A Multi-Objective Genetic Algorithm Approach for Co-Optimizing Routing and Scheduling in Public Transit

Jorge Valdivia Padron
Tecnologico de Monterrey
Monterrey, Mexico
A00831133@tec.mx

Jorge Mario Cruz Duarte
Tecnologico de Monterrey
Monterrey, Mexico
jorge.cruz@tec.mx

Jesús Guillermo Falcón Cardona
Tecnologico de Monterrey
Monterrey, Mexico
jfalcon@tec.mx

ABSTRACT

Efficiency in designing routes and scheduling vehicles for public transportation is a multidimensional and highly complex problem. It is essential to address this issue to reduce operating costs and enhance service quality while balancing conflicting objectives. Existing methods tend to combine these objectives in a weighted manner to form a single objective, limiting the solution to a single outcome and making it challenging to balance different criteria. This study introduces a methodology based on Multi-Objective Evolutionary Algorithms to overcome the limitations of single-objective approaches. These algorithms are particularly suitable for handling conflicting objectives.

The methodology first addresses the design of optimal routes and then vehicle scheduling. Both are integrated into the optimization process, considering both operational costs and unserved start times. The solutions are validated using real data from transportation in Monterrey. The proposed methodology is validated using data on routes and schedules of public buses in Monterrey. The results aim to achieve a more sustainable and efficient public transportation system framework for decision makers, which would lead to an improvement in the quality of life for the residents of Monterrey.

KEYWORDS

Route Optimization, Vehicle Scheduling, Multi-Objective Evolutionary Algorithms

ACM Reference Format:
Jorge Valdivia Padron, Jorge Mario Cruz Duarte, and Jesús Guillermo Falcón Cardona. 2023. A Multi-Objective Genetic Algorithm Approach for Co-Optimizing Routing and Scheduling in Public Transit. In *Proceedings of Genetic and Evolutionary Computation Conference (GECCO '23).* ACM, New York, NY, USA, 7 pages. https://doi.org/10.475/123_4

1 INTRODUCTION

1.1 Background Information

Public transportation is the backbone of urban mobility. Its impact stretches beyond mere transportation, significantly influencing the economic vitality, social dynamics, and environmental sustainability of cities. Efficient routing and scheduling are paramount in these systems. Routing encompasses the strategic planning of transit paths to maximize coverage and accessibility within the urban landscape, ensuring that all key areas are well-served. Scheduling, on the other hand, involves the careful timing of each vehicle's journey, which directly influences passenger wait times and the overall frequency of service. Traditional methods in these areas, often reliant on manual calculations or semi-automated processes, face challenges in keeping pace with the dynamic nature of urban demands. These methods can result in overlapping routes, underserved areas, and inconsistent service intervals, compromising the reliability and effectiveness of public transit.

1.2 Problem Statement

The challenge of optimizing public transportation systems presents a multi-dimensional problem that significantly affects urban mobility. Key issues lie in the routing and scheduling of transit vehicles, two areas that are inherently interlinked and complex to manage. In many urban environments, these critical tasks are still handled with minimal technological aid, leading to inefficiencies and suboptimal outcomes. The routing aspect deals with assigning transport modes to designated routes, a crucial task for ensuring comprehensive urban coverage and efficient vehicle flow. However, a reliance on conventional planning methods can yield routes that are ill-suited to current urban layouts or evolving passenger needs, resulting in redundant services or notable gaps in coverage. Similarly, the scheduling aspect, which focuses on setting optimal times for each vehicle's route, is pivotal for maintaining service regularity and efficiency. Poorly managed schedules can lead to service overlaps, extended passenger wait times, and gaps in service, all of which diminish the system's overall efficiency. These challenges are further complicated by external factors such as fluctuating urban growth patterns, unpredictable traffic conditions, and changing passenger demands, underscoring the need for an advanced, adaptable solution that can effectively co-optimize both routing and scheduling.

