Pareto front visualizes the trade-offs between different path characteristics, such as distance, smoothness, obstacle avoidance, etc. Figure 1 shows an example of a Pareto front with non-dominated solutions in red and dominated solutions in blue.

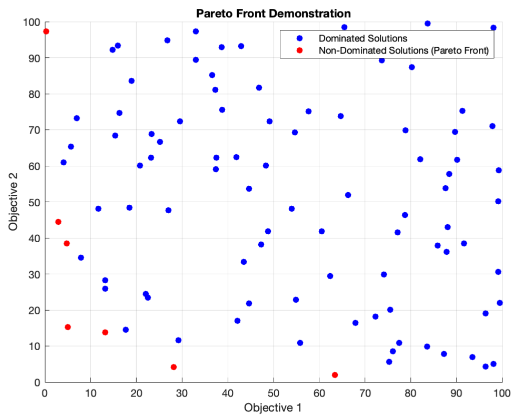

Fig. 1. An arbitrary Pareto front demonstrating the non-dominated (red) and dominated (blue) solutions in a two-objective minimization scenario.

3) Implications in Path Planning: In path planning, Pareto optimality allows for a nuanced understanding of the trade-offs involved in navigation. For instance, a shorter path might navigate closer to obstacles, compromising safety, while a safer path might be longer or less smooth. The Pareto front in this context helps in visualizing these trade-offs, providing insights into the nature of optimal paths under multiple criteria.

4) Application of Pareto Optimality: Applying Pareto optimality in MOO for path planning involves generating a diverse set of solutions that span the Pareto front. This approach ensures that the final set of solutions provides a comprehensive overview of possible trade-offs, allowing decision-makers to select a path that best aligns with their specific requirements or preferences.

C. NSGA-II

A renowned method for obtaining a Pareto front is the Non-dominated Sorting Genetic Algorithm II (NSGA-II), developed by K. Deb et al. [7]. It is used in generating optimal solutions for a Pareto front. The key components of NSGA-II are:

1) **Non-dominated Sorting:** Solutions are classified into different fronts of dominance levels. The first front consists of non-dominated solutions, while subsequent fronts are formed by solutions dominated by those in the previous fronts.
2) **Crowding Distance Assignment:** To ensure diversity in the solution set, each solution within a front is assigned a crowding distance, reflecting its spacing from neighboring solutions.
3) **Genetic Operators:** Selection, breeding, and mutation processes are employed to create new solution candidates. Breeding involves combining the characteristics (genes) of two individuals in a population, while mutation deals with applying small changes to the genes of an individual. These operators help in exploring the search space thoroughly.
4) **Population Management:** The algorithm evolves the population through generations by combining parents and offspring, selecting the best candidates based on their rank and crowding distance.
5) **Elitism:** This feature ensures the preservation of top-quality solutions over generations, resulting in NSGA-II identifying increasingly optimal solutions at each generation.

These components together enable NSGA-II to efficiently locate a diverse and high-quality set of non-dominated solutions, making it effective for solving complex multi-objective optimization problems, such as the one in this paper.

D. Evolutionary Strategy

With the foundational concepts of NURBS representation, Pareto Optimality, and NSGA-II laid out, we can proceed to implement the method described in [5]. This method employs a Hybrid Evolutionary Strategy (HES) that combines the flexibility of NURBS in path representation with the robustness of NSGA-II in generating Pareto optimal solutions.

1) Path Representation: In the proposed evolutionary strategy, the first crucial step is the representation of paths using NURBS. To ensure simplicity and consistency in the evolutionary process, certain parameters in the NURBS formulation are standardized. Specifically, the degree (p) of the NURBS curve ($C(u)$) is fixed at 3, and the number of control points (np) for all curves is set to 50.

