Doctoral Dissertation

A New Hybrid Framework for Preference-Aware Multi-Objective Vehicle Routing Problem with Time Windows and Demand Priority

Thanapat Leelertkij

Supervisor: Van Nam Huynh

Graduate School of Advanced Science and Technology
Japan Advanced Institute of Science and Technology
[Knowledge Science]

March 2026

Abstract

This dissertation addresses a key gap in real-world logistics optimization by proposing the Multi-Objective Vehicle Routing Problem with Time Windows and Demand Priority (MO-VRPTWDP). Unlike traditional VRP models that focus solely on cost minimization, this new formulation incorporates customer satisfaction via a weighted waiting time mechanism, enabling more equitable and service-oriented routing, especially in domains such as healthcare logistics and premium delivery.

To address this problem across different scales and decision-making contexts, the research can be divided into 3 parts. First, a Mixed Integer Linear Programming (MILP) model is developed to validate the formulation and analyze small-scale solution behavior. It captures trade-offs between operational cost and service levels based on customer priority.

Second, a novel Multi-Thread Simulated Annealing (MTSA) algorithm is proposed to enhance scalability and exploration. MTSA introduces parallel threads and cooperation among them, significantly improving the diversity and quality of Pareto frontier approximations. Experiments show that MTSA outperforms the benchmark algorithm (MOSA).

Third, a reinforcement learning–based extension, RL-MTSA, is introduced to enable preference-aware optimization. By embedding a learning agent into the MTSA algorithm, RL-MTSA dynamically steers the search toward user-specified regions of interest. It achieves faster convergence and higher alignment with decision-maker preferences than uniform-search methods.

Overall, this research contributes a new VRPTW variant with soft-priority modeling, scalable optimization techniques, and adaptive, user-preference search strategies. The proposed methods offer practical decision support tools for the preference-aware optimization in the multi-objective vehicle routing problem.

Keywords: Vehicle Routing Problem, Multi-Objective Optimization, Simulated Annealing, Reinforcement Learning, Preference-aware Optimization

Acknowledgment

I would like to express my deepest gratitude to my supervisor, Prof. Huynh Van-Nam, for his continuous support and insightful guidance throughout the course of this research. From the very beginning, he has played a vital role in shaping this dissertation, offering valuable comments, overseeing the research progress, and assisting with manuscript revisions for journal submission. I am also especially thankful for his generous support in helping me prepare the documents necessary for my successful application to the MEXT scholarship, which has enabled me to pursue my graduate studies at JAIST with full financial support.

I am sincerely grateful to Assoc. Prof. Jirachai Buddhakulsomsiri, who has overseen this research from the very beginning and has been my trusted advisor since my undergraduate studies. His mentorship has been instrumental in guiding me through both academic and personal challenges. He not only introduced me to Huynh-sensei and encouraged me to apply for the MEXT scholarship, but also offered continuous encouragement, perspective, and unwavering support during every stage of this research, including during times of personal difficulty.

I would also like to thank Prof. Tsutomu Fujinami for his thoughtful suggestions and guidance on related research topics, which helped broaden the scope and technical rigor of this work.

I wish to thank my father and mother, Sompong Leelertkij and Suree Leelertkij, who have worked tirelessly for more than 30 years to ensure that their children would have the opportunity to study at well-established institutions. Their sacrifice, dedication, and unconditional support have been the foundation of everything I have achieved.

Lastly, I would like to thank my friends for their constant support, encouragement, and companionship throughout this academic journey.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The Vehicle Routing Problem (VRP) is a foundational combinatorial optimization problem in logistics and transportation, first introduced by Dantzig and Ramser (1959). It involves determining the optimal set of routes for a fleet of vehicles that must deliver goods to a set of customers from a central depot. Each vehicle operates under constraints such as limited capacity, route feasibility, and scheduling requirements. The main objective is typically to minimize the total travel distance or cost while fulfilling all customer demands.

At its core, the VRP generalizes the well-known Traveling Salesman Problem (TSP), extending it to multiple vehicles and a broader set of constraints. Figure 1.1 illustrates a simple example of VRP with three delivery routes originating from a single depot and serving multiple customers. Each vehicle must start and end at the depot, and no customer should be visited more than once.

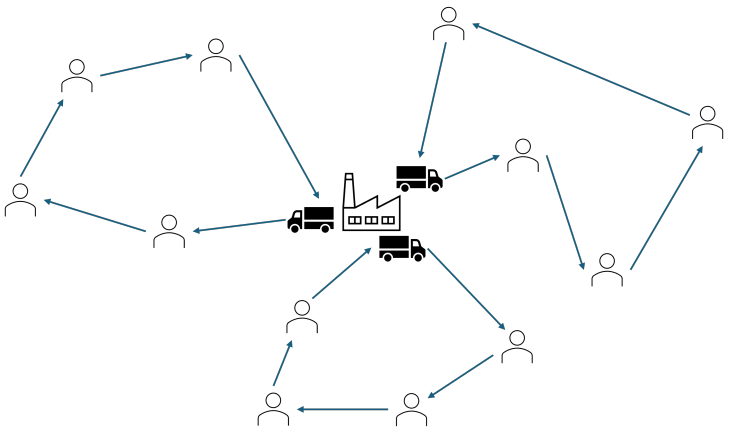

Figure 1.1: Example of a Vehicle Routing Problem (VRP) solution with three vehicle routes.

The VRP has become one of the most widely studied problems in operations research because of its direct relevance to real-world applications such as parcel de-

livery, ride-sharing, healthcare logistics, and supply chain management. At the same time, its complexity has made it a fertile ground for developing new optimization techniques, ranging from exact mathematical programming to modern metaheuristics and machine learning approaches. This dual character, practical importance and computational difficulty, provides the foundation for the present study.

1.1 Research Motivations

While the Vehicle Routing Problem (VRP) provides a unifying framework for many logistics and transportation applications, its practical deployment faces several limitations. Real-world distribution systems must handle not only the computational complexity of large-scale routing but also the need to reflect service quality considerations and deliver solutions aligned with stakeholder priorities. Addressing these issues requires both extending the classical VRP formulation and advancing the metaheuristic techniques used to solve it.

The motivation for this dissertation can be viewed from two perspectives: a broader research gap in the literature, and a sequential progression of solutions that emerged during the course of this study.

1. **Modeling service quality and priorities:** The classical Vehicle Routing Problem with Time Windows (VRPTW) ensures that all customers are served within specified time intervals, but does not explicitly consider service quality or differentiated priorities. This limitation is increasingly unrealistic in domains such as healthcare logistics, emergency response, and premium delivery services, where responsiveness is as critical as cost efficiency. To capture these practical concerns, this dissertation introduces the Vehicle Routing Problem with Time Windows and Demand Priority (VRPTWDP). A Mixed Integer Linear Programming (MILP) model of VRPTWDP enables the study of customer-priority effects in small-scale instances, but its computational complexity restricts its applicability to larger problems.

2. **Enhancing metaheuristic exploration:** To address scalability, metaheuristic approaches are required. Evolutionary algorithms such as NSGA-II have proven effective in generating diverse Pareto fronts through population-based exploration, but simulated annealing (SA), while strong in local convergence, suffers from limited diversity in multi-objective contexts. This creates an opportunity to enhance SA's exploration capabilities without losing its convergence strengths. To this end, we propose Multi-Thread Simulated Annealing

(MTSA), which leverages parallel search thread mechanisms to improve coverage of the Pareto frontier. Importantly, experiments with MTSA revealed that weight vectors not only distribute search effort across the frontier but also influence search behavior, hinting at the potential for adaptive, guided exploration.

3. **Supporting decision relevance through preference-aware search:** From a practical perspective, decision-makers are rarely interested in the entire Pareto frontier; they are interested in specific trade-offs aligned with strategic or policy goals. Existing solvers, however, typically explore the frontier uniformly. Building on the insight that weights can steer MTSA, we are motivated to integrate artificial intelligence to automate this process. Therefore, this dissertation introduces Reinforcement Learning–Multi-Thread Simulated Annealing (RL-MTSA), where reinforcement learning dynamically adjusts weights to guide the search toward user-specified regions of the Pareto frontier. This hybrid approach combines scalability, diversity, and preference-awareness, producing solutions that are both computationally efficient and decision-relevant.

In summary, this dissertation is motivated by both theoretical and practical gaps: (1) extending VRPTW to incorporate service quality and priorities, (2) designing scalable metaheuristics that enhance simulated annealing for multi-objective optimization, and (3) developing adaptive, preference-aware mechanisms that align optimization outcomes with stakeholder needs. The research progresses naturally from the VRPTWDP formulation to the scalable MTSA algorithm and finally to the reinforcement learning–guided RL-MTSA framework.

1.2 Research Objectives

This dissertation aims to advance the field of multi-objective vehicle routing by addressing limitations in classical VRPTW models, particularly the absence of mechanisms for incorporating customer priority and user preferences into scalable optimization frameworks. The research is structured into three integrated parts, including problem formulation, metaheuristic design, and preference-aware optimization, each with specific objectives as follows:

1. **Problem Formulation:** To develop a comprehensive mathematical model for the Multi-Objective Vehicle Routing Problem with Time Windows and

Demand Priority (MO-VRPTWDP). The formulation introduces a soft priority mechanism by incorporating weighted customer waiting time into a secondary objective, complementing the classical objective of minimizing total travel distance. This model aims to:

- Reflect real-world operational needs where differentiated service levels are important but rigid priority enforcement is impractical.
- Enable planners to balance efficiency with fairness by influencing route construction in favor of high-priority customers while maintaining feasibility under time window and capacity constraints.

Validation through small-scale computational experiments will be conducted to confirm the correctness of the formulation and to explore the behavioral impacts of priority settings on routing outcomes.

2. **Metaheuristic Development and Evaluation:** To design and evaluate a scalable metaheuristic algorithm based on simulated annealing principles, extended to handle multi-objective optimization for large-scale MO-VRPTWDP instances. This objective focuses on:

   - Extending classical simulated annealing into a multi-objective setting by maintaining a Pareto archive and employing dominance-based acceptance criteria.
   - Leveraging parallel search threads to explore different regions of the solution space to enhance diversity and convergence.
   - Overcoming the exploration limitations of traditional simulated annealing while preserving its strong local refinement capability.
   - Benchmarking the approach against established multi-objective algorithms to demonstrate its effectiveness in producing well-distributed, high-quality Pareto frontier approximations within reasonable computational times.

3. **Preference-Aware Optimization with Learning Integration:** To extend the metaheuristic with reinforcement learning, enabling dynamic, preference-aware optimization within the Pareto frontier for MO-VRPTWDP. This objective aims to:

   - Integrate a learning agent into the optimization loop to guide the search toward user-specified regions of the Pareto frontier, reflecting evolving operational preferences and decision-maker priorities.

- Concentrate computational effort on the most relevant regions of the solution space, improving the practical usability of generated solutions while maintaining diversity.
- Evaluate the impact of preference guidance on solution quality, convergence speed, and computational efficiency through experiments on benchmark instances.

Collectively, these objectives aim to deliver a complete methodological pipeline, advancing from rigorous problem formulation to scalable and adaptive optimization, thereby supporting user-aligned, multi-objective decision-making in complex vehicle routing environments.

1.3 Research Contributions

This dissertation advances the state of multi-objective vehicle routing by addressing gaps in classical VRPTW and its extensions through the development of a new problem formulation, scalable metaheuristic methods, and adaptive preference-aware optimization frameworks. Building upon the research objectives, these contributions demonstrate the feasibility, scalability, and practical relevance of the proposed methods and are structured according to the three main research components of this work.

1. **Formulation and Analytical Study of MO-VRPTWDP:** A novel problem formulation, the Multi-Objective Vehicle Routing Problem with Time Windows and Demand Priority (MO-VRPTWDP), is proposed to address the lack of mechanisms for explicitly incorporating customer service priority into VRP models. This formulation introduces soft customer priority through a weighted waiting time objective, enabling differentiation of customer importance without imposing hard sequencing constraints that could restrict operational flexibility. A Mixed Integer Linear Programming (MILP) model is developed to:
    - Precisely define the problem structure, constraints, and multi-objective functions in a manner compatible with exact and heuristic solvers.
    - Validate the feasibility and correctness of the proposed formulation through computational experiments on small-scale instances.
    - Systematically analyze the impact of customer priority weight settings on routing outcomes, providing quantitative insights into the trade-offs between total traveling distance and total customer waiting time.

This contribution extends the body of knowledge in VRPTW research by demonstrating how soft priority can be seamlessly integrated into a multi-objective framework, supporting practical decision-making in logistics operations where differentiated service levels are critical.

2. **Development of Multi-Thread Simulated Annealing (MTSA) for Scalable Optimization:** To address the computational challenges associated with solving MO-VRPTWDP on large instance scales, this dissertation proposes a novel metaheuristic: Multi-Thread Simulated Annealing (MTSA). MTSA extends classical Simulated Annealing by:

    - Incorporating a multi-thread architecture, where multiple parallel search threads explore different regions of the solution space simultaneously.
    - Retaining the exploitation strengths of SA for local refinement while addressing its traditional limitations in exploration and diversity.

    Through extensive computational experiments, MTSA is shown to produce high-quality Pareto frontier approximations with competitive or superior convergence properties and computational efficiency compared to benchmark MOSA. This contribution demonstrates how SA-based metaheuristics, when enhanced with cooperative parallelism, can become viable and effective tools for complex multi-objective VRP optimization.

3. **Integration of Reinforcement Learning for Preference-Aware Optimization (RL-MTSA):** Building upon the scalable search capabilities of MTSA, this dissertation further contributes by developing RL-MTSA, a reinforcement learning-augmented extension that enables preference-aware optimization within the multi-objective VRPTWDP framework. RL-MTSA introduces:

    - An integrated reinforcement learning agent that dynamically guides the metaheuristic search process toward user-specified regions of the Pareto frontier, adapting to evolving decision-maker preferences during optimization.
    - An adaptive learning mechanism that adjusts search trajectories based on preference-aligned rewards, reducing cognitive burden on decision-makers while improving solution relevance.
    - The ability to operationalize stakeholder preferences in a scalable optimization environment, supporting decision-making for complex logistics

planning.