1.3 Project Overview

This project introduces a novel approach to enhancing public transit systems through the use of a multi-objective genetic algorithm, aiming to optimize the routing and scheduling processes. The focus is on creating a model capable of simultaneously optimizing various aspects of public transportation. These include routing efficiency, which ensures comprehensive area coverage and streamlined paths; scheduling effectiveness, which aims to provide timely and regular service; and operational cost management, which seeks to optimize

GECCO '23, 2023, Location

ACM ISBN 123-4567-24-567/08/06…$15.00
https://doi.org/10.475/123_4

resource allocation. The model comprises two integral components: the Route Generator and the Scheduler. The Route Generator utilizes Yen's K Shortest Paths algorithm within a weighted graph framework, where nodes represent specific transit points characterized by unique attributes such as priority levels and zone types. These attributes are carefully selected to provide a comprehensive assessment of each route's quality, enabling the identification of multiple efficient paths. This approach ensures a broad spectrum of routing options are available, facilitating a holistic optimization process that considers more than just the shortest distance. The Scheduler component utilizes the Non-dominated Sorting Genetic Algorithm II (NSGA-II). This algorithm evaluates various factors, including time cost and vehicle utilization, continuously evolving through iterations to derive the most efficient scheduling solutions. A key focus is on minimizing operational costs, primarily by reducing the number of buses required. This approach reflects an understanding that fewer buses mean lower expenses in terms of fuel, maintenance, and staffing, thereby optimizing resource use and efficiency. By integrating the node attributes identified during the routing phase, the Scheduler strategically incorporates candidate routes into each individual within the NSGA-II framework. This process enables the discovery of not only the most efficient schedule and bus allocation but also the selection of high-quality routes for each scenario. The integration of these components aims to develop a framework for public transit systems that is not just efficient and cost-effective but also highly adaptable, and capable of responding adeptly to the dynamic demands of the urban environment.

1.4 Objectives

The primary objectives of this project are to enhance the efficiency, effectiveness, and sustainability of public transit systems through the use of a multi-objective genetic algorithm. Specifically, the project aims to:

- **Optimize Route Efficiency:** Develop an algorithmic approach to identify the most efficient routes for public transit, considering factors such as distance, traffic conditions, and passenger demand, to reduce travel time and enhance service reliability.
- **Improve Scheduling Effectiveness:** Create a dynamic scheduling system that adapts to real-time changes and optimizes vehicle allocation and timetables, leading to better service frequency and reduced wait times for passengers.
- **Minimize Operational Costs:** Reduce the operational costs associated with public transportation, including fuel consumption, vehicle maintenance, and staffing, by utilizing resources more effectively and efficiently.
- **Enhance User Satisfaction:** Reduce overall operational expenses in public transportation. This goal is achieved by optimizing the number of vehicles in use, adhering to the principle that fewer vehicles equate to lower overall costs. Additionally, the model emphasizes the selection of shorter, more efficient routes. This not only enhances route efficiency but also indirectly contributes to cost savings by reducing variables such as fuel usage and travel time.
- **Promote Environmental Sustainability:** Contribute to environmental conservation efforts by optimizing public transit operations to lower emissions and reduce the overall carbon footprint of urban transportation.
- **Support Equitable Transit Access:** Ensure that the transit system serves diverse urban areas equitably, providing accessible and efficient transportation options to all segments of the population.
- **Facilitate Data-Driven Decision Making:** Utilize the collected data and insights from the model to inform future urban planning and public transit development strategies.

1.5 Hypothesis

The application of multi-objective genetic algorithms to transit scheduling problems can significantly improve the efficiency and quality of transit services by optimizing routes, timetables, and resource allocation, ultimately leading to reduced travel times, reduced operational costs, and improved passenger satisfaction

2 LITERATURE REVIEW

2.1 Traditional Optimization Methods in Public Transit

Traditional approaches in public transit optimization often treat routing and scheduling as a single objective problem, usually employing linear or combinatorial optimization techniques. A notable example is the work by Knut et al. [4], who developed a set partitioning model with side constraints for simultaneous vehicle and crew scheduling, focusing on minimizing total operational and fixed costs. This model was resolved using a branch-and-bound procedure embedded with a column generation method. Similarly, Lin et al. [5] proposed a bi-level multi-objective programming model for bus crew and vehicle scheduling, reflecting the hierarchical nature of decision-making in public transit systems.