Each path in this context is considered as an individual (I), with each individual defined by a set of design variables ($\mathbf{z}$) and corresponding step-sizes ($\boldsymbol{\sigma}$):

$$I = \{\mathbf{z}, \boldsymbol{\sigma}\} \tag{2}$$

The design variable vector ($\mathbf{z}$) and the step-size vector ($\boldsymbol{\sigma}$) are defined as follows:

$$\mathbf{z} = \begin{bmatrix} \vec{x_0} & \vec{y_0} & \vec{w_0} \end{bmatrix}^T \tag{3}$$

$$\boldsymbol{\sigma} = \begin{bmatrix} \vec{\sigma}_{x0} & \vec{\sigma}_{y0} & \vec{\sigma}_{w0} \end{bmatrix}^T \tag{4}$$

Here, $\mathbf{z}$ includes the control points and weights of the path, while $\boldsymbol{\sigma}$ represents the step-size for each design variable. Thus, the dimension of an individual, I, can be expressed as $[1 \times 6 \times np]$. In the Python implementation of the NURBS paths, the 'geomdl' library's NURBS package is utilized to represent lists of control points and weights as NURBS curves for subsequent analysis.

This representation of paths as individuals with fixed degrees and a set number of control points creates a consistent framework for the evolutionary strategy. It allows for the nuanced manipulation of path characteristics while maintaining the integrity and feasibility of each path solution within the MOO process.

2) Initial Population Generation: This is where the "Hybrid" comes from in the Hybrid Evolutionary Strategy described by "Hybrid Evolutionary Approach to Multi-objective Path Planning for UAVs," [5]. The initial generation process involves two primary methods: Dijkstra's Algorithm and Yen's Algorithm, in tandem with the traditional random generation technique to introduce genetic diversity.

1) **Employing Dijkstra's and Yen's Algorithms:** The initial step involves using Dijkstra's Algorithm to find the shortest path in the environment. The rationale behind this choice is that Dijkstra's Algorithm is efficient in finding the shortest path in a graph-based representation of the environment, providing a solid foundation for the initial population. Following this, Yen's Algorithm is applied to ascertain the subsequent k-shortest paths. This iterative process involves removing nodes from the previously identified shortest path and recalculating to discover alternative optimal paths. This is repeated until k shortest paths are determined. The use of Yen's Algorithm allows for the exploration of path variants that are close to the optimal solution, thereby enriching the initial population with feasible and diverse path choices.
2) **Promoting Genetic Diversity:** In addition to the paths found by Dijkstra's and Yen's Algorithms, η random paths are generated to further promote genetic diversity in the population. This is achieved by creating a straight line path from the start to the goal and introducing Gaussian random noise to various points along this line. The introduction of these random paths ensures that the initial population isn't overly concentrated around the paths found by Dijkstra's and Yen's Algorithms, thereby avoiding premature convergence and encouraging a greater exploration of the solution space.
3) **Forming the Initial Population:** The combination of paths from Dijkstra's/Yen's and the random path generation results in an initial population consisting of $n = k + \eta$ paths. For ease of implementation, the paths initially generated are interpolated into 50 waypoints, which serve as control points for the NURBS curves, aligning with the constraint set on np earlier. This diverse set of paths forms the basis for the subsequent evolutionary operations, ensuring that the algorithm has an optimal yet varied set of solutions to evolve from.
4) **Path and Step-Size Initialization:** As recommended by Hohmann et al., each path in the initial population has its weights set to 1, forming the basis of the NURBS curve representation. Correspondingly, the step-size vector for each individual, $\boldsymbol{\sigma}$, is initialized as $\begin{bmatrix}\vec{1} & \vec{1} & \vec{0}\end{bmatrix}^T$.