Through comprehensive experiments, RL-MTSA is demonstrated to improve the efficiency of finding solutions aligned with user preferences while maintaining solution diversity and computational tractability. This contribution addresses a critical gap in existing VRP research, where most methods focus on uniform Pareto frontier approximation without integrating user-guided search in a systematic and scalable manner.

In summary, these contributions provide a complete methodological advancement for multi-objective vehicle routing, progressing from the theoretical development of priority-inclusive models, through scalable and effective optimization methods, to adaptive, user-aligned decision support. The proposed framework offers practical value for real-world logistics and distribution systems where service differentiation, computational efficiency, and responsiveness to stakeholder priorities are essential for sustainable and competitive operations.

Structure of the Dissertation

This dissertation is organized into seven chapters as follows:

- **Chapter 1** introduces the research background, motivation, objectives, and contributions.
- **Chapter 2** reviews the literature on VRPTW, multi-objective optimization, customer satisfaction modeling, and reinforcement learning in routing.
- **Chapter 3** presents the formal problem definition of MO-VRPTWDP, including the MILP model and analysis of its properties on a small problem instance.
- **Chapter 4** describes the design and implementation of the MTSA algorithm, along with computational experiments and performance evaluation on benchmark instances.
- **Chapter 5** details the RL-MTSA framework, including the reinforcement learning environment, agent design, and preference-aware optimization. Computational experiments are conducted to assess its effectiveness in targeted search.

- **Chapter 6** provides a discussion of the methodological roles, integration of the proposed approaches, and their practical implications.
- **Chapter 7** concludes the dissertation, summarizing key findings and outlining directions for future research.

The structure of this dissertation reflects the threefold architecture of the research. The first part, focused on problem modeling, is covered in Chapter 3 through the formulation of MO-VRPTWDP and the MILP model, along with associated computational analysis. The second part, concerning scalable multi-objective optimization, is addressed in Chapter 4 via the development and empirical evaluation of the MTSA algorithm. The third part, emphasizing preference-aware optimization, is presented in Chapter 5 through the introduction of RL-MTSA and its corresponding experiments.

Each chapter integrates both the method development and its computational validation, ensuring that theoretical contributions are grounded in practical performance. This alignment offers a coherent and systematic narrative from problem formulation to user-aligned optimization strategies.

To situate these contributions within the broader academic landscape, the next chapter surveys related work in VRPTW modeling, multi-objective metaheuristics, and preference-aware optimization.

Chapter 2

Background and Related Work

This chapter reviews the key concepts and previous research that support the three-fold methodology proposed in this dissertation. The review covers four major areas: priority modeling in vehicle routing, multi-objective optimization and metaheuristics, preference-based optimization strategies, and the integration of reinforcement learning into combinatorial problem solving. Each section concludes with identified research gaps that motivate the corresponding contributions in later chapters.

2.1 Vehicle Routing Problem with Demand Priority (VRPDP)

Incorporating customer or demand priority into the Vehicle Routing Problem (VRP) has emerged as a critical extension for reflecting service differentiation in real-world logistics. Depending on the context, such as healthcare delivery, emergency response, or premium logistics services, customers may exhibit varying levels of urgency or importance. To address this, priority modeling in VRP can be categorized into two broad types: **soft priority** and **hard priority**.

Soft priority models treat priority as a flexible, service-level indicator, where vehicles may occasionally visit lower-priority customers earlier if it leads to overall operational improvements, such as reduced travel cost. Hard priority models, on the other hand, impose strict service precedence rules, ensuring that high-priority customers are always served before lower-priority ones. This distinction is essential for designing routing systems that balance efficiency with fairness and responsiveness.

2.1.1 Soft Priority Models

Soft priority models have gained considerable traction in vehicle routing literature due to their flexibility and suitability for practical applications where strict service ordering is neither feasible nor desirable. These models enable planners to incorporate customer importance without rigidly enforcing sequencing, allowing operational trade-offs between service equity and efficiency.

One early contribution is the work of Nucamendi-Guillén et al. (2020), who proposed a multi-objective VRP formulation that minimizes both total travel distance and customer tardiness. In their framework, tardiness is not defined relative to fixed due dates, but rather as a function of customer priority, penalizing cases where lower-priority customers are serviced earlier than higher-priority ones. This formulation encourages the routing algorithm to defer service for less important customers unless such deferrals conflict with cost objectives or route feasibility. The priority mechanism is embedded directly into the fitness evaluation, providing a seamless way to model soft service differentiation.

Several other studies combine soft priority with time window constraints, resulting in nuanced models of customer satisfaction. For example, Ghannadpour (2019) presented a multi-objective model that seeks to minimize energy consumption and fleet size while maximizing customer satisfaction. Satisfaction is modeled using two nested time windows: a tight preferred window and a broader acceptable window. Full satisfaction is granted when service occurs within the preferred window, while partial satisfaction is awarded for service within the broader window. This formulation enables a graded notion of service quality, balancing efficiency with customer experience.

Baradaran et al. (2019) extended this concept by introducing stochastic demand and allowing customers to define multiple time windows with associated priority levels. Their model minimizes a weighted sum of penalties for serving customers outside their higher-priority windows. This approach is particularly relevant for uncertain environments, such as on-demand delivery, where demand fluctuations and scheduling complexity prevent strict priority enforcement. Similarly, Beheshti et al. (2015) addressed VRPs with multiple prioritized time windows and proposed minimizing both total travel cost and the cumulative rank of served windows, thereby allowing the optimization algorithm to balance routing efficiency with responsiveness to priority levels.

Soft priority mechanisms have also been explored in multimodal delivery systems. Das et al. (2020) examined a synchronized truck-drone delivery model that incorporates service-level constraints based on customer priority. In their model,

high-priority customers are serviced using drone-truck coordination strategies that reduce delivery latency and increase reliability. The priority mechanism is implemented via service-level indicators, such as the probability of on-time service, which are integrated into the optimization process as soft constraints or objective components.

Collectively, these studies illustrate that soft priority can be implemented through a variety of mathematical constructs: weighted objectives, penalty functions, satisfaction scores, or service-level constraints. This versatility makes soft priority modeling suitable for a broad range of real-world logistics applications, including last-mile delivery, e-commerce, healthcare logistics, and subscription-based services. Moreover, soft priority frameworks allow decision-makers to explore trade-offs between fairness, cost, and timeliness, enabling context-sensitive routing strategies that would not be possible under rigid priority enforcement.

The proposed MO-VRPTWDP in this dissertation builds upon this body of work by incorporating soft priority through a weighted waiting time objective. Unlike satisfaction-based formulations, the waiting time approach provides a direct and interpretable means of prioritizing customers based on their delay from the earliest allowable service time. The use of priority weights ensures that higher-priority customers contribute more heavily to the objective, thereby influencing route construction in their favor without introducing infeasible hard constraints. This mechanism facilitates a natural balance between fairness and efficiency, making it well-suited to complex logistics environments where strict sequencing is impractical but service differentiation is essential.

2.1.2 Hard Priority Models

Hard priority models are designed to strictly enforce customer precedence, ensuring that higher-priority customers are always serviced before lower-priority ones. Unlike soft priority, which allows for operational flexibility, hard priority introduces rigid constraints into the routing process. These constraints are often essential in mission-critical scenarios, where delays in servicing high-priority nodes could result in severe consequences. However, the inclusion of hard priority also increases the complexity of the optimization problem and can lead to infeasibility when resources are limited.

Avila-Torres et al. (2020) proposed a hard-priority vehicle routing model for medical oxygen supply, where hospitals are legally mandated to receive priority service. Their formulation ensures that under no circumstances can vehicles prioritize other customer types over hospitals, regardless of travel cost or inventory needs. This model captures the regulatory and ethical imperatives involved in emergency

medical logistics, making it highly relevant for applications such as disaster response or healthcare supply chains.

In another application, Wu and Zeng (2023) studied a dynamic VRP with stochastic demand, where customer locations are fixed but the demands vary over time. Their model enforces hard priority only when the vehicle capacity is insufficient to serve all customers in a single tour. Under such conditions, a hierarchical service rule is applied to exclude lower-priority customers entirely, thereby guaranteeing service for high-priority clients. This conditional application of hard priority reflects real-world triage behavior, where limited resources are rationed to serve the most critical demands first.

Doan et al. (2021) introduced a multi-purpose VRP framework that allows for customizable urgency levels. Their model enables users to classify requests based on urgency, with high-urgency demands treated using strict hard-priority rules. These requests receive absolute service precedence, even if it leads to longer routes or higher operational costs. Conversely, lower-urgency requests are subject to more flexible service rules, allowing the model to trade off between cost and service quality. This approach offers a hybrid between hard and soft priority strategies, making it adaptable to a wider range of operational contexts.

While hard priority provides strong guarantees for high-importance customers, it can also introduce practical challenges. Strict precedence rules may limit feasible routing combinations, especially when combined with other constraints such as time windows, vehicle capacity, and service time. Moreover, the need to reserve space or time for priority customers can result in underutilization of vehicle capacity or increased idle time, thereby reducing overall routing efficiency.

In this dissertation, although the proposed MO-VRPTWDP model does not adopt hard priority constraints, it draws inspiration from their ability to emphasize service differentiation. Instead of enforcing strict sequencing, the model uses a weighted objective function that encourages reduced waiting times for higher-priority customers. This approach retains the flexibility of soft priority while maintaining a degree of preferential treatment for important clients, thereby achieving a balance between operational feasibility and service equity.

2.1.3 Discussion and Positioning

As summarized in Table 2.1, prior research reflects a diverse set of approaches to modeling customer priority in vehicle routing. These include both soft and hard formulations, integration with time windows, and the use of multi-objective functions. However, many models either simplify priority representation or do not

integrate it directly into the optimization objective.

Table 2.1: Summary of VRPTW with Priority Features in Existing Literature

Study	Priority Type	Time Windows	Pick-up	Multi-Obj.	Waiting Time
Avila-Torres et al. (2020)	**Hard**	–	–	–	–
Baradaran et al. (2019)	**Soft**	**Multiple**	–	✓	–
Beheshti et al. (2015)	**Soft**	**Multiple**	–	✓	–
Das et al. (2020)	**Soft**	**Multiple**	–	✓	–
Doan et al. (2021)	**Soft & Hard**	–	–	–	–
Ghannadpour (2019)	**Soft**	**Single**	–	✓	–
Nucamendi-Guillén et al. (2020)	**Soft**	–	–	✓	✓
Wu and Zeng (2023)	**Hard**	–	✓	–	✓
This study	**Soft**	**Single**	–	✓	✓

The model proposed in this dissertation, Multi-Objective Vehicle Routing Problem with Time Windows and Demand Priority (MO-VRPTWDP), falls under the soft priority category and distinguishes itself by embedding priority scores directly into the multi-objective function. In particular, the second objective seeks to minimize the total customer waiting time, adjusted by customer-specific priority scores. This design allows higher-priority customers to exert greater influence on route selection, without enforcing rigid service order constraints that could lead to infeasibility or operational inefficiency.

Unlike earlier models that use satisfaction indicators or ranked penalties, the MO-VRPTWDP explicitly models priority-adjusted waiting time, offering a more interpretable and flexible representation of service quality. Moreover, by combining this with a classical distance-minimization objective in a multi-objective framework, the model facilitates trade-off analysis and supports decision-making in applications where responsiveness, fairness, and operational efficiency must be balanced.

2.2 Multi-Objective Optimization for Vehicle Routing Problem

In real-world logistics problems, especially vehicle routing, decision-makers often face conflicting objectives, such as minimizing total distance, reducing environmental impact, and improving customer service levels. As a result, significant research has focused on developing *multi-objective optimization* approaches for the Vehicle Routing Problem (VRP), particularly for its complex variants like the VRPTW. Two major algorithmic paradigms have been widely explored in this context: **Multi-Objective Evolutionary Algorithms (MOEAs)** and **Multi-Objective Simulated Annealing (MOSA)**.

2.2.1 Multi-Objective Evolutionary Algorithms (MOEAs)

Multi-Objective Evolutionary Algorithms (MOEAs) are among the most widely adopted metaheuristics for solving multi-objective optimization problems, especially in complex combinatorial domains such as the Vehicle Routing Problem (VRP). MOEAs are well-suited for such problems due to their inherent ability to explore diverse regions of the solution space and maintain a population of non-dominated solutions across generations. This capability is critical for approximating the Pareto front in problems where objectives conflict, such as minimizing operational cost while maximizing customer service level or reducing environmental footprint.

The Nondominated Sorting Genetic Algorithm II (NSGA-II) (Deb et al., 2002) and the Strength Pareto Evolutionary Algorithm 2 (SPEA2) (Zitzler et al., 2001) are two foundational MOEAs that have been widely used in VRP research. Both algorithms maintain an archive of non-dominated solutions and employ mechanisms such as crowding distance (NSGA-II) or fitness assignment (SPEA2) to preserve diversity and convergence across the Pareto front. These algorithms serve as the basis for numerous variants and hybrid approaches aimed at enhancing performance in real-world VRP settings.

Several researchers have customized MOEAs for solving the Vehicle Routing Problem with Time Windows (VRPTW), a particularly challenging variant due to the temporal constraints imposed on service times. For example, Srivastava et al. (2021) modified NSGA-II by incorporating domain-specific crossover and mutation operators tailored to VRPTW, thereby improving convergence and diversity. Similarly, Long et al. (2019) introduced a Hybrid Genetic Local Search (HGLS) that combines evolutionary exploration with local refinement, yielding superior results compared to SPEA2 on benchmark instances.