2.2 Integrated Transit Network Models

The study by Almasi et al. [1] presents a comprehensive framework for urban transit network optimization in Daejeon, Korea. Their approach, targeting both fixed and variable demands, integrates bus transit systems with existing railway networks to satisfy multiple objectives. The methodology includes the generation of initial candidate route sets, network analysis for transit trips and service frequencies, and the application of metaheuristic optimization algorithms.

2.3 Pathfinding Algorithms in Network Design

Dijkstra's algorithm, introduced by Edsger W. Dijkstra in 1959, is a foundational algorithm in graph theory used for finding the shortest path between nodes in a graph [3]. It serves as a building block for many other pathfinding algorithms, including Yen's K Shortest Paths algorithm. Yen's algorithm, as proposed by Jin Y. Yen [7], extends the concept of shortest pathfinding by identifying multiple efficient paths in a network.

2.4 Multi-Objective Optimization in Vehicle Scheduling

Chen and Zuo's study [6] addresses the vehicle scheduling problem in urban bus lines through an advanced multi-objective genetic algorithm, specifically utilizing an enhanced version of Deb's Non-dominated Sorting Genetic Algorithm II (NSGA-II) [2].

The study's approach begins with generating a comprehensive set of candidate vehicle blocks for each start time in a bus line's timetable. These blocks, representing sequences of trips, form a versatile pool from which scheduling solutions can be constructed. The NSGA-II algorithm is then employed to optimize key objectives: minimizing the number of vehicles and drivers required, and reducing the incidence of uncovered start times in the schedules.

Solutions are represented as integer-valued vectors. Each gene in these vectors corresponds to an initial start time, with the gene's value indicating a chosen vehicle block. The NSGA-II algorithm's process involves several stages, such as initialization, crossover, mutation, and selection. These stages collectively contribute to the evolution of the population towards more efficient and effective scheduling solutions [6].

3 METHODOLOGY

3.1 Overview

Addressing the gap in existing public transit optimization models, this research proposes a methodology that co-optimizes routing and scheduling. By integrating these two aspects into a unified framework, the study aims to enhance both operational efficiency and service quality.

3.2 Routing

3.2.1 Graph Representation. In our proposed approach, the transportation network is conceptualized as a weighted graph, denoted by $G = (V, E, W)$, where V represents the set of nodes, E represents the set of edges, and W is a function assigning weights to the edges.

Each node $v \in V$ corresponds to a specific point within the transportation network, such as a bus stop or a train station. The nodes are uniquely identified by their IDs.

An edge $e \in E$, represented as a tuple (v_i, v_j), connects two nodes v_i and v_j in the graph. The edges signify possible paths or routes between the nodes in the transportation network.

The weight function $W : E \rightarrow \mathbb{R}^+$ assigns a positive real number to each edge, representing the cost, distance, or time required to travel from one node to the other. The weight of an edge (v_i, v_j) is denoted by $W(v_i, v_j)$.

The `Graph` class maintains a collection of nodes and edges. Nodes are stored in a dictionary keyed by their IDs for efficient access. Edges are stored in a list, each represented by an `Edge` object encapsulating the start node, end node, and the weight of the edge. In addition, the class provides functionalities to temporarily remove or restore nodes and edges from the graph.

A method to calculate the total cost of a given path is included. For a path $P = [v_1, v_2, \ldots, v_n]$, the total cost is computed as the sum of weights of the consecutive edges in the path, given by $\sum_{i=1}^{n-1} W(v_i, v_{i+1})$.

3.2.2 Initialization. The initialization phase of the routing algorithm involves setting up the graph $G = (V, E, W)$ and preparing for the execution of Yen's K Shortest Paths algorithm. Nodes (V) and edges (E) are added to G using `add_node` and `add_edge` functions. Each node is identified by a unique ID, while edges store start and end node identifiers along with the weight (W) signifying the cost or distance.