3) Breeding and Mutation Processes: In the evolutionary strategy, breeding and mutation are crucial for introducing genetic diversity and enabling the exploration of new and potentially more optimal paths. This step is where the evolutionary strategy differs from NSGA-II.

a) Breeding Operation: The breeding operation involves the combination of features from two selected individuals to produce offspring that inherit high-performing traits from both parents. This process averages the step sizes ($\boldsymbol{\sigma}$) of two randomly chosen individuals $I_{i,t}$ and $I_{j,t}$ at a particular generation t:

$$\boldsymbol{\sigma}_{i,t+1} = \boldsymbol{\sigma}_{j,t+1} = 0.5 \times (\boldsymbol{\sigma}_{i,t} + \boldsymbol{\sigma}_{j,t}) \tag{5}$$

This averaging of step sizes leads to the offspring inheriting a blend of the parents' characteristics, potentially leading to new path solutions that may perform better in terms of the objective functions. The decision to perform breeding is based on a probability threshold, ensuring that it occurs randomly and contributes to the diversity of the population. This approach differs from the traditional NSGA-II method, where the design variables of the individual would typically be bred. In this strategy, the focus is on breeding the step-sizes, which represent the change in design variables. This nuanced focus on the rate of change rather than the variables themselves introduces a unique dynamic to the evolutionary process, potentially leading to innovative and feasible paths with improved performance in the path planning objectives.

b) Mutation Operation: The mutation process introduces random changes to the individuals, also to the step-size vector, $\boldsymbol{\sigma}$. In our approach, this operation is applied to each element s of the step-size vector $\boldsymbol{\sigma}$ of the individuals. Mathematically, the mutation for each element s is represented as:

$$\sigma_{s,t+1} = e^{\tau_0 \xi_0} \left[\begin{bmatrix}\sigma_{1,t} e^{\tau \xi_1} & \dots & \sigma_{D,t} e^{\tau \xi_D}\end{bmatrix}\right]^T \tag{6}$$

Here, ξ represents a normally distributed random variable, and the learning rates τ_0 and τ are calculated based on the dimension D of the problem, where $D = 3np + 4$.

c) New Generation Creation: Following the breeding and mutation processes, new individuals $I_{r,t+1}$ are generated. The step-sizes after mutation guide the development of these new individuals, potentially leading to paths with better fitness scores:

$$I_{r,t+1} = [[w_{0,t} + \sigma_{1,t+1}\xi_1 + \ldots + w_{np,t} + \sigma_{D,t+1}\xi_D]]^T \quad (7)$$

This is where the paths are physically and geometrically altered, based on the results of the breeding and mutation operations. Figure 2 illustrates an example of a path before and after mutation, showcasing the changes in the path's geometry.

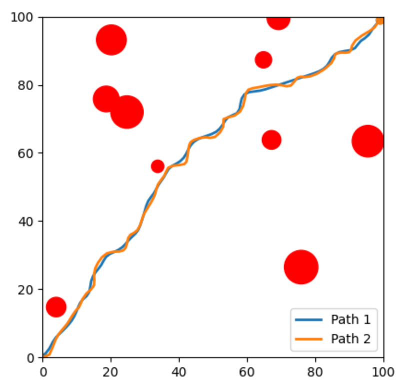

Fig. 2. Example of a NURBS path before and after mutation. Path 1 (pre-change) and Path 2 (post-change) demonstrate the effect of the mutation on the path's geometry.

4) Objective Functions Evaluation: The evaluation of each individual in the population is a crucial aspect of the evolutionary strategy, as it directly impacts the selection process for the next generation. Our approach employs two primary objective functions: path distance and path smoothness or safety distance from obstacles. The specific objective function used (path smoothness or safety distance) is interchanged based on the scenario to observe the results of MOO, illustrating the useability of the approach in different path planning contexts.

a) Path Distance: Path distance, a fundamental criterion in path planning, is the total length of the path. It is calculated as the sum of Euclidean distances between consecutive control points along the NURBS curve. Minimizing path distance is often a primary goal in efficient path planning, as shorter paths typically imply reduced travel time and energy consumption.