To better address problem-specific objectives, hybrid frameworks have been explored. Barma et al. (2023) proposed GRASP-NSGA-II for a latency-focused bi-objective VRP. Their framework leverages the Greedy Randomized Adaptive Search Procedure (GRASP) to initialize high-quality populations before applying NSGA-II, demonstrating significant performance gains over baseline methods. In a more advanced formulation, Beheshti et al. (2015) introduced a Cooperative Coevolutionary Multi-Objective Quantum-Genetic Algorithm (CCMQGA), which models interdependencies among problem components and handles VRPs with multiple prioritized time windows. Their results showed that CCMQGA outperforms NSGA-II in both diversity and convergence.

The applicability of MOEAs extends beyond classical VRPTW. In logistics systems involving hybrid delivery modes, such as truck-drone routing, Kuo et al.

(2023a) and Zhang et al. (2022) adapted NSGA-II and its variants to optimize objectives including makespan, energy consumption, and cost. These studies highlight the adaptability of MOEAs to emerging logistics contexts with more complex operational objectives.

In the realm of green and sustainable logistics, Eslamipoor (2024) and Sun and Wang (2023) employed Multi-Objective Particle Swarm Optimization (MOPSO) alongside NSGA-II to solve multi-echelon and environmentally-aware VRPs. Their findings consistently show that MOPSO offers competitive or superior performance in convergence speed and diversity preservation, especially when dealing with multiple conflicting sustainability objectives. Similarly, Kuo et al. (2023b) used MOPSO for multi-objective VRPTW involving supply chain costs and carbon emissions, confirming the viability of swarm-based approaches as alternatives to MOEAs.

In addition to evolutionary and swarm-based strategies, bio-inspired algorithms like Ant Colony Optimization (ACO) have been adapted for multi-objective VRP. Das et al. (2020) proposed a Collaborative Pareto Ant Colony Optimization (P-ACO) algorithm for synchronized truck-drone delivery with time windows. Their approach integrates cooperative pheromone updating mechanisms and Pareto archiving to enhance both solution quality and computational efficiency compared to NSGA-II.

Despite these advances, most MOEA-based methods focus on approximating the entire Pareto frontier without necessarily incorporating user preferences or domain-specific prioritization into the search process. Moreover, while MOEAs offer strong exploratory power, they may suffer from premature convergence or limited exploitation capability unless augmented with local search or adaptive control mechanisms.

This dissertation addresses these limitations by complementing the exploratory strengths of MOEAs with a Simulated Annealing-based framework, MTSA, and a Reinforcement Learning-guided extension, RL-MTSA, that introduces adaptive, preference-aware optimization in the context of VRPTW with Demand Priority. These contributions aim to overcome the lack of fine-grained user control and scalable exploitation strategies often observed in conventional MOEAs.

2.2.2 Multi-Objective Simulated Annealing (MOSA)

While Multi-Objective Evolutionary Algorithms (MOEAs) dominate the landscape of multi-objective optimization in vehicle routing, Multi-Objective Simulated Annealing (MOSA) provides a compelling alternative, particularly for applications that benefit from simpler algorithmic frameworks and stronger local refinement capabilities. Originally proposed by Suppapitnarm et al. (2000), MOSA extends the

classical Simulated Annealing (SA) method by incorporating dominance-based acceptance criteria and solution archiving strategies. Instead of relying on population evolution, MOSA maintains an archive of non-dominated solutions and explores the search space through stochastic neighborhood perturbations guided by temperature schedules.

The core idea behind MOSA is to accept new solutions not only based on energy difference (as in single-objective SA), but also on whether the candidate solution improves the Pareto front or contributes to maintaining solution diversity. This makes MOSA a lightweight yet effective approach for approximating Pareto fronts, especially in problems where fine-grained local optimization is crucial and where memory or computational constraints make population-based methods less desirable.

Several studies have demonstrated the effectiveness of MOSA in solving VRP-related problems. For example, Baños et al. (2013) proposed a parallel Multi-Temperature Pareto Simulated Annealing (pMT-PSA) algorithm for the VRPTW with load imbalance constraints. By leveraging parallelism and adaptive temperature control, their approach produced high-quality Pareto fronts while outperforming parallel SPEA2 in terms of runtime and solution diversity. This result highlights the advantage of SA-based methods in computationally constrained environments where rapid convergence is essential.

In another study, Zidi et al. (2012) employed MOSA for the Multi-Objective Dial-a-Ride Problem, optimizing both route duration and service quality metrics. The algorithm showed competitive performance across multiple benchmark instances, demonstrating MOSA's suitability for problems with passenger satisfaction and fairness constraints, which are analogous to demand priority in VRPTWDP.

A notable hybridization of MOSA is presented in the work of Wang et al. (2020), who introduced MOSA-ACO, a method that combines Ant Colony Optimization (ACO)'s exploratory ability with MOSA's intensification capability. Applied to the Periodic Vehicle Routing Problem with Time Windows and Service Choice (PVRPTW-SC), the hybrid algorithm outperformed standalone ACO and MOSA implementations, illustrating the complementary strengths of metaheuristics in balancing global and local search.

Beyond SA-based methods, recent work has explored alternative swarm-inspired metaheuristics for multi-objective VRP. Golmohammadi et al. (2024) proposed the Multi-Objective Dragonfly Algorithm (MODA) for Location Routing Problems (LRPs), which integrates food source attraction and repulsion mechanisms from natural dragonfly behavior to enhance exploration. Their experiments show that

MODA can outperform NSGA-II in both convergence speed and Pareto front quality. Similarly, Li et al. (2022b) introduced a Discrete Multi-Objective Grey Wolf Optimizer (DMOGWO) for home healthcare routing problems. This algorithm mimics hierarchical leadership and hunting strategies, delivering superior performance in terms of solution spread and convergence when benchmarked against NSGA-II, NSGA-III, and PSO.

Despite their promise, SA-based methods like MOSA have received comparatively less attention in recent years due to the surge of interest in evolutionary and swarm intelligence algorithms. However, their simplicity, ease of adaptation, and strong exploitation behavior make them highly relevant for large-scale VRP problems where fast local refinement is crucial.

This dissertation contributes to this underexplored area by proposing a new Multi-Thread Simulated Annealing (MTSA), which enhances MOSA through parallel exploration and inter-thread cooperation. Unlike conventional MOSA, which operates in a single thread, MTSA enables simultaneous, cooperative searches in different regions of the solution space, improving the approximation of the Pareto frontier. Additionally, MTSA forms the basis of a learning-augmented extension, RL-MTSA, which further incorporates user preference into the search process. These contributions aim to rejuvenate the relevance of SA-based methods in modern multi-objective VRP optimization, particularly in contexts where hybrid strategies and adaptive guidance are essential.

2.2.3 Discussion and Positioning

Multi-objective optimization in vehicle routing has been predominantly shaped by the use of Multi-Objective Evolutionary Algorithms (MOEAs), owing to their strong exploratory capabilities and the ease with which they approximate complex, high-dimensional Pareto fronts. Their population-based structure allows them to maintain diversity across solutions and explore multiple trade-off regions simultaneously. However, MOEAs typically come with significant computational costs, both in terms of memory and processing time, and often require careful parameter tuning. Moreover, their reliance on stochastic operators can lead to slower convergence and less predictable behavior in local refinement, especially for large and tightly constrained instances.

In contrast, Multi-Objective Simulated Annealing (MOSA) represents a more lightweight and computationally efficient alternative. Its trajectory-based search makes it particularly suitable for problems that benefit from strong local exploitation. Due to its simplicity, MOSA is also easier to implement and adapt to domain-

specific constraints. However, the main drawback of MOSA lies in its limited ability to maintain diversity and perform global exploration effectively. As problem size and objective complexity grow, the lack of population dynamics and parallel search mechanisms hinders its performance in discovering a well-distributed Pareto front.

Recognizing these complementary strengths and limitations, this dissertation revisits the potential of SA-based methods by proposing a Multi-Thread Simulated Annealing (MTSA) algorithm tailored for multi-objective VRP. MTSA enhances the classical MOSA framework by introducing multiple parallel search threads, each operating with an independent weighted objective. These threads periodically exchange information to diversify search directions and prevent premature convergence. This parallel and cooperative design enables MTSA to retain the refinement precision of SA while expanding its exploratory capacity across the Pareto front.

The introduction of MTSA addresses a notable gap in the literature by bridging the performance divide between population-based and trajectory-based metaheuristics. It offers a scalable and tunable alternative that is particularly well-suited for complex VRP variants where both convergence efficiency and solution diversity are essential. Furthermore, MTSA serves as the foundation for the reinforcement learning-enhanced framework (RL-MTSA), which brings preference-awareness into the optimization process, an aspect rarely addressed in traditional MOSA or MOEA implementations.

This positioning not only highlights the methodological contributions of this dissertation but also emphasizes its relevance in advancing the field of multi-objective vehicle routing under realistic, large-scale, and user-driven decision environments.

2.3 Knee-Oriented and Preference-Based Multi-Objective Optimization

In multi-objective optimization, decision-makers are often interested not in the entire Pareto front, but in specific regions that reflect their preferences or operational priorities. This has led to the development of preference-based optimization strategies, which aim to guide the search process toward more relevant or desirable solutions. Among these, two main paradigms have emerged: *knee-oriented optimization*, which targets inherently balanced trade-off points, and *explicit preference-based optimization*, which seeks to align search behavior with user-defined objectives or regions of interest. The following subsections review the literature on these two approaches and position the proposed RL-MTSA framework within this context.

2.3.1 Knee-Oriented Multi-Objective Optimization

Knee-oriented optimization focuses on identifying regions along the Pareto front where marginal improvements in one objective incur disproportionately large sacrifices in another. These so-called *knee points* often represent well-balanced trade-off solutions and are therefore of particular interest in decision-making contexts. They are especially valuable when stakeholders seek compromise solutions that offer high efficiency across competing objectives.

A number of recent studies have explored knee-point-driven optimization across various domains. For instance, Yu et al. (2022) integrated a knee-point-based selection mechanism into a multi-objective economic and environmental dispatch problem, improving both convergence and decision relevance. Similarly, Zhao et al. (2022) proposed a knee-guided dominance mechanism to refine Pareto front approximations in many-objective optimization, enhancing selection efficiency and interpretability. Zhang et al. (2021) introduced MMO-EvoKnee, a multimodal evolutionary algorithm capable of detecting global knee points while filtering out redundant solutions, thus preserving meaningful trade-offs. In a related effort, Xu and Yu (2023) developed a grid-based measurement approach that combines knee-point identification with plane-based performance metrics to improve solution quality in environmental economic dispatch problems.

Further work by Chen et al. (2024) applied knee-point-based strategies to bi-level optimization, strategically allocating computational effort to the most relevant trade-off regions while reducing unnecessary search overhead. In transportation, Guo et al. (2023) introduced a hybrid evolutionary strategy for air traffic flow management that balances exploitation of knee regions with broader search mechanisms, enhancing scalability in large-scale problems.

Despite these advances, knee-point methods remain underexplored in the context of vehicle routing problems (VRPs). Most multi-objective VRP methods focus on generating uniformly distributed approximations of the Pareto front, often without explicitly emphasizing trade-off-sensitive regions. Moreover, these techniques tend to rely on fixed heuristics or dominance-based criteria that do not dynamically adapt to real-world complexities such as time windows, fluctuating demand, or operator preferences.

2.3.2 Preference-Based Multi-Objective Optimization

Traditional multi-objective optimization methods aim to generate a well-distributed approximation of the Pareto front, from which decision-makers select solutions after

the optimization is complete. However, in many real-world applications, only a specific subset of the Pareto front is of practical interest, often driven by stakeholder preferences or operational constraints. This has given rise to *preference-based multi-objective optimization*, where search is focused on user-defined or learned regions of interest.

Initial strategies for incorporating preferences include scalarization methods (e.g., weighted-sum, ε-constraint), reference-point approaches (e.g., R-NSGA-II), and decomposition-based methods (e.g., MOEA/D). However, these typically require fixed preference articulation and are not well-suited to dynamic, uncertain, or evolving environments.

More recent approaches adopt *interactive* and *learning-based* paradigms. For example, Li et al. (2023) proposed a framework that learns user preferences through ranking feedback and incorporates them into an evolutionary multi-objective algorithm. The learned ranker biases the population toward regions that reflect decision-maker interest. Similarly, Huang et al. (2024) applied a dueling bandit mechanism to acquire preferences via pairwise comparisons, removing the need for explicit utility functions.

In the reinforcement learning domain, Xu et al. (2022) introduced a preference-based multi-objective reinforcement learning framework for optimizing multi-microgrid operations in smart grids. Their method dynamically adapts to different user preference vectors, enabling flexible trade-off control and significantly improving operational outcomes. The ability to embed user-defined preference vectors as reward signals offers an elegant mechanism for preference modeling in high-dimensional environments.

Despite increasing popularity, preference-based methods remain underexplored in the vehicle routing problem. Most VRP research continues to focus on uniformly sampling the Pareto front, without emphasizing practical trade-off zones. In this context, the proposed RL-MTSA framework in this dissertation stands out by enabling user-defined region prioritization. Rather than searching the full Pareto front, RL-MTSA integrates preference signals to guide search toward user-specified objective ranges. This preference-aware capability enables interactive, adaptive decision support in multi-objective vehicle routing problems.

2.3.3 Discussion and Positioning

While knee-point optimization focuses on identifying regions of the Pareto front that represent strong trade-offs between objectives, it remains fundamentally driven by the geometry of the front itself. This makes it useful for post-optimization decision-

making, but less suitable for scenarios where decision-makers have specific, evolving preferences that cannot be captured solely by inherent front properties. In the context of vehicle routing problems (VRP), where practical considerations such as customer types, regional priorities, or service agreements significantly influence solution relevance, knee-based methods often fall short.