3.2.3 Dijkstra's Algorithm. Dijkstra's algorithm is a core component of the routing process, applied within Yen's K Shortest Paths to identify the optimal paths in the transportation network graph $G = (V, E, W)$ [3]. The algorithm operates as follows:

(1) Initialize all node distances in G to infinity (∞), except for the start node, which is set to zero.
(2) Employ a priority queue to manage nodes based on their current shortest distance.
(3) Repeatedly extract the node with the smallest distance from the queue and update the distances of its adjacent nodes.
(4) For each edge e starting from the current node, calculate the distance as the sum of the current node's distance and edge weight. Update the distance of the edge's end node if the new distance is smaller.
(5) Maintain a record of preceding nodes for each node to reconstruct the shortest path upon completion.
(6) Continue this process until the priority queue is empty.
(7) Construct and return the shortest path from the start node to the end node, ensuring the path begins with the start node.

3.2.4 Yen's K Shortest Paths Algorithm. Yen's K Shortest Paths Algorithm is an extension of Dijkstra's algorithm, tailored to find multiple shortest paths in a graph. This algorithm is pivotal in the route generation phase, determining the K shortest paths between a pair of nodes in the transportation network graph $G = (V, E, W)$ [7]. The process is as follows:

(1) Begin by finding the shortest path between the start and end nodes using Dijkstra's algorithm. Denote this path as the initial path and add it to the set A of shortest paths.
(2) For each path p in A, and for each node n in p, execute the following steps:
  (a) Define a root path, which is a sub-path of p ending at node n.
  (b) Temporarily remove all nodes in the root path, except for n, from graph G. This prevents repetition of the root path in subsequent spur paths.
  (c) For each path in A that shares the same root path, remove the edges leading out of node n in these paths from G. This step is crucial for discovering new spur paths.
  (d) Apply Dijkstra's algorithm to find a spur path from node n to the end node in the modified graph.
  (e) Restore the removed nodes and edges to G to maintain the graph's integrity for subsequent iterations.
  (f) If a valid spur path is found, concatenate it with the root path to form a new candidate path. Add this new path to the set B of potential paths if it is not already in A or B.
(3) Select the shortest path from B and move it to A. Repeat this process until K shortest paths are identified or no more new paths can be found.

3.2.5 Path Selection and Restoration. The Path Selection and Restoration process in Yen's K Shortest Paths algorithm plays a critical role in ensuring the integrity of the graph and the diversity of the paths generated. This section of the algorithm focuses on managing and restoring the state of the graph after each iteration. The steps are as follows:

(1) **Path Selection:** After the generation of spur paths, a crucial step is the selection of the most optimal path from the candidate set B. This step involves:
 (a) Comparing the total cost of each path in B, which is the sum of the weights of the edges comprising the path.
 (b) Selecting the path with the lowest total cost and moving it from B to A, the set of K shortest paths.

(2) **Graph Restoration:** To maintain the original structure and properties of the graph for subsequent iterations, it is essential to restore the nodes and edges temporarily removed during the spur path generation. This involves:
 (a) Reintegrating all nodes and edges previously removed. This step ensures that the graph returns to its original state, allowing for the accurate generation of new paths in the next iterations.
 (b) Ensuring the integrity of the graph is crucial for the algorithm's correctness, as any alterations might lead to inaccurate or suboptimal path generation in subsequent iterations.

3.2.6 Path Cost Calculation. The Path Cost Calculation is a vital component of Yen's K Shortest Paths algorithm, as it determines the efficiency of each path generated. This subsection describes the method used to calculate the total cost of paths. The process is outlined as follows:

(1) For a given path, the total cost is calculated by summing the weights of all the edges that constitute the path. This is expressed as:

$$\text{Cost}(Path) = \sum_{i=1}^{n-1} w(e_{i,i+1}) \tag{1}$$

where $Path$ is a sequence of nodes $[node_1, node_2, \ldots, node_n]$, $e_{i,i+1}$ represents the edge from $node_i$ to $node_{i+1}$, and $w(e_{i,i+1})$ is the weight of edge $e_{i,i+1}$.

(2) The algorithm iteratively calculates the cost for each path generated in the set A of K shortest paths. For each path P_k in A, the total cost is computed using the above formula, ensuring that the cost reflects the sum of weights of the edges in P_k.

3.3 Scheduling

3.3.1 Individual Representation. In the proposed scheduling algorithm, each solution is encapsulated in an *Individual* class. An individual is characterized by a set of 'blocks', representing the vehicle schedules, and a 'route', which is a sequence of node IDs indicating the path taken by a transit vehicle. The representation is inspired by Chen and Zuo's study [6].