b) Path Smoothness: Path smoothness, crucial for navigational ease and stability, especially in autonomous systems, is quantified by the average curvature along the path. This measurement is obtained by iteratively calculating the curvature at numerous points along the curve by evaluating the first and second derivatives at each point. This utilizes 'geomdl's NURBS' inherent properties that facilitate easy computation of derivatives. Smoother paths are generally preferred as they indicate less frequent and abrupt changes in direction, which can be vital in sensitive or high-speed navigational contexts.

c) Safety Distance from Obstacles: Safety distance from obstacles represents the minimum clearance of the path from any surrounding obstacles. This measure is essential for ensuring the safety and reliability of the path, especially in cluttered or dynamic environments. The objective here is to maximize this distance, ensuring that the path maintains a safe buffer from obstacles, which is crucial in avoiding potential collisions and ensuring operational safety.

To evaluate an individual's performance, a NURBS curve is constructed using the individual's control points and weights. The initial step involves assessing the feasibility of the curve, checking for any intersection with obstacles or deviation from predefined bounds. Any curve violating these constraints receives a high penalty, reflecting its unsuitability.

For curves meeting feasibility criteria, we proceed to calculate the two objective measures: total path distance and either path smoothness or safety distance from the nearest obstacle. The inverses of path smoothness and safety distance are used as fitness values to align with the minimization process inherent in MOO. This inverse approach ensures that higher values correspond to smoother paths or paths with greater clearance from obstacles, thereby standardizing the optimization process across different objectives.

5) NSGA-II Selection: The selection process matches that of NSGA-II as it is based on the fitness results obtained during the evaluation of each individual. This fitness assessment considers the path distance, path smoothness, and safety distance from obstacles, among other factors like population diversity. The 'DEAP' (Distributed Evolutionary Algorithms in Python) library, is used to implement the selection process. It performs the selection mechanism of NSGA-II, which as mentioned before, involves sorting the population based on non-domination levels and calculating the crowding distances to maintain diversity.

6) Algorithm Execution: The evolutionary loop forms the backbone of the strategy, where the integral components of NURBS path representation, Pareto optimality, and NSGA-II selection combine in a dynamic, iterative process. Central to this loop is the generation of new populations through breeding and mutation, fostering genetic diversity and enabling the exploration of novel path solutions.

Crucially, each cycle of the evolutionary loop is designed to prevent convergence to local minima.

By continuously introducing new genetic variations and assessing the population against multi-objective criteria, the algorithm ensures that each generation is either an improvement or at least as competent as the previous one. Algorithm 1 shows the process for this evolutionary loop.

Algorithm 1 Hybrid Evolutionary Strategy for Multi-objective Path Planning

```
Initialize the space with random obstacles
Initialize Initial Population
Set Max Number of Generations
Set No Improvement Limit
Set Population to Initial Population
Initialize All Pareto Fronts to an empty list
Initialize No Improvement Count to zero

while not reached Max Number of Generations do
    Evaluate Population and assign fitness scores
    Calculate diversity of Population
    Select best individuals for Offspring
    if Breeding condition met then
        Perform breeding on Offspring
    end if
    Mutate Offspring
    Revaluate Offspring
    Combine Population and Offspring
    Select best using NSGA-II for new Population
    Determine and save current Pareto front
    Check for improvements in Pareto front
    if Improvement in Pareto front then
        Reset No Improvement Count
    else
        Increment No Improvement Count
    end if
    if No Improvement Count exceeds limit then
        Break
    end if
end while
Extract Pareto optimal paths from final front
```

V. Results

Three different random environments and initial populations were initialized to analyze the evolutionary method. The populations varied in size, with the first consisting of 16 total paths (10 generated using Yen's Algorithm and 6 randomly generated, i.e., $n = 16, k = 10, \eta = 6$). The second population had 100 total paths (70 from Yen's Algorithm, 30 random, i.e., $n = 100, k = 70, \eta = 30$), and the third comprised 500 paths (200 from Yen's Algorithm, 300 random, i.e., $n = 500, k = 200, \eta = 300$).