Preference-based optimization approaches attempt to address this gap by incorporating user preferences. While effective in some domains, many of these approaches are limited by static preference articulation or require frequent user intervention. Moreover, a few of these frameworks are integrated into large-scale metaheuristics for combinatorial problems like VRP, where the solution structure is complex and domain constraints are tight.

The Reinforcement Learning Multi-Thread Simulated Annealing (RL-MTSA) framework proposed in this dissertation distinguishes itself on both fronts. First, it goes beyond knee-oriented search by prioritizing user-defined regions of interest rather than relying on geometric inflection points. Second, it advances beyond traditional preference-based methods by embedding a reinforcement learning agent directly into the optimization loop. This agent dynamically adjusts the search trajectory in response to preference-aligned feedback, enabling real-time adaptation to evolving decision priorities.

Unlike scalarization-based methods, RL-MTSA does not require fixed weight vectors or utility functions to be defined a priori. And unlike interactive MOEAs, it does not rely on high-frequency human interaction during the search. Instead, it treats preference representation as a learning task, training an agent to recognize and exploit areas of the Pareto front that align with the user's preference.

This flexibility is particularly valuable for complex real-world VRPs, where the notion of an "ideal" trade-off may change with operational context, such as fleet availability, service-level agreements, or delivery urgency. By guiding the search adaptively, RL-MTSA enhances computational efficiency and solution relevance, supporting more targeted, responsive decision-making.

In summary, RL-MTSA contributes a novel, scalable methodology for solving preference-driven multi-objective routing problems. It bridges the limitations of both knee-point and traditional preference-based methods through its learning-based, adaptive control of search focus, offering a practical path forward for real-world logistics optimization.

2.4 Reinforcement Learning Based Algorithm for Vehicle Routing Problems

Reinforcement Learning (RL) has gained considerable attention in addressing combinatorial optimization problems, including the Vehicle Routing Problem (VRP), due to its ability to learn adaptive policies through iterative interactions with the environment. Unlike classical heuristics that rely on static decision rules, RL agents improve their decision-making strategies by receiving feedback in the form of rewards, making them particularly suitable for dynamic and uncertain logistics environments.

2.4.1 Reinforcement Learning Based Algorithm

In the context of VRP, RL research can be broadly categorized into three main streams: end-to-end approaches, step-by-step methods, and hybrid RL-metaheuristic frameworks. End-to-end RL models learn a direct mapping from raw problem instances to routing decisions without requiring manual rule design, leveraging deep reinforcement learning (DRL) with neural network architectures to capture complex spatial-temporal dependencies. Recent studies, including Lin et al. (2022), Pan and Liu (2023), Zou et al. (2024b), Li et al. (2022a), and Bono et al. (2021), have employed attention-based encoder-decoder and Transformer architectures to address variants such as the Electric VRPTW, Dynamic VRP, Multi-Depot VRP, and Heterogeneous Capacitated VRP. These studies focus on objectives such as cost reduction, carbon minimization, and feasibility under real-time constraints. While demonstrating strong potential, such models often require extensive computational resources and large-scale data for effective policy training.

Step-by-step RL methods incrementally construct or refine routes, enabling adaptive decision-making during execution while incorporating local search refinements. For instance, Wu et al. (2022) proposed an RL-guided improvement method using 2-opt and node-swap operators, while Phiboonbanakit et al. (2021) integrated RL with regression tree models to enhance dynamic vehicle assignment. Zhang et al. (2023) developed a hierarchical RL framework for customer selection and dispatching under demand uncertainty, showing improved adaptability while maintaining computational tractability.

Hybrid RL-metaheuristic frameworks combine the flexibility of RL with the structured exploration capabilities of metaheuristics to solve complex VRP instances. Zhao et al. (2021) and Kalatzantonakis et al. (2023) employed RL for

adaptive operator selection within Variable Neighborhood Search, improving convergence and solution quality. Pugliese et al. (2023) incorporated RL into a learning-augmented Variable Neighborhood Search for crowd-shipping VRPs, while Fitzpatrick et al. (2024) combined RL with MILP decomposition for large-scale VRP to enhance scalability. Gao et al. (2025) leveraged RL with graph convolutional networks within a column generation pipeline for VRPs with release dates and loading constraints, and Zou et al. (2024a) introduced an RL-guided hybrid evolutionary algorithm with edge assembly crossover for latency location routing problems (LLRP), showing notable improvements in targeted objectives and computational efficiency.

Additionally, Qin et al. (2021) proposed a hybrid RL-based hyper-heuristic framework for VRP, where RL dynamically selects among heuristic operators during the search process, balancing exploration and exploitation effectively. Hu et al. (2020) utilized a GNN-based RL model to address real-world logistics VRPs, capturing the relational structures of routing networks to enhance decision-making quality under practical constraints. Liu et al. (2023) extended the application of RL to multi-objective contexts by integrating RL with multi-objective evolutionary algorithms (MOEAs) for the multi-objective orienteering problem (MO-OP), demonstrating that RL can guide the search toward high-quality, diverse solutions across multiple conflicting objectives.

Despite these advancements, a notable limitation persists across these studies: the predominant focus on single-objective VRPs with limited mechanisms for dynamically aligning search processes with evolving user preferences or for targeting specific regions of interest within the Pareto frontier. Existing RL frameworks excel at generating diverse solutions across the objective space but often lack the capability to focus computational effort on preference-aligned trade-off zones critical for decision support in real-world logistics operations.

2.4.2 Discussion and Positioning

Table 2.2 summarizes recent RL-based VRP studies, categorizing them by objective type, RL approach, hybridization structure, and problem focus. It illustrates that while a variety of methods effectively address single-objective VRPs using end-to-end, step-by-step, or hybrid RL-metaheuristic approaches, preference-aware multi-objective VRP optimization remains underexplored. Existing RL frameworks predominantly focus on generating high-quality solutions across the Pareto frontier without providing mechanisms to dynamically align the search with evolving decision-maker preferences or to target specific, user-defined trade-off regions within the solution space.

Table 2.2: Summary of RL-Based VRP Literature

Study	**Objective**	**RL Type**	**Hybridization**	**Preference**
Lin et al. (2022)	Single	End-to-End	–	–
Pan and Liu (2023)	Single	End-to-End	–	–
Zou et al. (2024b)	Single	End-to-End	–	–
Li et al. (2022a)	Single	End-to-End	–	–
Bono et al. (2021)	Single	End-to-End	–	–
Wu et al. (2022)	Single	Step-by-Step	RL + Local Search	–
Phiboonbanakit et al. (2021)	Single	Step-by-Step	RL + Regression Trees	–
Zhang et al. (2023)	Single	Step-by-Step	RL + Hierarchical Decision	–
Zhao et al. (2021)	Single	Hybrid	RL + Local Search	–
Kalatzantonakis et al. (2023)	Single	Hybrid	RL + VNS	–
Pugliese et al. (2023)	Single	Hybrid	RL + ML-VNS	–
Qin et al. (2021)	Single	Hybrid	RL + Hyper-heuristic	–
Fitzpatrick et al. (2024)	Single	Hybrid	RL + MILP	–
Gao et al. (2025)	Single	Hybrid	RL + Column Generation	–
Zou et al. (2024a)	Single	Hybrid	RL + EA	–
Hu et al. (2020)	Single	Hybrid	RL + GNN	–
Liu et al. (2023)	Multiple	Hybrid	RL + MOEA	–
RL-MTSA	Multiple	Hybrid	RL + MTSA	✓

To address this research gap, this dissertation proposes the Reinforcement Learning Multi-Thread Simulated Annealing (RL-MTSA) framework, which enables dynamic, user-specified, preference-focused exploration within the Pareto frontier. By integrating reinforcement learning into a multi-thread simulated annealing environment, RL-MTSA adaptively focuses the search on user-identified regions of interest while preserving diversity and computational efficiency. This integration bridges the gap between decision-maker needs and algorithmic solution generation, offering a scalable and practical methodology for real-world multi-objective routing and logistics planning where responsiveness and targeted decision support are essential.

2.5 Summary and Research Positioning

This chapter has provided a comprehensive review of the background and related work that underpins this dissertation, covering four key areas: vehicle routing with demand priority, multi-objective optimization and metaheuristics, preference-aware optimization strategies, and reinforcement learning for combinatorial optimization. Each section highlighted both the progress and the limitations within the existing literature, framing the research gaps that this dissertation aims to address.

In the domain of **priority modeling in vehicle routing**, while various soft and hard priority frameworks have been explored, most existing studies either employ rigid hard-priority constraints that may reduce operational flexibility or adopt soft-priority mechanisms without embedding them explicitly within multi-objective optimization frameworks. Furthermore, priority modeling in VRPTW has typically been approached using satisfaction scores or ranked penalties rather than direct, interpretable weighted waiting time objectives that allow clear trade-off analysis between service equity and operational efficiency.

In the area of **multi-objective optimization for VRP**, Multi-Objective Evolutionary Algorithms (MOEAs) have dominated due to their strong exploratory capabilities, but they often come with high computational costs and may require extensive parameter tuning to achieve balanced convergence and diversity. Conversely, Multi-Objective Simulated Annealing (MOSA) offers a lightweight alternative with strong local exploitation capabilities but suffers from limited global exploration and diversity maintenance, hindering its effectiveness on large-scale, complex VRP instances.

Regarding **preference-based optimization**, while knee-oriented and preference-guided strategies have been developed in various domains, they remain underutilized in VRP contexts. Most VRP-focused multi-objective frameworks aim to approx-

imate the full Pareto front without mechanisms to align the search with evolving user preferences or to focus on regions of operational interest within the solution space.

Finally, in the field of **reinforcement learning for combinatorial optimization**, RL has demonstrated potential for routing and scheduling problems, primarily in single-objective contexts, with a strong focus on end-to-end learning or hybridization with evolutionary or swarm-based metaheuristics. However, the integration of RL into trajectory-based metaheuristics such as Simulated Annealing, particularly for enabling preference-aware, multi-objective optimization in VRP, remains largely unexplored.

Building on these observations, this dissertation positions itself to address the following identified research gaps:

1. The absence of soft-priority, multi-objective VRPTW formulations that directly incorporate customer importance using interpretable, weighted waiting time objectives.

2. The underutilization of Simulated Annealing for multi-objective VRP due to challenges in maintaining diversity and effective global exploration.

3. The scarcity of frameworks that support dynamic, user-guided, preference-aware optimization within multi-objective vehicle routing contexts.

4. The lack of reinforcement learning integration within trajectory-based metaheuristics for scalable, adaptive, and preference-aligned multi-objective optimization in VRP.

To address these gaps, this dissertation proposes a threefold contribution:

1. The development of a **Multi-Objective Vehicle Routing Problem with Time Windows and Demand Priority (MO-VRPTWDP)** formulation, embedding soft-priority mechanisms through weighted waiting time objectives alongside classical cost objectives.

2. The design of a **Multi-Thread Simulated Annealing (MTSA)** algorithm, extending classical SA with parallel exploration and cooperative search to enhance Pareto front approximation while retaining strong local refinement capabilities.

3. The creation of the **RL-MTSA framework**, integrating reinforcement learning within MTSA to enable dynamic, preference-aware guidance of the search process toward user-specified regions of the Pareto frontier.

Collectively, these contributions establish a complete, scalable, and adaptable methodology for preference-aware, multi-objective vehicle routing optimization, bridging the gap between algorithmic search and decision-maker priorities in complex, real-world logistics environments.

The next chapter will formally introduce the problem formulation of MO-VRPTWDP, detailing its mathematical structure, operational assumptions, and validation experiments to demonstrate its feasibility and relevance for practical vehicle routing scenarios.

Chapter 3

Multi-Objective VRPTW with Demand Priority

This chapter presents the formal definition and mathematical formulation of the Multi-Objective Vehicle Routing Problem with Time Windows and Demand Priority (MO-VRPTWDP). The model incorporates customer-specific service priorities as soft constraints, aiming to balance route efficiency with fairness in customer service. We begin with a description of the problem setting and the motivation behind using soft demand priority, followed by the mathematical model formulation, a small computational experiment, and a discussion of its properties.

3.1 Problem Description

We propose a soft priority model for VRPTW that allows route planners to account for customer waiting times, particularly relevant when customer satisfaction depends on how long they wait to be served. Traditional VRPTW models treat all customers equally with respect to timing. However, this assumption is often unrealistic, as some customers (e.g., hospitals, premium services) require faster service than others.

The proposed model introduces an adjusted waiting time objective that reflects this variability. Rather than enforcing a strict service sequence, we treat priority levels as weights in the objective function. This allows higher-priority customers to influence the route planning without violating feasibility constraints, especially those imposed by time windows.

3.1.1 Demand Priority

Demand priority is modeled as a soft constraint. That is, vehicles are allowed to serve lower-priority customers before higher-priority ones if necessary to satisfy other constraints (e.g., time windows or capacity). The key idea is to minimize the *adjusted waiting time* across all customers, giving more influence to high-priority requests.

This design ensures flexibility in real-world applications while maintaining fairness. Additional constraints may be imposed to limit the maximum allowable waiting time for high-priority customers, allowing planners to adjust sensitivity to different service expectations.

3.1.2 Model Assumptions

To formulate the VRPTWDP with demand priority, several assumptions are adopted to balance realism with tractability:

1. **Fleet and Depot.** A homogeneous fleet of vehicles is stationed at a single depot. Each vehicle has an identical capacity and must start and end its route at the depot.

2. **Customer Characteristics.** Each customer is associated with a fixed demand, a deterministic service time, and a specified time window during which service must begin. Service before the earliest time is not allowed, while service after the latest time is infeasible.

3. **Priority Representation.** Each customer is assigned a nonnegative priority score, reflecting its relative importance. Priorities are treated as *soft constraints*, meaning that lower-priority customers may be served earlier if necessary to satisfy feasibility (e.g., time windows or capacity). The effect of priority is incorporated through a weighted adjustment of waiting times in the objective function.