The fitness of an individual, stored in the 'fitness' attribute, is evaluated based on several objectives: the number of uncovered start times, the number of blocks (correlating to the number of vehicles or drivers), and the cost of the route. The route cost is calculated as $C(\text{route}) = \sum_{i=1}^{N}(w_1 \cdot \text{Priority}(n_i) + w_2 \cdot \text{ZoneType}(n_i))$, where N is the number of nodes, n_i is the i^{th} node, and w_1, w_2 are weights for the node's priority and zone type, respectively.

The 'front' and 'crowding-distance' attributes of the individual are crucial for the multi-objective optimization process. The 'front' attribute indicates the individual's rank based on the non-domination sorting principle, and 'crowding-distance' measures its diversity in relation to other individuals in the same front.

3.3.2 Population Initialization. A population in this context is a collection of 'Individual' instances, each representing a potential solution to the scheduling problem. The initialization process is designed to create a diverse set of solutions, essential for the exploration of the multi-dimensional solution space.

Each individual is initialized with a random set of vehicle blocks and a randomly selected route from the precomputed K-shortest paths. The vehicle blocks, crucial for defining the schedule, are generated through a function $_{\text{generate\_random\_block}}(T, L)$, where T is the target schedule, and L is the maximum block length. This function ensures that the individuals have a varied and realistic set of schedules, mirroring the stochastic nature of real-world transit operations. The route component of each individual, derived from the K-shortest paths, adds a spatial dimension to the scheduling problem.

The initial population size is set to ensure adequate coverage of the solution space while balancing computational efficiency. The individuals in this initial population form the basis for subsequent generations, evolving through selection, crossover, and mutation operations to explore and exploit the solution space effectively.

3.3.3 Objective Function Evaluation. The first objective, O_1, focuses on minimizing the number of uncovered start times in the schedule. It is calculated by counting the times in the target schedule that are not covered by the individual's blocks. Mathematically, it is expressed as:

$$O_1 = \sum_{t \in T} \left[1 - \text{Ind}_{\{\text{Covered\_Start\_Times}(t)\}}\right]$$

where T is the target schedule, and Ind represents the indicator function, which is 1 if t is covered and 0 otherwise.

The second objective, O_2, aims to minimize the number of blocks (drivers) used in the schedule. It directly reflects the operational costs and is given by:

$$O_2 = \text{length}(\text{Blocks})$$

The third objective, O_3, evaluates the cost of the selected route based on node attributes. This cost is a function of the route's nodes, incorporating factors such as priority levels and zone types. The route cost for an individual i with a route R_i is computed as:

$$O_3(i) = \sum_{n \in R_i} \text{Cost}(n)$$

where $\text{Cost}(n)$ is the cost function based on node attributes.

3.3.4 Selection, Crossover, and Mutation. The evolutionary process of the NSGA-II algorithm is propelled by the mechanisms of selection, crossover, and mutation, which collectively refine the population over generations [2].

Selection: The Binary Selection Tournament is employed to select individuals for the next generation. This process involves randomly choosing pairs of individuals and selecting the one with a better front (lower rank) or, in the case of a tie, the one with a higher crowding distance. For two individuals I_1 and I_2, the selection criterion is defined as:

$$\text{Select}(I_1, I_2) = \begin{cases} I_1, & \text{if front}(I_1) < \text{front}(I_2) \\ & \text{or } (\text{front}(I_1) = \text{front}(I_2) \\ & \text{and distance}(I_1) > \text{distance}(I_2)) \\ I_2, & \text{otherwise} \end{cases}$$

Crossover: The crossover operator generates new offspring by combining parts of two parent individuals. For parents P_1 and P_2, the crossover point c is chosen randomly. The offspring O_1 and O_2 are then created as:

$$O_1.\text{blocks} = P_1.\text{blocks}[: c] + P_2.\text{blocks}[c :]$$
$$O_2.\text{blocks} = P_2.\text{blocks}[: c] + P_1.\text{blocks}[c :]$$

Mutation: Mutation introduces random changes to an individual's attributes, enhancing genetic diversity. For an individual I with a mutation probability p_m, the mutation may alter the blocks or route:

$$\text{Mutate}(I, p_m) = \begin{cases} \text{Add, Remove,} & \text{with probability } p_m \\ \text{or Modify block,} & \\ \text{Change route,} & \text{with probability } p_m \end{cases}$$