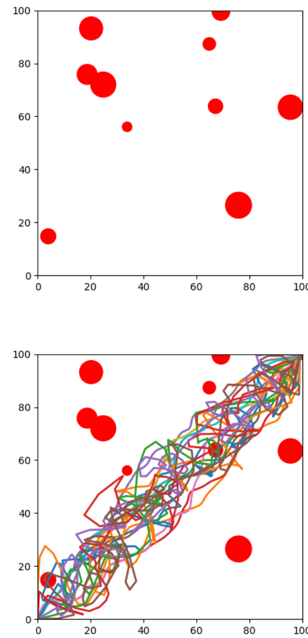

Fig. 3. Environment and Initial Paths for 16 Individuals.

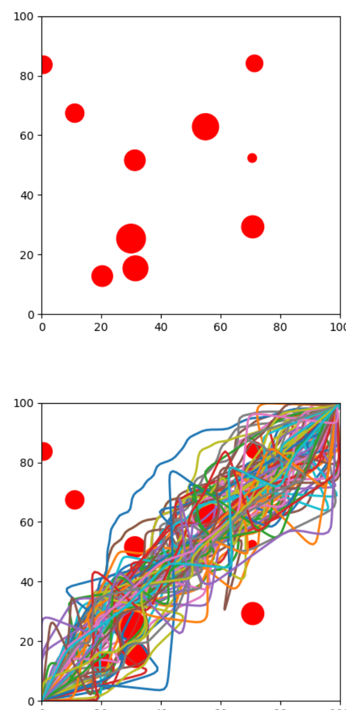

Fig. 4. Environment and Initial Paths for 100 Individuals.

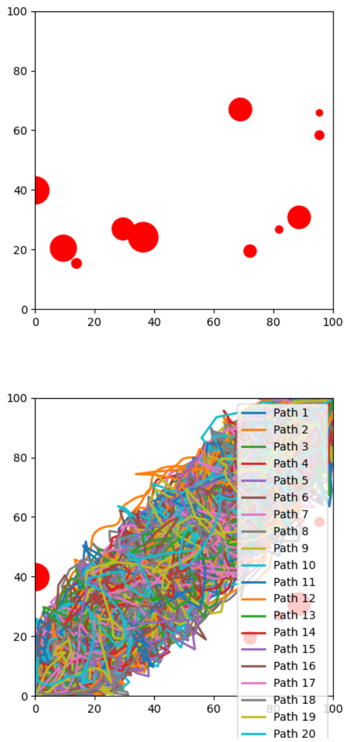

Fig. 5. Environment and Initial Paths for 500 Individuals.

The initial populations displayed minor collision avoidance, though due to the nature of the random path generation, some paths resulted in collisions.

Initially, Objective 1 = Path Distance and Objective 2 = Inverse of Path Smoothness. With these objectives, the three populations underwent processing through the evolutionary algorithm.

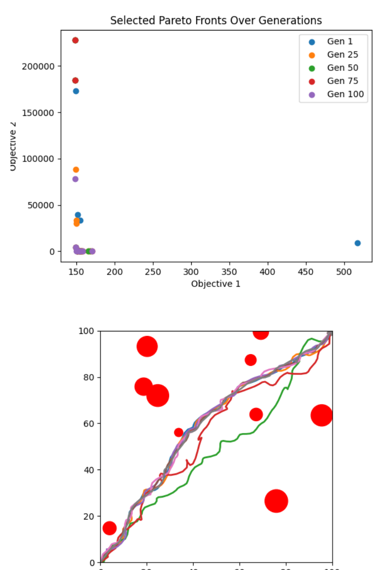

Fig. 6. Pareto Plot and Final Paths for 16 Individuals (Smoothness Optimized).