4. **Waiting Time.** Waiting time is defined as the difference between the actual start of service and the earliest allowable arrival time. Travel distance is assumed to be directly proportional to travel time, with one distance unit corresponding to one time unit. Adjusted waiting time is computed by multiplying the waiting time by the customer's priority score, thereby emphasizing delays for high-priority customers.

5. **Routing Feasibility.** Each customer must be visited exactly once by exactly one vehicle. Vehicle capacity constraints must not be violated, and travel times are deterministic and equal to travel distances under the constant-speed assumption.

6. **Optimization Objectives.** The model considers two objectives simultaneously: (i) minimizing total travel distance and (ii) minimizing the total adjusted waiting time across all customers. The inclusion of demand priority ensures that routes remain flexible while giving greater influence to high-priority customers.

These assumptions ensure that the VRPTWDP formulation captures both logistical efficiency and differentiated customer service, while remaining solvable within reasonable computational resources.

3.2 MILP Formulation of MO-VRPTWDP

Before presenting the full MILP formulation of MO-VRPTWDP, it is important to justify the role and necessity of using a Mixed Integer Linear Programming model in this context. While simplified formulations may help convey the problem's logic, they often fall short in representing the full operational complexity required for rigorous optimization and reproducibility. MILP offers a mathematically precise and flexible framework to model discrete decision variables, logical constraints, and multi-objective trade-offs, features that are central to vehicle routing problems (VRPs).

Moreover, MILP formulations serve as a foundation for exact solution approaches, such as branch-and-bound and cutting-plane algorithms, which can provide optimal solutions for small to medium instances. These exact results are essential for validating the correctness of the model and serve as benchmarks for the evaluation of meta-heuristic methods in subsequent chapters. The use of MILP in VRP research is well-established in the literature, particularly in the foundational work by Toth and Vigo (2002), which formalized many exact and heuristic approaches and established MILP as a core modeling tool in vehicle routing problem and combinatorial optimization.

We now present the Mixed Integer Linear Programming (MILP) formulation. The model builds upon the classical VRPTW formulation (El-Sherbeny, 2010) by incorporating demand priority into one of the objectives.

Let $G = (V, E)$ be a complete undirected graph, where $V = \{0, 1, 2, \ldots, N\}$ denotes the set of nodes, with node 0 representing the depot and nodes 1 through N representing the customers. The set of edges is defined as $E = \{(i, j) \mid i, j \in V, i \neq j\}$, where each edge represents a possible connection between two distinct nodes.

Decision Variables

- $X_{ijk} = \begin{cases} 1, & \text{if vehicle } k \text{ travels from node } i \text{ to } j \\ 0, & \text{otherwise} \end{cases}$
- s_{ki} : Time of the start of service for vehicle k at node i
- seq_i : Sequence of customer node i

Parameters

- a_j: Earliest allowable time at which service can begin for customer j.
- b_j: Latest allowable time by which service must begin for customer j.
- C_{ij}: Travel cost or distance between node i and node j.
- T_{ij}: Travel time required to move from node i to node j.
- d_j: Demand of customer j, representing the load to be delivered.
- $serv_j$: Service time required to complete delivery at customer j.
- K: Maximum number of available vehicles in the fleet.
- Q: Capacity limit of each vehicle.
- N: Total number of customer nodes. The depot is indexed as node 0, and customers are indexed as $1, \ldots, N$.
- p_j: Priority score assigned to customer j, representing the relative importance or urgency of service.

Objective Functions

$$\text{Minimize} \quad TD = \sum_{i=0}^{N} \sum_{j=0}^{N} \sum_{k=1}^{K} X_{ijk} C_{ij} \tag{3.1}$$

$$\text{Minimize} \quad WT = \sum_{j=1}^{N} \left[p_j \left(\left(\sum_{k=1}^{K} s_{kj} \right) - a_j \right) \right] \tag{3.2}$$

Constraints

$$X_{iik} = 0, \quad \forall i \in \{0, \ldots, N\}, \forall k \in \{1, \ldots, K\} \tag{3.3}$$

$$\sum_{i=0}^{N} \sum_{k=1}^{K} X_{ijk} = 1, \quad \forall j \in \{1, \ldots, N\} \tag{3.4}$$

$$\sum_{i=0}^{N} X_{izk} = \sum_{j=0}^{N} X_{zjk}, \quad \forall z \in \{0, \ldots, N\}, \forall k \in \{1, \ldots, K\} \tag{3.5}$$

$$\sum_{i=0}^{N} \sum_{j=1}^{N} X_{ijk} d_j \leq Q, \quad \forall k \in \{1, \ldots, K\} \tag{3.6}$$

$$\sum_{j=1}^{N} \sum_{k=1}^{K} X_{0jk} \leq K \tag{3.7}$$

$$\sum_{j=1}^{N} X_{0jk} \leq 1, \quad \forall k \in \{1, \ldots, K\} \tag{3.8}$$

$$\sum_{j=1}^{N} X_{0jk} - \sum_{j=1}^{N} X_{j0k} = 0, \quad \forall k \in \{1, \ldots, K\} \tag{3.9}$$

$$s_{kj} \leq b_j, \quad \forall j \in \{0, \ldots, N\}, \forall k \in \{1, \ldots, K\} \tag{3.10}$$

$$\sum_{k=1}^{K} s_{kj} \geq a_j, \quad \forall j \in \{0, \ldots, N\} \tag{3.11}$$

$$s_{ki} + T_{ij} + serv_i - L(1 - X_{ijk}) \leq s_{kj}, \quad \forall i \in \{1, \ldots, N\}, \forall j \in \{0, \ldots, N\}, \forall k \in \{1, \ldots, K\} \tag{3.12}$$

$$T_{0j} - L(1 - X_{0jk}) \leq s_{kj}, \quad \forall j \in \{1, \ldots, N\}, \forall k \in \{1, \ldots, K\} \tag{3.13}$$

$$s_{kj} \leq \sum_{i=0}^{N} L X_{ijk}, \quad \forall j \in \{1, \ldots, N\}, \forall k \in \{1, \ldots, K\} \tag{3.14}$$

$$seq_j - seq_i \geq 1 - N(1 - \sum_{k=1}^{K} X_{ijk}), \quad \forall i, j \in \{1, \ldots, N\}, i \neq j \tag{3.15}$$

$$s_{ki} \geq 0, \quad \forall i \in \{0, \ldots, N\}, \forall k \in \{1, \ldots, K\} \tag{3.16}$$

$$seq_i \geq 0, \quad \forall i \in \{0, \ldots, N\} \tag{3.17}$$

$$X_{ijk} \in \{0, 1\}, \quad \forall i, j \in \{0, \ldots, N\}, \forall k \in \{1, \ldots, K\} \tag{3.18}$$

Model Description

The objective functions and constraints of the MO-VRPTWDP MILP model can be interpreted as follows:

- **Eq. (1)** defines the first objective, minimizing the total traveling cost or distance.
- **Eq. (2)** defines the second objective, minimizing the total customer waiting time adjusted by priority scores.
- **Eq. (3)** prohibits self-loops, ensuring that a vehicle cannot travel from a node to itself.
- **Eq. (4)** ensures that every customer is visited exactly once, by summing over all possible predecessors and vehicles.
- **Eq. (5)** enforces *flow conservation* for each node and vehicle. It ensures that if a vehicle enters a node, it must also leave that node.
- **Eq. (6)** imposes the vehicle capacity constraint: the total demand on any route must not exceed the vehicle's capacity Q.
- **Eqs. (7)–(9)** collectively control vehicle usage and depot-related constraints:
  - **Eq. (7)** limits the total number of vehicles that can leave the depot.
  - **Eq. (8)** restricts each vehicle to leave the depot at most once.
  - **Eq. (9)** ensures that if a vehicle departs the depot, it must also return — enforcing round-trip tours.
- **Eqs. (10)–(11)** enforce service time feasibility at each customer node:
  - **Eq. (10)** ensures the start of service at any node does not exceed its latest allowable time b_j.
  - **Eq. (11)** ensures that the total start time is not earlier than the earliest service time a_j.
- **Eqs. (12)–(14)** control time progression between visited nodes and synchronize arrival time variables:
  - **Eq. (12)** ensures temporal feasibility: if vehicle k travels from node i to node j, it must arrive at j after completing travel and service at i. Note that L represents a large number (Latest arrival time at depot).
  - **Eq. (13)** applies the same logic to arcs originating from the depot.
  - **Eq. (14)** ensures that the service time variable s_{kj} is correctly bounded only when a vehicle visits node j; it effectively deactivates timing constraints when a node is not visited.

- **Eq. (15)** eliminates subtours by enforcing an ordering constraint via sequence variables seq_i. This guarantees a connected route for each vehicle.
- **Eqs. (16)–(17)** define variable bounds:
  - **Eq. (16)** ensures non-negativity of the service start time.
  - **Eq. (17)** ensures non-negativity of the sequence variables.
- **Eq. (18)** defines the decision variables X_{ijk} as binary, indicating whether arc (i, j) is traveled by vehicle k.

Priority-Based Waiting Time Threshold Constraint

To further control fairness and responsiveness, the model can optionally incorporate upper bounds on customer-specific waiting times using the following constraint:

$$\sum_{k=1}^{K} s_{kj} - a_j \leq TH_j, \quad \forall j \in \{1, \ldots, N\} \tag{3.19}$$

This constraint restricts the waiting time of each customer j to be within a specified threshold TH_j, which may be defined in relation to the customer's priority level. For instance, higher-priority customers may have tighter thresholds to guarantee better service responsiveness.

3.3 Computational Validation of the MILP Model

To verify and validate the proposed mathematical formulation of the MO-VRPTWDP, a set of computational experiments is conducted using a reduced-size benchmark instance. Specifically, a subset of customers is extracted from the well-known Solomon R201 instance. Only 10 customer nodes are selected: {0, 1, 10, 20, 30, 32, 52, 66, 69, 70, 90}, where node 0 represents the depot. This reduced instance allows exact solutions to be obtained using a MILP solver, enabling a focused examination of the model's behavior.

The vehicle capacity is set to 1000, and two vehicles are available. Initially, all customers are assigned a uniform priority score of 1. The instance is solved using the CBC MILP solver. The ε-constraint method is used to explore trade-offs between TD and WT.

3.3.1 Trade-Off Analysis Using the ε-Constraint Method

To investigate the trade-off between total distance and adjusted waiting time, the ε-constraint method is applied by minimizing WT while setting upper bounds on TD. The problem is iteratively solved for a series of increasing distance limits: {135, 140, 145, 150, 155, 160, 165, 170, 175}. Table 3.1 presents the results, showing the corresponding WT and route solutions for each setting.

Table 3.1: Trade-off solutions using the ε-constraint method (uniform priority)

Objective Setting	**Distance (TD)**	**Adjusted Waiting Time (WT)**	**Solution (Customer Visit Order)**
Min TD	129.7	183.4	[52, 69, 30, 90, 10, 20, 66, 32, 70, 1]
Min WT & TD $\leq$ 135	129.7	183.4	[52, 69, 30, 90, 10, 20, 66, 32, 70, 1]
Min WT & TD $\leq$ 140	138.7	159.4	[69, 52, 30, 90, 10, 20, 66, 32, 70, 1]
Min WT & TD $\leq$ 145	143.3	84.3	[52, 69, 30, 90, 10, 20, 66, 32, 1, 70]
Min WT & TD $\leq$ 150	143.3	84.3	[52, 69, 30, 90, 10, 20, 66, 32, 1, 70]
Min WT & TD $\leq$ 155	152.3	60.3	[69, 52, 30, 90, 10, 20, 66, 32, 1, 70]
Min WT & TD $\leq$ 160	156.0	51.5	[52], [69, 30, 90, 10, 20, 66, 32, 1, 70]
Min WT & TD $\leq$ 165	163.9	7.9	[69, 30, 20, 66, 32, 70], [52, 90, 10, 1]
Min WT & TD $\leq$ 170	168.8	7.9	[52, 30, 20, 66, 32, 1], [69, 90, 10, 70]
Min WT & TD $\leq$ 175	174.0	7.9	[52, 30, 90, 20, 66, 32, 1], [69, 10, 70]
Min WT	182.4	7.9	[52, 30, 90, 20, 66, 70], [69, 10, 32, 1]

The results show a clear trade-off between the two objectives: minimizing TD often leads to significantly higher WT, and vice versa. In particular, the solution that minimizes WT yields a much longer route, while the minimum-distance solution results in higher total waiting time. This validates the ability of the model to capture multi-objective trade-offs effectively.

3.3.2 Impact of Priority Levels on Route Selection

To examine how customer priority affects route selection, a second experiment is conducted using the same customer subset but under a single-vehicle setting. Two priority configurations are tested:

- **Uniform Priority (Level 1):** All customers have equal priority.
- **Three-Level Priority:** Customers 20 and 66 (identified as having the longest waiting times in the first setting) are assigned priority level 2. Customers 30, 32, 52, and 69 are assigned level 1, and the remaining customers are assigned level 0.

Table 3.2 summarizes the result of minimizing WT under these two settings. The results confirm that the model is sensitive to priority levels. Adjusting customer priority level setting leads to a significant change in the actual waiting time of the customers, ranging from 60.3 to 273.7 for the 1-level to 3-level priority setting,

respectively. This adjustment occurs because the model selects the routes according to the new priority levels. However, despite efforts to minimize the adjusted waiting time, customer 66 with the highest priority still encounters prolonged waiting times compared to the other customers. This is primarily due to customers 20 and 66 having similar time windows and priority levels, indicating that one cannot be served without incurring additional waiting times for the other.