3.3.5 Fast Non-Dominated Sorting and Crowding Distance. Fast Non-Dominated Sorting (FNDS) plays a pivotal role in ranking individuals based on the concept of Pareto dominance. In our NSGA-II implementation, each individual I in the population is evaluated against others to determine Pareto dominance relationships. The process categorizes individuals into different *fronts* based on the number of solutions dominating them and the number they dominate. The first front (F_1) contains individuals that are not dominated by any other, while subsequent fronts are populated iteratively.

For each individual I, we define:

- domination_count(I): The number of individuals that dominate I.
- dominated_solutions(I): The set of individuals dominated by I.

Individuals in the same front are then subjected to Crowding Distance assessment, a crucial step in maintaining diversity in the population. This metric quantifies the density of individuals surrounding a particular individual in the objective space. A larger crowding distance implies an individual is located in a less crowded region, which is desirable for preserving diversity.

For each front F_i, the Crowding Distance $d(I)$ for an individual I is calculated as follows:

$$d(I) = \sum_{m=1}^{M} \frac{f_m^{(I+1)} - f_m^{(I-1)}}{f_m^{\max} - f_m^{\min}} \tag{2}$$

where M is the number of objectives, $f_m^{(I)}$ is the m-th objective value of individual I, and $f_m^{\max}, f_m^{\min}$ are the maximum and minimum values of the m-th objective in F_i.

3.3.6 Evolutionary Process and Generation Formation. The evolutionary process in NSGA-II is a cyclical algorithm where each cycle represents a generation. This iterative process involves selection, crossover, mutation, and the formation of a new generation through a systematic approach. The primary aim is to evolve the population towards an optimal set of solutions that represent the best trade-offs among the objectives.

The formation of each generation is based on a sequential progression of steps outlined as follows:

(1) **Initialization**: A population P_0 of N individuals is generated, where each individual is randomly initialized with a feasible solution.
(2) **Evaluation**: For each individual in P_0, the objective function values are calculated. These values are used to perform Fast Non-Dominated Sorting and to assign a front number to each individual.
(3) **Selection**: A selection mechanism, such as Binary Selection Tournament, is employed to choose individuals for mating. This step ensures the propagation of superior genes.
(4) **Crossover and Mutation**: Genetic operators—crossover and mutation—are applied to the selected individuals to create offspring. These operations introduce new genetic material and variations into the population.
(5) **Formation of Next Generation**: The combined population of parents and offspring is sorted using Fast Non-Dominated Sorting. Crowding Distance is calculated to maintain diversity. The next generation is formed by selecting individuals based on their front rank and crowding distance.
(6) **Iteration**: This process is repeated for a predefined number of generations or until a termination criterion is met.

4 EXPERIMENT DESIGN

The experiment was designed to optimize the bus routing and scheduling for a university area in Monterrey, Nuevo Leon. The area was modeled as a graph comprising 23 nodes, where each node represented a significant location, and the edges symbolized the distances between these locations. This graph was utilized to simulate the university's existing bus route and to explore optimized alternatives.

Two specific nodes, Node 1 and Node 23, were selected as endpoints for the model. These nodes correspond to the start and end points of the university's current bus route. The optimization was conducted separately for two distinct schedules:

(1) Schedule from Node 1 to Node 23.
(2) Schedule from Node 23 to Node 1.

Initially, Yen's K Shortest Paths Algorithm was employed to determine a set of optimal routes based on the shortest distance criteria. The obtained routes were then inserted into the NSGA-II algorithm, alongside the university's operational schedules. The algorithm aimed to find the Pareto optimal solutions for each direction of the route, focusing on minimizing unserved start times, the number of buses required, and the overall cost of the route.

5 RESULTS

5.1 Optimization Results

The NSGA-II algorithm was applied separately for the two distinct schedules: from Node 1 to Node 23 and from Node 23 to Node 1. The results were evaluated based on the Pareto fronts generated for each schedule, emphasizing the trade-offs between the objectives: unserved start times, number of buses used, and route cost.