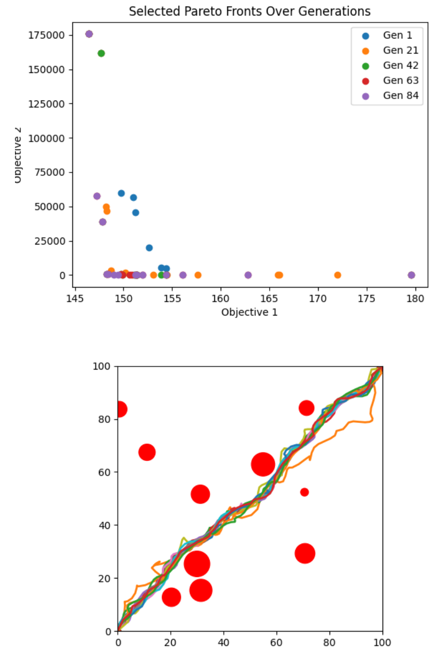

Fig. 7. Pareto Plot and Final Paths for 100 Individuals (Smoothness Optimized).

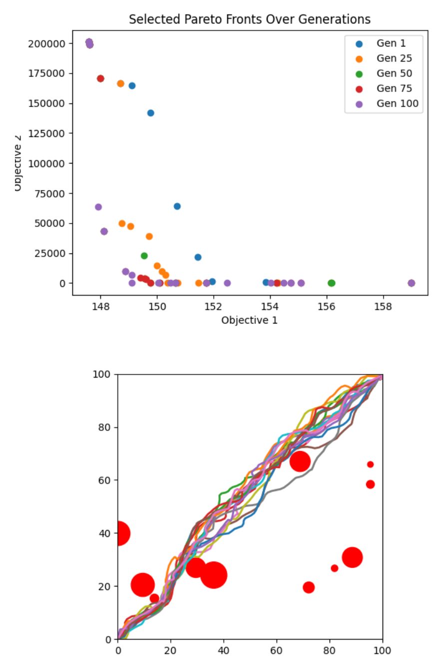

Fig. 8. Pareto Plot and Final Paths for 500 Individuals (Smoothness Optimized).

The Pareto plots from the evolutionary algorithm's output reveal that with increasing population size, more defined Pareto plots were observed. However, a distinct Pareto curve does not emerge in all cases. Furthermore, the resulting optimal paths appeared similar, suggesting a lack of contention between the two objectives, leading to sub-optimal multi-objective optimization.

To introduce more contending objectives, path smoothness is replaced with path safety (maximizing the minimum distance from obstacles) while keeping path distance as a primary objective (Objective 1 = Path Distance and Objective 2 = Inverse of Safety Distance).

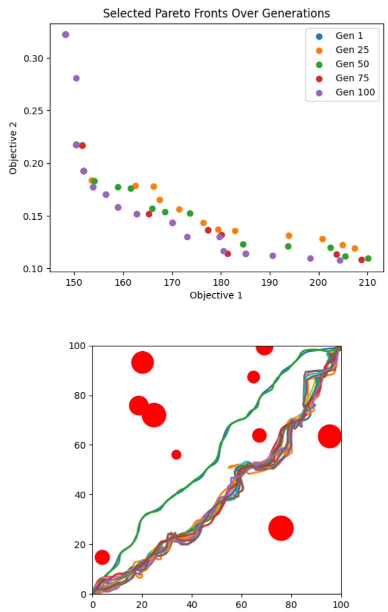

Fig. 9. Pareto Plot and Final Paths for 16 Individuals (Safety Optimized).

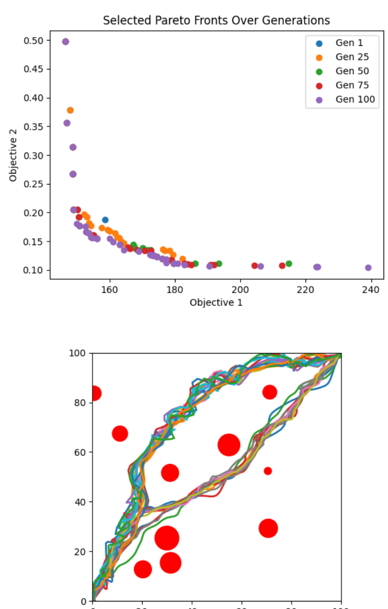

Fig. 10. Pareto Plot and Final Paths for 100 Individuals (Safety Optimized).