Table 3.2: Minimizing adjusted waiting times with different priority-setting

Objective	**Actual Waiting Time**	**Adjusted Waiting Time (WT)**	**Solution**
Minimize WT *(1 Priority level)*	60.3	60.3	[69, 52, 30, 90, 10, 20, 66, 32, 1, 70]
Minimize WT *(3 Priority levels)*	273.7	108.8	[69, 52, 30, 90, 20, 10, 66, 32, 70, 1]

Priority Level and Waiting Time for 1 Priority Levels

Customer ID	**Priority Level**	**Actual Waiting Time**	**Adjusted Waiting Time**
1	1	0	0
10	1	0	0
20	1	21.8	21.8
30	1	0	0
32	1	0	0
52	1	8.8	8.8
66	1	29.7	29.7
69	1	0	0
70	1	0	0
90	1	0	0

Priority Level and Waiting Time for 3 Priority Levels

Customer ID	**Priority Level**	**Actual Waiting Time**	**Adjusted Waiting Time**
1	0	141	0
10	0	29.8	0
20	2	0	0
30	1	0	0
32	1	4.4	4.4
52	1	8.8	8.8
66	2	47.8	95.6
69	1	0	0
70	0	41.9	0
90	0	0	0

3.3.3 Priority Sensitivity and Constraint Enforcement

To further examine the behavioral flexibility of the proposed MO-VRPTWDP model, two additional experiments were conducted to explore how priority settings and explicit constraints affect routing outcomes. These experiments aim to assess how the model responds when service priorities are restructured and whether it can enforce stricter service-level guarantees under user-specified constraints.

In Case A (Table 3.3), the focus is on the relative importance of customers 20 and 66, who previously had similar time windows and were both assigned a high priority level. To examine how the model adapts to priority reassignments, customer 66's priority is elevated to level 2, while customer 20's is demoted to level

Table 3.3: Case A: Customer 66 & Customer 20 priority adjustments

Objective	Actual Waiting Time	Adjusted Waiting Time (WT)	Solution
Minimize WT *(3 Priority levels)*	321.1	20.4	[69, 52, 30, 90, 66, 10, 20, 32, 70, 1]

Priority Level and Waiting Time for 3 Priority Levels

Customer ID	Priority Level	Actual Waiting Time	Adjusted Waiting Time
1	0	141	0
10	0	48	0
20	0	69.8	0
30	1	0	0
32	1	11.6	11.6
52	1	8.8	8.8
66	2	0	0
69	1	0	0
70	0	41.9	0
90	0	0	0

Table 3.4: Case B: Minimizing adjusted waiting times with 3-level priority setting and limited waiting time

Objective	Actual Waiting Time	Adjusted Waiting Time (WT)	Solution
Minimize WT *(3 Priority levels with)* *(limited waiting time)*	459.2	160	[69, 52, 30, 90, 66, 10, 20, 32, 1,70]

Priority Level and Waiting Time for 3 Priority Levels

Customer ID	Priority Level	Actual Waiting Time	Adjusted Waiting Time
1	0	141	0
10	0	48	0
20	2	69.8	139.6
30	1	0	0
32	1	11.6	11.6
52	1	8.8	8.8
66	2	0	0
69	1	0	0
70	0	180	0
90	0	0	0

0. This adjustment reflects a realistic scenario where the operator must respond to an urgent or high-value delivery request (customer 66) while deprioritizing others. The result is a meaningful change in the routing solution. The model reorders the sequence of visits to serve customer 66 earlier, successfully reducing their actual waiting time to zero. Meanwhile, customer 20 experiences a longer delay due to the route adjustment, as expected. This confirms the model's sensitivity to priority parameters: even small changes in priority weights can significantly affect route structure, delivery order, and individual customer experience.

In Case B (Table 3.4), the model's ability to enforce upper-bound constraints on customer waiting time is tested. Both customers 20 and 66 are assigned the same high priority level of 2. However, an additional constraint is imposed to cap the waiting time of customer 66 at 15 units. This case is particularly important in logistics scenarios involving time-critical deliveries such as emergency supplies, where exceeding a delay threshold is unacceptable. The result shows that the model

successfully satisfies the imposed constraint. Customer 66 is served early enough to ensure a waiting time of zero. However, this enforcement introduces an unintended consequence: customer 20, who has a similar service window and competes for early service, now experiences a significant increase in waiting time. In fact, customer 20's adjusted waiting time becomes the dominant component of the total WT in this solution. This demonstrates that although constraint enforcement is feasible, it may lead to degraded overall performance or unbalanced service for other customers, especially when the number of vehicles and time window feasibility are limited.

These two cases illustrate the model's dual capacity to (1) adaptively shift routing behavior based on priority levels and (2) enforce strict service-level constraints when necessary. At the same time, the experiments underscore a critical insight for practical deployment: such adjustments come with trade-offs. Prioritizing one customer, either through higher weights or strict constraints, can lead to disproportionately long delays for others, particularly when vehicle availability and scheduling flexibility are limited. This is further evidenced by the infeasibility observed when the maximum waiting time for both customers 66 and 20 was constrained to 15 units. Although each constraint was individually reasonable, their combined effect over-constrained the solution space, demonstrating how tight service guarantees for multiple overlapping customers can lead to infeasible routing plans. Therefore, effective application of the model in real-world logistics requires careful calibration of both priorities and constraint thresholds to maintain both feasibility and fairness in service distribution.

Thus, the model provides not only a powerful formulation for routing under priority-sensitive environments but also a cautionary framework. Effective application in real-world settings must be guided by an understanding of how priority interactions affect the global solution and by carefully balancing customer needs within operational limits.

3.4 Discussion of Results and Model Behavior

The computational experiments conducted in this section serve two primary purposes: validating the correctness of the proposed MILP formulation for the MO-VRPTWDP and providing insight into the effect of demand priority modeling on routing decisions. The results from Table 3.1 clearly demonstrate the trade-off relationship between the two objectives, total travel distance (TD) and adjusted customer waiting time (WT). When the model is optimized for distance alone, it reproduces the optimal solution from the R201 benchmark, confirming the sound-

ness of the basic routing logic. On the other hand, optimizing for waiting time leads to significantly different routes, with lower service delays but increased total distance. These trade-offs are expected in multi-objective optimization and highlight the need for approaches that can effectively explore the Pareto front.

Further, the experiments show that using the ε-constraint method is an effective technique for identifying solutions under varying objective preferences. As the bound on TD is relaxed, the solution gradually shifts to favor lower WT. Notably, multiple solutions yield the same optimal WT of 7.9, but at varying travel costs, which reveals the presence of multiple trade-off configurations with equivalent performance under the WT objective.

The second set of experiments (Table 3.2) illustrates the impact of customer priority on routing outcomes. When all customers are assigned equal priority, the model minimizes the sum of waiting times, treating each customer equally. However, when priorities are diversified using a three-level scheme, the routes adapt to serve higher-priority customers earlier, resulting in reduced adjusted waiting time (WT), even though the overall actual waiting time increases. This indicates that the model successfully incorporates the user-defined service differentiation, aligning delivery plans with strategic objectives such as responsiveness to critical customers.

A more focused analysis is conducted by adjusting the priorities of two specific customers (20 and 66) who previously had similar service delays. Table 3.3 shows that elevating customer 66 to a higher priority level and demoting customer 20 results in the model reordering the route to favor customer 66, significantly reducing their delay. However, this comes at the cost of increased delay for customer 20. This behavior exemplifies the flexibility of the soft-priority model and demonstrates how small changes in the priority parameters can meaningfully influence routing behavior.

To further examine the model's ability to enforce fairness constraints, the waiting time constraint from Eq.(19) is introduced. In the final experiment (Table3.4), a maximum waiting time of 15 is imposed on customer 66, while both customers 66 and 20 are assigned the same priority level. The resulting route complies with the waiting time limit for customer 66 but does so by significantly delaying other customers, including customer 20. This trade-off highlights both the effectiveness and the limitations of constraint-based service guarantees. The ability to enforce maximum delays is powerful but can also reduce flexibility, particularly when multiple high-priority customers share overlapping time windows. In this specific instance, tightening the constraint further led to infeasibility, underscoring the operational risks of overly rigid service constraints.

Overall, the experiments confirm that the proposed MILP model is valid, interpretable, and capable of expressing real-world service differentiation objectives. It provides a flexible foundation for routing decisions that must account for multiple conflicting goals, such as cost efficiency and customer satisfaction. Moreover, it enables users to shape service behavior not only through objective weighting but also through hard constraints on individual customer performance, making it a versatile tool for logistics planning. These findings justify the development of metaheuristic algorithms in later chapters to handle larger problem instances with the same flexibility and decision-making support.

3.5 Chapter Remarks

This chapter introduced the formulation of the Multi-Objective Vehicle Routing Problem with Time Windows and Demand Priority (MO-VRPTWDP), a novel extension of the classical VRPTW that incorporates soft demand priority into the optimization framework. The chapter began by motivating the need for differentiated service levels in modern logistics and defined the concept of demand priority as a soft constraint based on weighted customer waiting times.

A Mixed Integer Linear Programming (MILP) formulation was presented, capturing the dual objectives of minimizing total travel distance and adjusted waiting time while respecting standard VRPTW constraints such as time windows and vehicle capacities. An additional optional constraint was introduced to enforce upper bounds on individual customer waiting times, providing greater modeling flexibility for real-world applications that demand strict service guarantees.

To validate the correctness and behavior of the model, a series of small-scale computational experiments was conducted using a modified Solomon R201 instance. These experiments demonstrated the model's ability to produce trade-offs between cost and service quality, to adapt routing behavior based on priority settings, and to enforce user-defined service constraints. The results confirmed that adjusting priority weights and applying waiting time limits can significantly influence routing decisions and customer experience, sometimes leading to infeasibility when constraints are overly strict.

Overall, this chapter establishes the theoretical and practical foundation for the remainder of the dissertation. The next chapters focus on designing and evaluating scalable metaheuristic algorithms MTSA and RL-MTSA to solve large-scale instances of MO-VRPTWDP efficiently while preserving its ability to reflect user preferences and real-world delivery constraints.

Chapter 4

Multi-Thread Simulated Annealing

This chapter presents the *Multi-Thread Simulated Annealing (MTSA)* algorithm developed to address the Multi-Objective Vehicle Routing Problem with Time Windows and Demand Priority (MO-VRPTWDP) at scale. While classical Simulated Annealing (SA) offers strong local refinement and simplicity, it struggles with maintaining diversity and effective exploration across complex Pareto frontiers. MTSA enhances exploration, exploitation, and convergence stability by employing parallel, cooperative search threads with adaptive weight strategies, enabling efficient high-quality Pareto Frontier Approximation (PFA) within practical computational budgets.

The chapter is structured as follows: Section 4.1 details the MTSA methodology, Section 4.2 presents computational experiments on benchmark instances, Section 4.3 discusses practical implications and insights from MTSA performance, and Section 4.4 concludes the chapter.

4.1 MTSA Methodology

The proposed Multi-Thread Simulated Annealing (MTSA) algorithm is developed to address multi-objective optimization problems by enhancing both exploration and exploitation across the Pareto Frontier Approximation (PFA). The core idea behind MTSA is to enable each Simulated Annealing (SA) thread to explore distinct regions of the PFA by assigning different weights to the objectives, thereby steering each thread's search trajectory according to specific trade-off preferences.

Within MTSA, two types of threads are defined based on their acceptance strategies. The first type, referred to as the Control thread, operates with explicitly as-

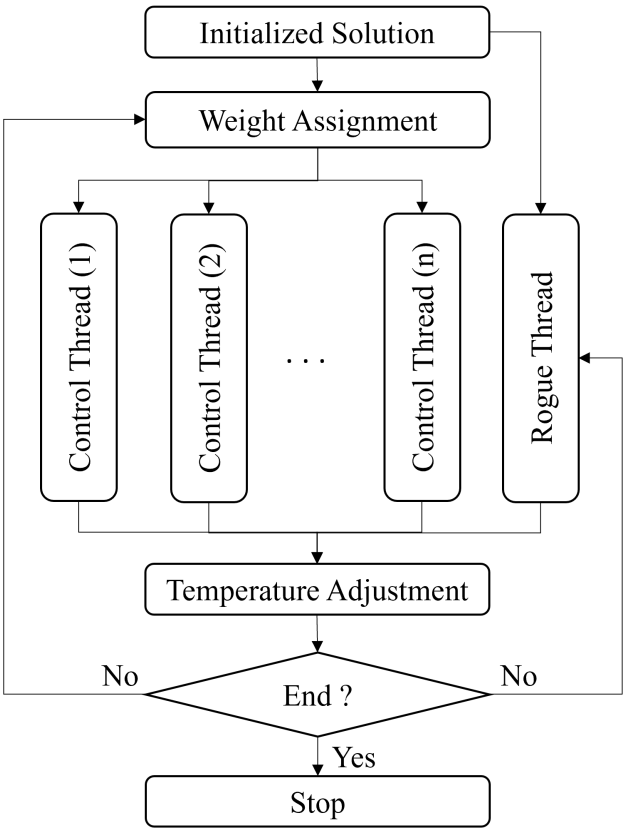

Figure 4.1: Overview of the MTSA algorithm

signed weight configurations on objectives. A Control thread accepts a new solution under three conditions: (1) if the new solution is non-dominated with respect to the current Pareto archive or dominates an existing non-dominated solution, (2) if it improves the weighted aggregate objective values, or (3) if it meets a probabilistic acceptance criterion, allowing occasional acceptance of inferior solutions to enhance exploration.

The second type, termed the Rogue thread, functions without fixed weight assignments. Instead, it applies acceptance rules designed to encourage broader exploration: (1) accepting new solutions that are non-dominated or dominate existing archive members, (2) accepting solutions that improve at least one objective while the other remains within a specified threshold, and (3) probabilistically accepting inferior solutions to prevent premature convergence.

To further enhance exploration, MTSA incorporates a return-to-point strategy that is activated when a thread fails to achieve improvements in the weighted objectives or cannot find a solution that dominates any member of the Pareto archive within a defined number of iterations. This strategy repositions the thread to a promising solution within the archive, selected based on either its location in the

largest gap on the PFA or its duration of persistence in the archive, indicating high potential for further improvement.