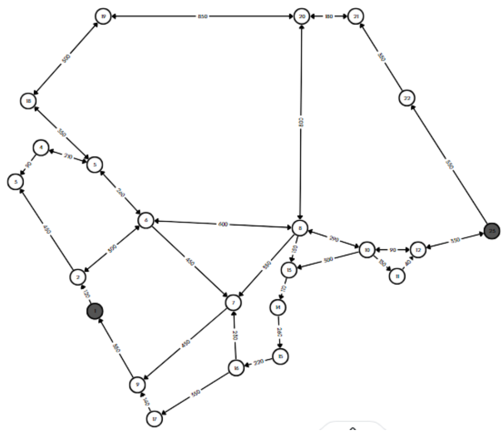

Figure 1: Graph of the selected area.

5.1.1 Schedule from Node 1 to Node 23. For the schedule from Node 1 to Node 23, the best individual exhibited the following characteristics:

- **Blocks:** A sequence of start times for each bus block. Example: $[555, 570, \ldots, 660]$, $[420, 435, \ldots, 525]$, ...
- **Fitness:** $[1, 10, 2.9]$ indicating 1 unserved start time, using 10 buses, and a route cost of 2.9.
- **Route:** $[1, 2, 6, \ldots, 23]$, with a total distance of 1800 meters.

Remarkably, the algorithm matched the current university route distance of 1800 meters while optimizing other aspects.

5.1.2 Schedule from Node 23 to Node 1. The best individual from the Pareto front for the schedule from Node 23 to Node 1 demonstrated notable efficiency:

- **Blocks:** A sequence of start times for each bus block. Example: $[866, 881, \ldots, 971]$, $[551, 566, \ldots, 656]$, ...
- **Fitness:** $[0, 10, 3.8]$ representing 0 unserved start times, utilizing 10 buses, and a route cost of 3.8.
- **Route:** $[23, 22, 21, \ldots, 1]$, covering a total distance of 3030 meters.

This optimized route showed a reduction in total distance compared to the existing university route of 3300 meters.

The results are visualized using Pareto front graphs, illustrating the trade-offs between the different objectives.

5.2 Comparative Analysis

The optimized routes derived from the NSGA-II algorithm provided significant improvements over the current university routes. The reduction in total distance for the route from Node 23 to Node 1 and the matching distance for the route from Node 1 to Node 23, along with improved bus scheduling and reduced route costs, underscore the effectiveness of the multi-objective optimization approach.

5.3 Resources and Technical Specifications

The complete source code for this project is hosted in a public repository, which can be accessed for detailed review and replication purposes at https://github.com/valjor98/MOGA-RS-PublicTransit.

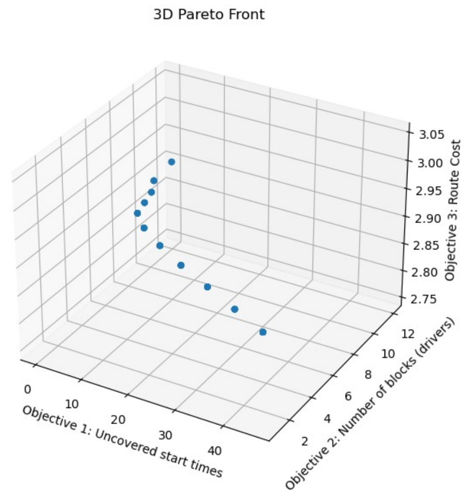

Figure 2: Pareto front for the first route.

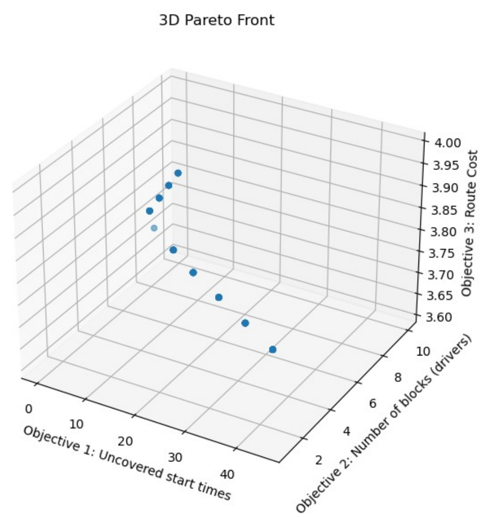

Figure 3: Pareto front for the second route.