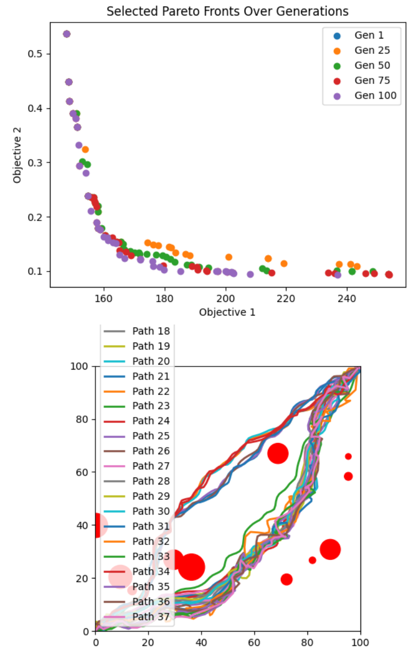

Fig. 11. Pareto Plot and Final Paths for 500 Individuals (Safety Optimized).

Upon analyzing the evolution with path safety as an objective, distinct Pareto fronts are evident across all population sizes, as seen by the curves in Figures 9 - 11. The fronts improve across generations, reflecting the inherent trade-off between path distance and safety and effectiveness of the evolutionary process.

VI. Discussion

Viewing the results of the analysis, the outcome confirms that the evolutionary algorithm was effectively navigating the conflicting objectives, as shorter paths often compromise safety. Despite varying population sizes, the emergence of distinct Pareto fronts across them underscored the effectiveness of MOO in path planning.

With this, cases are shown for and against the use of MOO for path planning. In the case of optimizing distance and path smoothness, the repeated lack of a clear Pareto front and the groupings of solutions in the objective space highlight the unnecessariness of MOO in this situation. Although the evolutionary process will generate increasingly optimal solutions, due to the nature of both objectives being synchronized in how they are optimized, operations to explore the objective space and find a diverse set of optimal solutions are counter-productive. In contrast, in the case of optimizing distance and safety, both objectives work against eachother, which as demonstrated results in the evolution working properly to generate a diverse set of solutions with trade-offs between the objectives.

Going on, the analysis showcases the challenge of applying MOO and the importance of understanding the problem and mission requirements. Although for the environments tested in this analysis, distance and smoothness were not contentious, in another problem or workspace, they may be contentious. The same goes for distance and safety.

Looking beyond the Pareto fronts, an interesting observation in the optimal paths can be made. For when smoothness is optimized, we essentially see only one path produced, which is expected. However, when safety is optimized, although there is relatively more diversity in the output paths, there are still obvious groupings appearing. Clusters of similar paths within these fronts suggested a need for increased genetic diversity and potential refinements in the algorithm to prevent local optima convergence. This indicated opportunities for enhancing the algorithm, such as introducing more variability in mutations and breeding or adjusting environmental constraints to foster a broader range of solutions. Furthermore, the set hyper-parameters in the NURBS definitions simplify the implementation and could be tuned more as well. The results highlight the intricacies of balancing path planning trade-offs and underscore the potential for further optimization in the algorithm.

Although all technical questions within "Defining Path Characteristics" and "Combining Path Characteristics for Optimization" are addressed by the use of NURBS and the evolutionary strategy, they are not necessarily answered. The NURBS result shows promising results, but more analysis must be done into the limitations of using NURBS. Furthermore, although the evolutionary method produced increasingly optimal results, this analysis does not benchmark the results against other planners to check if the evolutionary strategy truly performs better than other multi-objective and even some single objective planners.