An overview of the MTSA workflow is illustrated in Figure 4.1. The process begins with the initialization of feasible solutions and the assignment of initial weight configurations to each thread. Throughout the iterative search process, weight configurations are periodically updated based on predefined conditions, and the acceptance probabilities within the SA mechanism are dynamically adjusted to balance exploration and exploitation. The algorithm proceeds iteratively across all threads until either the maximum iteration limit is reached or a termination condition is satisfied.

The detailed mechanisms and algorithmic components of MTSA are described in the following sections.

4.1.1 Initial Solution Generation and Infeasibility Handling

To initialize the VRPTWDP within the MTSA framework, this study employs a modified heuristic based on the well-known Time Window Insertion Heuristic (TWIH). Traditionally, TWIH constructs routes by sequentially adding customers in the order of their earliest arrival times until vehicle capacity or time window constraints are violated. While effective for generating feasible initial solutions, the standard TWIH does not explicitly consider total travel distance during the insertion process.

To address this limitation, the modified TWIH introduced in this study incorporates distance considerations to improve initial route quality. The process begins by identifying customers whose earliest feasible arrival times align with the current route's schedule while respecting time window and capacity constraints. The algorithm then inserts the customer nearest (in Euclidean distance) to the previously added customer (or the depot for the first insertion) into the route, prioritizing travel distance minimization during construction. This insertion process continues until the route reaches its capacity limit, at which point a new route is initiated to continue serving unassigned customers.

If the maximum number of allowable routes is reached and there are still unassigned customers, these customers are forcibly inserted into existing routes in a manner that minimizes the total travel distance across all routes. This forced insertion does not consider time window or capacity feasibility, as the primary goal is to produce a complete initial solution for the subsequent repair and improvement process within MTSA. The insertion order for these remaining customers is fixed based on ascending customer indices (e.g., [10, 15, 25] are inserted sequentially) since in-

feasibility is inevitable under forced insertion, and the emphasis is on maintaining computational efficiency during initialization.

To guide the algorithm toward feasibility during the search, MTSA incorporates a penalty mechanism for infeasible solutions. Solutions containing infeasible nodes are penalized in the weighted sum objective function using the following formulation:

$$\text{Weighted Objective} = \text{Weighted Objective} \times (1 + \gamma \cdot n), \tag{4.1}$$

where n represents the number of infeasible nodes in the solution, and γ is a penalty coefficient that scales the objective value in proportion to the level of infeasibility. In this study, γ is empirically set to 0.0175 based on parameter tuning experiments. This penalization discourages the algorithm from retaining infeasible solutions, effectively guiding MTSA to prioritize reducing the number of infeasible nodes during the search process while simultaneously improving the quality of feasible routes. It is important to note that infeasible solutions are excluded from inclusion in the Pareto Frontier Approximation (PFA), ensuring that the archive remains valid throughout the optimization process.

4.1.2 Weight Assignments and Weight Adjustments

To compute the aggregate weighted objective in MTSA, weights are assigned to each objective, enabling threads to explore distinct trade-offs along the Pareto front. Initially, weights are uniformly distributed across all threads to ensure broad coverage of the solution space. The weighted sum objective is defined as:

$$\begin{aligned} &\text{Weighted Objective} = w_{TD}(TD) + w_{WT}(WT), \\ &\text{where} \quad w_{WT} = 1 - w_{TD}. \end{aligned} \tag{4.2}$$

However, uniform weight allocation may lead to imbalanced exploration if there is a significant disparity in the number of Pareto members contributed by each control thread. To mitigate this issue, a weight adjustment mechanism is incorporated to adaptively modify thread weights, encouraging the exploration of underrepresented regions on the Pareto Frontier Approximation (PFA).

This weight adjustment mechanism consists of two primary steps. First, it identifies threads with fewer PFA contributions than a predefined threshold. In this

study, the threshold is set at 10% of the total number of PFA members, marking threads with contributions below this level as candidates for weight adjustment. If no such candidate is found, the weight adjustment process is skipped for that iteration.

Once a candidate thread is identified, the algorithm proceeds to locate the largest gap on the current PFA, defined as the greatest Euclidean distance between consecutive Pareto solutions. The weight for the selected thread is then adjusted to target this underexplored region, thereby enhancing front diversity and improving overall coverage. If no significant gap is detected on the PFA, the algorithm assigns a random weight within the interval $w_{TD} \in [0.25, 0.75]$, enabling exploration near potential knee or elbow points on the Pareto frontier while maintaining flexibility in search direction.

4.1.3 Neighborhood Search Structure

The neighborhood search structure is a critical component influencing the effectiveness and efficiency of the Multi-Thread Simulated Annealing (MTSA) algorithm. It enables iterative exploration of the solution space by systematically generating new candidate solutions, impacting both the computational efficiency and the quality of solutions obtained during the search.

Given the complexity of VRPTWDP, employing specialized neighborhood operators that respect the unique constraints and objectives of the problem is essential. Effective neighborhood strategies should be capable of exploring diverse solution variations while adhering to the time window and capacity constraints inherent in VRPTWDP instances.

In this study, the MTSA algorithm utilizes several neighborhood operators to enhance search effectiveness:

- **Swap:** The Swap operator exchanges the positions of two customer nodes within a route or across routes, facilitating exploration of alternative service sequences. Various strategies are used to determine the first node for swapping, including:
  - Selecting the node with the longest waiting time within its route.
  - Choosing a node that currently violates time window or capacity constraints; if multiple infeasible nodes exist, one is selected at random.
  - Identifying the node with the longest idle time between visits.
  - Selecting the node with the longest available time window.

- Choosing a node from the route that incurs the longest travel distance.
- Random selection from the route.

For selecting the second node in the swap operation, several criteria can be applied:

- Applying shifting moves within the route.
- Performing swaps based on time window alignment.
- Swapping with a node from another route to reduce travel distance.
- Random selection.

- **Insert:** The Insert operator removes one or more customer nodes from their current positions and reinserts them at new positions within the same route or across different routes. The selection of nodes for removal and the determination of insertion points follow strategies similar to those used in the Swap operator, allowing flexibility in exploring alternative route structures.
- **Multiple-Insert:** The Multiple-Insert operator begins by selecting a starting customer node. A score is then calculated for the remaining nodes based on their distance and time window differences relative to the start node, where a lower score indicates closer spatial and temporal proximity. Nodes with the lowest scores are selected for removal, after which they are reinserted sequentially into feasible positions within the routes, ensuring that both distance efficiency and time window feasibility are considered during reinsertion.

The integration of these neighborhood operators within the MTSA framework enables systematic exploration of the solution space, balancing intensification and diversification to improve the quality of the Pareto Frontier Approximation while maintaining computational efficiency in large-scale VRPTWDP instances.

4.1.4 Control Thread

The control thread within the MTSA framework is responsible for systematic exploration and refinement of solutions, utilizing key components including neighborhood search, non-dominated archive updates, acceptance evaluation, and a return-to-point mechanism, as illustrated in Figure 4.2.

The process begins with the generation of a new candidate solution through a neighborhood search applied to the current solution. This new solution is then evaluated using the weighted sum of objectives based on the thread's assigned weight

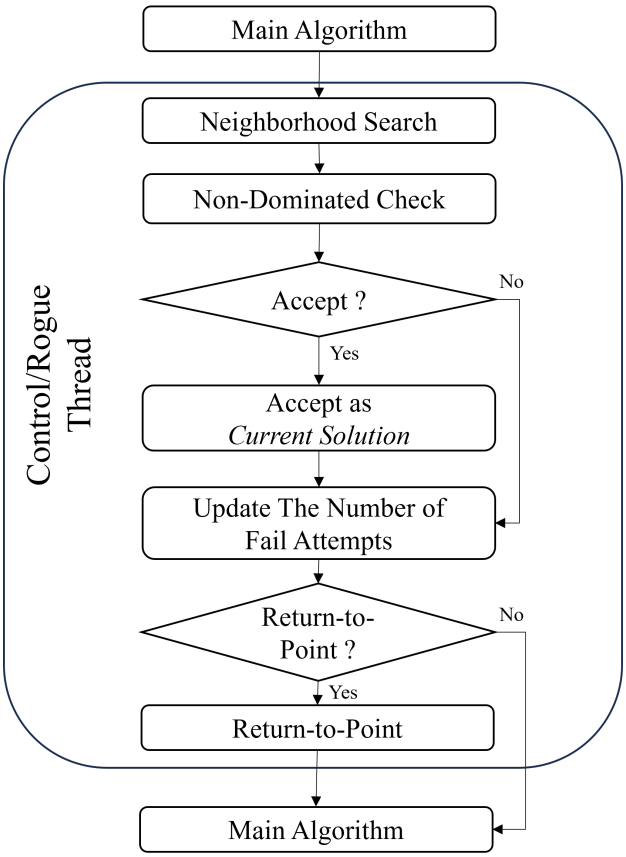

Figure 4.2: Structure of Control and Rogue Threads

configuration. Following evaluation, the algorithm checks whether the candidate solution dominates any existing members of the current Pareto archive.

A new solution may be accepted as the current solution for the next iteration if it satisfies at least one of the following criteria:

1. The candidate solution either dominates members of the Pareto archive or remains non-dominated with respect to the archive.

2. The weighted objective value of the candidate solution is less than or equal to that of the current solution.

3. The solution is accepted probabilistically, with acceptance determined by a random value falling below a threshold acceptance probability (α).

If the control thread does not identify any improvement within a specified number of iterations, the *return-to-point policy* is activated to reinitialize the search from a more promising point. Two strategies are employed within this policy:

- **Gap Exploration Strategy:** The thread selects a solution located within

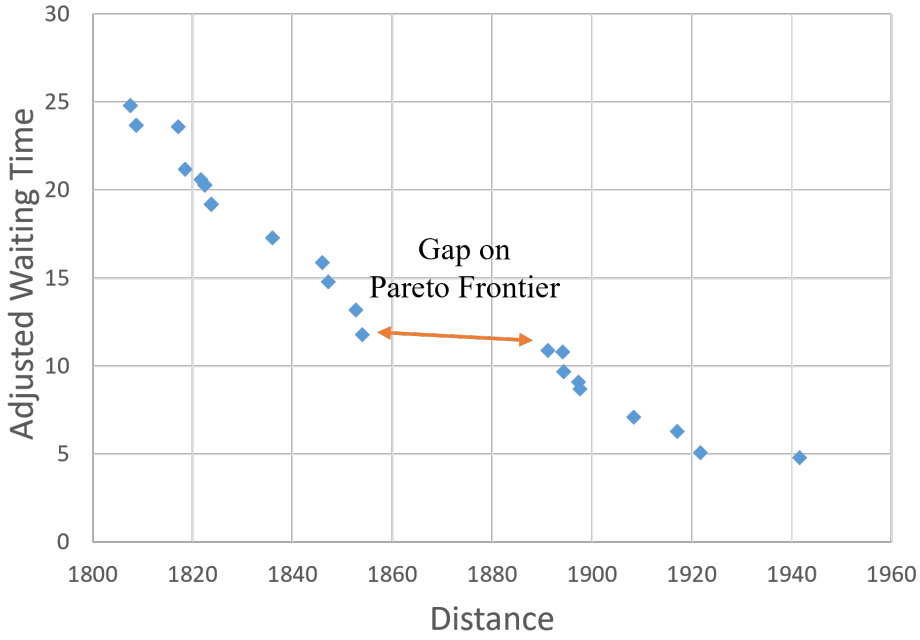

Figure 4.3: Illustration of the Pareto Front Gap Selection in the Return-to-Point Policy

the largest gap on the current Pareto front, encouraging the exploration of underrepresented regions (Figure 4.3).

- **Best Solution Exploitation Strategy:** The thread reinitializes to the best solution found by that thread to intensify search around high-quality regions, promoting deeper local refinement.

By alternating between these two strategies, the control thread maintains a balance between exploration of new regions on the Pareto front and exploitation of already promising regions, contributing to a well-distributed and high-quality Pareto Frontier Approximation within the MTSA framework.

4.1.5 Rogue Thread

The structure of the rogue thread closely mirrors that of the control thread shown in Figure 4.2, but with key differences in solution evaluation, acceptance criteria, and return-to-point strategies to align with its unique objectives. Specifically, the rogue thread is designed to enhance the exploration of the Pareto frontier by:

- Filling large gaps within the frontier,
- Extending extreme points along the frontier,

- Exploiting regions that may be underexplored by the control threads.

The process begins with the generation of a new candidate solution using neighborhood search applied to the current solution. This candidate is then evaluated for dominance against the existing Pareto archive.

A new solution may be accepted as the current solution for the next iteration if it meets at least one of the following conditions:

1. The candidate either dominates one of the solutions in the current Pareto archive or remains non-dominated with respect to the archive.

2. The candidate improves at least one objective relative to the current solution while ensuring that the other objectives do not deteriorate beyond a pre-specified threshold. For example, if acceptance is evaluated based on minimizing total distance, the acceptance conditions are:

$$TD_{New} \leq TD_{Current}$$

$$WT_{New} \leq WT_{Current} \times (1 + \beta)$$

   where β is a tolerance parameter ranging from 0 to 1.

3. The solution is accepted based on a probabilistic criterion if a randomly generated value falls below the acceptance threshold (α).

If the rogue thread fails to find a qualifying non-dominated solution within a specified number of iterations, a *return-to-point policy* is invoked to reinitialize the search from a new position. The rogue thread employs several strategies for this purpose:

- Selecting a solution located in the largest gap of the Pareto front to prioritize exploration of underrepresented regions.

- Expanding the front by selecting a solution with the lowest value in a target objective (either minimum total distance or minimum adjusted waiting time), with the objective randomly selected each time.

- Choosing the solution from the Pareto archive that has remained unchanged for the longest period, thereby promoting exploration in potentially stagnant regions.