All computational experiments and the execution of algorithms were carried out on a dedicated computing system to ensure reliability and efficiency. The system used for this study was equipped with an Intel Core i3-10110U processor, 8GB RAM, and ran on Windows 11-64 bit operating system. The algorithms were developed and tested using Python 3.8.

6 CONCLUSIONS

6.1 Discussion

The results from the optimization model applied to the Monterrey Nuevo Leon university area provide insightful observations. Notably, the model's effectiveness in optimizing bus routes for a localized area was demonstrated. The approach, employing a multi-objective genetic algorithm, successfully identified routes that are efficient in terms of operational cost, vehicle usage, and route cost.

The study revealed that the optimized routes, for both Node 1 to Node 23 and vice versa, not only match the existing university route in terms of coverage but also offer improvements in terms of operational efficiency. Specifically, the optimized route from Node 23 to Node 1 showed a notable reduction in the total route distance compared to the current university route, which is a significant achievement in urban transit planning.

Moreover, the application of this model in a real-world scenario underlines its potential in addressing similar challenges in urban transit systems. The findings suggest that such an approach can be effectively used to enhance public transport services, optimize resources, and reduce environmental impact through more efficient route planning.

6.2 Contributions

This study presents significant advancements in urban transit optimization through a novel multi-objective genetic algorithm. Key contributions include:

- Development of an optimized route planning model for a university bus system in Monterrey Nuevo Leon, effectively addressing multiple objectives like minimizing unserved start times, optimizing the number of buses, and reducing route costs.
- Application of the model to a real-world scenario, demonstrating its potential to improve operational efficiency and reduce costs in urban transit systems.
- Contribution to sustainable urban transportation by optimizing routes and vehicle usage, supporting efforts to lower carbon emissions and environmental impact.

6.3 Conclusions

The application of a multi-objective genetic algorithm for optimizing the routing and scheduling of a university bus system in Monterrey Nuevo Leon has demonstrated considerable success. This study not only showcases the algorithm's ability to efficiently generate routes that minimize travel time and operational costs but also maximizes service coverage. Its potential for wider application in urban transit systems is evident, offering a robust and adaptable tool for enhancing the efficiency of public transportation. Moreover, by optimizing resource utilization, the model aligns with urban sustainability goals, potentially reducing the environmental impacts of public transit systems. In essence, this research marks a significant contribution to the urban transit optimization field, presenting a versatile and effective solution to complex routing and scheduling challenges in public transportation.

REFERENCES

[1] Mohammad Hadi Almasi, Yoonseok Oh, Ali Sadollah, Young-Ji Byon, and Seungmo Kang. 2020. Urban transit network optimization under variable demand with single and multi-objective approaches using metaheuristics: The case of Daejeon, Korea. *International Journal of Sustainable Transportation* (2020). https://doi.org/10.1080/15568318.2020.1821414

[2] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6, 2 (2002), 182–197.

[3] Edsger W Dijkstra. 1959. A note on two problems in connexion with graphs. *Numerische mathematik* 1, 1 (1959), 269–271.

[4] H Knut, D Guy, and D Jacques. 2001. Simultaneous vehicle and crew scheduling in urban mass transit systems. *Transportation Science* 35, 3 (2001), 286–303.

[5] Y Lin, S Pan, L Jia, and N Zou. 2010. A bi-level multi-objective programming model for bus crew and vehicle scheduling. In *2010 International Conference on Intelligent Control and Automation (WCICA)*. IEEE, 2328–2333.

[6] Chunlu Wang, Hongyi Shi, and Xingquan Zuo. 2020. A multi-objective genetic algorithm based approach for dynamical bus vehicles scheduling under traffic congestion. *Swarm and Evolutionary Computation* 54 (2020), 100667. https://doi.org/10.1016/j.swevo.2020.100667

[7] Jin Y Yen. 1970. An algorithm for finding shortest routes from all source nodes to a given destination in general networks. *Quart. Appl. Math.* 27 (1970), 526–530. https://doi.org/10.1090/qam/253822