Additionally, while the study successfully implemented the MOO framework, it primarily operated in simulated environments without dynamic obstacles. This limitation points to the need for real-world testing or simulation, which is essential for a comprehensive evaluation of MOO strategies. Real-world scenarios, with their unpredictability and complexity, would provide valuable insights into the adaptability and robustness of the MOO framework.

VII. Conclusion

In this study, we have investigated the furthering of Multi-Objective Optimization for path planning. Our exploration has illuminated the potential of MOO in navigating the complexities of path planning in varied environments, laying a groundwork for more efficient, adaptive, and robust solutions in robotics and autonomous systems.

A key innovation in our approach is the implementation of interchangeable objective functions. This flexibility is crucial, allowing for adaptability to different scenarios as long as the objective functions are properly defined. This interchangeability enhances the system's versatility, enabling it to cater to a wide range of applications and environmental conditions

Despite the progress made in this study and analysis, there still remains areas to further explore. One such area is the integration of real-time sensor data. While not investigated in this study, real-time data integration is essential for enhancing the responsiveness and accuracy in dynamic environments. Addressing the challenges in data integration is critical to making MOO beneficial to application.

Another avenue for exploration is the investigation of alternative methods for path representation. While our study primarily utilized Non-Uniform Rational B-Splines, exploring other methods could provide new insights and solutions in MOO for path planning. This exploration might reveal more efficient or adaptable path planning strategies that could be applied in various contexts.

Looking ahead, there is a notable potential in increasing the dimensionality of our approach for 3D path planning and extending our optimization strategies to accommodate more than two objectives. This expansion would allow for a more comprehensive optimization process, particularly beneficial in complex environments where multiple factors must be simultaneously considered and optimized.

In conclusion, the research and analysis performed in this paper have contributed towards the application of MOO in real-world situations. The methodologies performed and insights gained have set the stage for further advancements in the field, encouraging more effective solutions for the challenges faced in applying MOO towards robotics and autonomous systems path planning.

References

[1] H. Ren, Y. Shi and Y. Qiao, "A multi-objective optimization model of robot path planning under different scenarios," *2023 IEEE 3rd International Conference on Power, Electronics and Computer Applications (ICPECA)*, 2023. doi: 10.1109/ICPECA56706.2023.10076142.

[2] V. S. Ajith and K. G. Jolly, "Hybrid Optimization Based Multi-Objective Path Planning Framework for Unmanned Aerial Vehicles," *Cybernetics and Systems*, vol. 54, no. 8, pp. 1397-1423, 2023. doi: 10.1080/01969722.2022.2157607.

[3] J. Jing, L. Zhang, C. Shen, and K. Zhang, "Research on Multi-modal Multi-objective Path Planning by Improved Ant Colony Algorithm," *Bio-Inspired Computing: Theories and Applications*, 2023. doi: 10.1007/978-981-99-1549-1_2.

[4] C. E. Cortés, D. Sáez, F. Milla, A. Núñez, and M. Riquelme, "Hybrid predictive control for real-time optimization of public transport systems' operations based on evolutionary multi-objective optimization," *Transportation Research Part C: Emerging Technologies*, vol. 18, no. 5, pp. 757-769, 2010. doi: 10.1016/j.trc.2009.05.016.

[5] N. Hohmann, M. Bujny, J. Adamy, and M. Olhofer, "Hybrid Evolutionary Approach to Multi-objective Path Planning for UAVs," in *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2021, pp. 1-8. doi: 10.1109/SSCI50451.2021.9660187.
[6] L. Piegl and W. Tiller, *The NURBS Book*, Springer Science & Business Media, 1996.
[7] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182-197, 2002. doi: 10.1109/4235.996017.

VIII. Additional Resources

For further insights into the resources associated with this study, please visit the GitHub repository:

https://github.com/ashwinraju8/MO-PATHWISE

This repository provides all the necessary materials for understanding and implementing of the "Multi-objective Optimization for PATHfinding With Intelligent System Evolution (MO-PATHWISE)" approach.