- Randomly selecting a solution from the Pareto archive as the new current solution.

The choice among these return-to-point strategies is made randomly based on a predefined probability distribution, ensuring that the rogue thread maintains a flexible balance between targeted exploration and broad search diversity within the MTSA framework.

4.1.6 Temperature Adjustment

Temperature adjustment is a fundamental component within Simulated Annealing (SA) algorithms, serving to gradually decrease the acceptance probability (α) of inferior solutions as the search progresses. This mechanism is essential in SA since the occasional acceptance of worse solutions helps the algorithm escape local optima during the early stages of the search. However, as the algorithm advances, reducing the acceptance rate becomes increasingly important to steer the search toward high-quality solutions.

Within the MTSA framework, if a control thread fails to discover an improved solution within a predefined number of iterations, thereby triggering the return-to-point mechanism, a temporary increase in temperature is applied. Specifically, the temperature is increased by a small percentage (typically 10%) while ensuring it does not exceed the initial temperature value. This temporary temperature increase enables the algorithm to reintroduce diversity into the search process, helping to mitigate the risk of stagnation or premature convergence when the temperature has become too low for effective exploration.

The 10% increment value is determined based on empirical fine-tuning during parameter calibration. Although the temperature bounce mechanism does not substantially impact the overall performance of MTSA, it serves as a simple yet effective diversification strategy to enhance search robustness during periods of stagnation.

4.2 Computational Experiment

This section presents a structured examination of the computational experiments conducted in this study, including the description of benchmark instances, performance evaluation of the MTSA algorithm, and comparative analysis with baseline methods. Additionally, an investigation into the optimal configuration of the MTSA algorithm, specifically the number of control threads, is performed to enhance practical implementation.

To assess the effectiveness of the proposed MTSA algorithm, its performance is compared against the original Multi-Objective Simulated Annealing (MOSA) al-

gorithm. The MOSA implementation utilized in this study is adapted from the original version proposed by Suppapitnarm et al. (2000) and modified for the MO-VRPTWDP context. Modifications include enhancements to the acceptance criteria and return-to-point mechanisms to align with the current study's objectives. Conceptually, the MOSA algorithm can be interpreted as equivalent to a single Rogue Thread operating within the MTSA framework presented in this dissertation. All computational experiments were executed on a 12th Gen Intel(R) Core(TM) i7-12650H 2.30 GHz processor.

4.2.1 Benchmark Instances

The computational experiments utilize the widely recognized Solomon benchmark set for the VRPTW (Solomon, 1987), which comprises 56 instances, each featuring a single depot and 100 customer nodes. Each node, including the depot, is characterized by defined earliest and latest service times, as well as specified service durations. Constraints on the maximum number of vehicles are imposed, with each vehicle having an identical maximum capacity.

These benchmark instances are organized into six categories: C1, C2, R1, R2, RC1, and RC2. In categories C1 and C2, customer locations are clustered, whereas in R1 and R2, customers are randomly distributed. Categories RC1 and RC2 represent a hybrid of clustered and random customer distributions. It is noted that instances in C1, R1, and RC1 typically feature tighter time window constraints compared to C2, R2, and RC2, which have relatively relaxed time windows. In this benchmark set, the travel cost C_{ij} and travel time T_{ij} between nodes are both defined as the Euclidean distances between the respective node pairs. As a result, distance and time share the same unit and are inherently on a common scale. Consequently, additional normalization of the weighted objective function is not necessary in this benchmark-based evaluation.

For the small-scale validation experiment, instance R201 from the benchmark set is selected due to its flexible time window characteristics, which are suitable for illustrating the impact of priority level adjustments on customer waiting times within the proposed model. To evaluate the comparative performance of MTSA and MOSA, a uniform priority level of 1 is assigned to all customers across all instances, indicating that the waiting time for each customer is considered equally within the experiments.

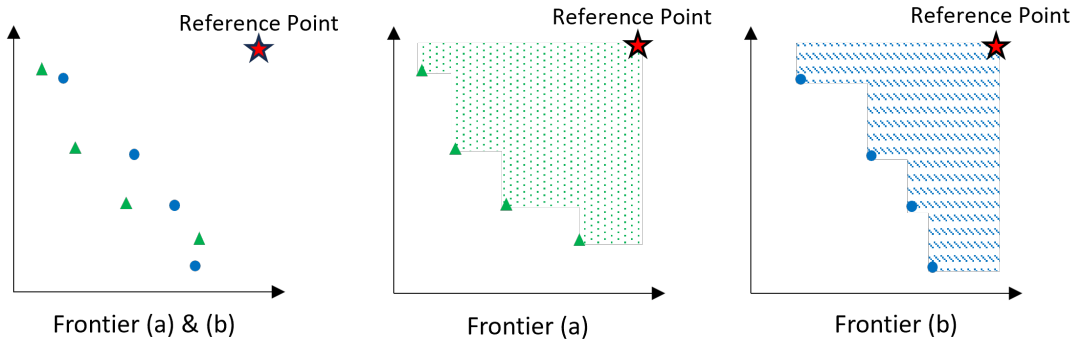

Figure 4.4: Illustration of hypervolume calculation for two PFAs

4.2.2 Performance Measurement for MTSA

To evaluate the performance of the MTSA algorithm, two key metrics are employed: computational time and solution quality. However, since the VRPTWDP is an NP-hard problem, the exact Pareto optimal front is unknown, making direct comparisons against a true Pareto front impractical. To address this, the hypervolume (HV) metric is used as a proxy for assessing the progress of the Pareto Front Approximation (PFA), following the approach of Baños et al. (2013).

Hypervolume measures the volume in the objective space that is dominated by the non-dominated solutions in the current front, relative to a predefined reference point that bounds the calculation space. As the PFA improves, the covered hypervolume increases, allowing for a quantitative comparison of convergence progress between different algorithms or configurations. Figure 4.4 illustrates this concept in a bi-objective minimization context, where Front (a) dominates a larger portion of the objective space than Front (b), resulting in a higher HV value. It should be noted that this figure is provided for illustrative purposes and does not represent the actual experimental results.

In this study, the reference point for hypervolume calculation is defined as:

$$\begin{aligned}\text{Reference Point} = (&2 \times \text{Initial Distance},\\ &\max(2 \times \text{Initial Adjusted Waiting Time},\\ &2 \times \text{Max Adjusted Waiting Time}))\end{aligned}$$

Here, the maximum adjusted waiting time refers to the waiting time observed in the optimal solution of the corresponding VRPTW instance. The computed HV values are expressed as a percentage of the area covered relative to the optimal boundary, which is defined by the region between the reference point and the optimal VRPTW solution. This approach enables consistent comparisons of PFA progress across different algorithm configurations.

In addition to hypervolume, the Schott Spacing metric (Schott, 1995) is used to assess the uniformity of solution distribution along the PFA. Unlike hypervolume, which primarily measures convergence and coverage, Schott Spacing evaluates the evenness of solution dispersion across the front. It is calculated as the standard deviation of the Euclidean distances between each solution and its nearest neighbor on the PFA, with lower values indicating more evenly distributed solutions. This property is advantageous in multi-objective optimization, as it provides decision-makers with a diverse and balanced set of trade-offs.

The inclusion of Schott Spacing offers a complementary perspective when evaluating algorithm performance, especially in cases where different configurations achieve similar hypervolume values but vary in distribution uniformity. In this study, Schott Spacing is used alongside hypervolume to comprehensively assess the diversity and distribution quality of the PFA generated by the MTSA algorithm and to compare its performance with the baseline MOSA implementation.

4.2.3 MTSA and MOSA Performance Comparison

This section evaluates the effectiveness of the proposed MTSA algorithm under various configurations and benchmarks its performance against a modified version of the MOSA algorithm. To ensure a fair comparison, both MTSA and MOSA utilize an identical neighborhood search structure and problem encoding, with the primary difference being the number of control threads implemented. MTSA is tested using four configurations, employing between three and six threads, where each setup includes one rogue thread while the remaining threads function as control threads.

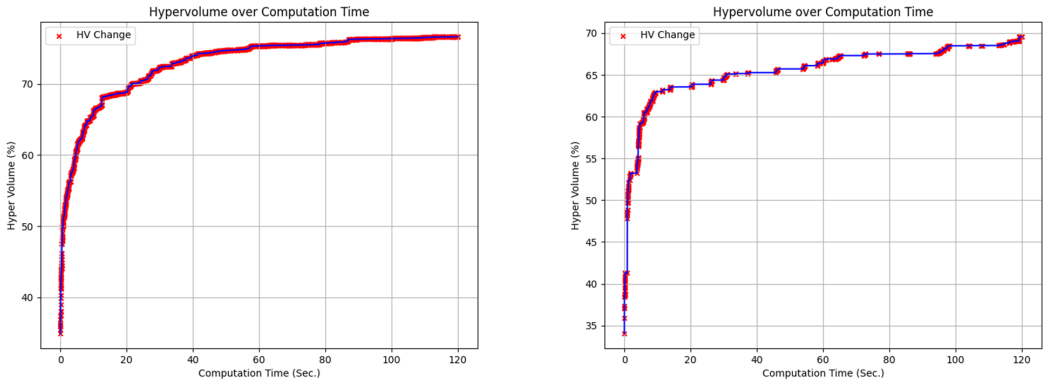

Figure 4.5: Convergence behavior of MTSA (left) and MOSA (right) in terms of hypervolume over computational time.

For consistency, the runtime for each configuration is fixed at 120 seconds. This time limit provides a practical computational budget and enables a fair comparison across algorithms. As shown in Fig. 4.5, both MTSA and MOSA achieve rapid improvements in solution quality early in the search, followed by a plateau with marginal hypervolume gains, indicating that 120 seconds is sufficient for convergence and performance evaluation.

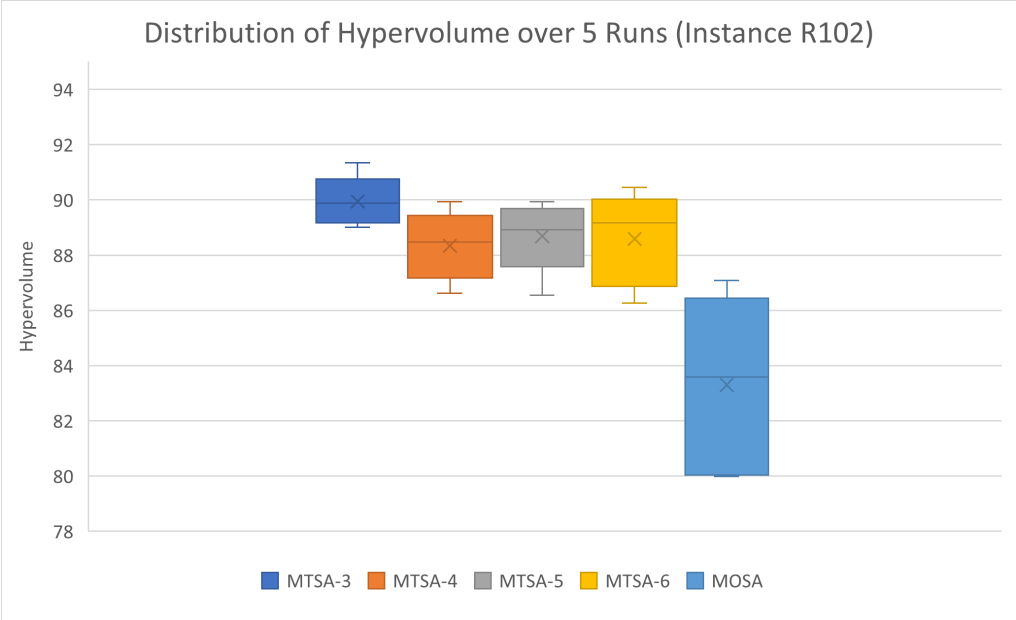

Figure 4.6: Box plot of hypervolume values for MTSA and MOSA over five independent runs on instance R102.

Testing is conducted using 56 benchmark instances, with each instance executed five times to evaluate the performance of MTSA and MOSA. As illustrated in Fig. 4.6, five independent runs are sufficient to distinguish the performance difference between MTSA and MOSA.

Additionally, results obtained under the 120-second fixed runtime are compared to results from a one-hour runtime, which serves as a near-optimal PFA reference to illustrate the potential of MTSA under extended computation.

Table 4.1 presents the results for the 120-second runtime, reporting the average HV values, the range of HV values, and the maximum HV achieved over five independent runs. The results indicate that the MTSA configuration with three threads achieves the highest average HV and attains the highest maximum HV in 29 out of the 56 instances, demonstrating superior performance over MOSA and other MTSA configurations. Figure 4.7 depicts the best non-dominated fronts obtained within the 120-second runtime for selected instances (C101, C201, R101, R201, RC101, and RC201) for both MTSA with three threads (MTSA-3) and MOSA,

alongside the best front achieved by MTSA with four threads (MTSA-4) under a one-hour runtime. The comparison reveals that MTSA-3 consistently outperforms MOSA in terms of front progression within the same computation time and closely approaches the quality of the fronts achieved by MTSA-4 over the extended runtime, demonstrating the efficiency of MTSA in generating high-quality solutions in shorter periods.

Further analysis shows that the variation in average HV values across MTSA configurations is minimal, with a range of 1.56. In contrast, when including MOSA in the comparison, the average range increases to 9.57, highlighting the consistent and stable performance of MTSA configurations relative to MOSA, which consistently underperforms across all configurations. Additionally, the variation between the minimum and maximum HV values within each MTSA configuration is recorded as 2.73, 2.40, 2.47, and 2.56 for the configurations with three to six threads, respectively, whereas MOSA exhibits a larger average range of 4.86. These observations affirm the stability and reliability of MTSA in exploring the non-dominated front compared to MOSA.

Table 4.2 presents the average Schott Spacing values across each instance group, further demonstrating the performance of MTSA. Across all tested instance groups, all MTSA configurations exhibit significantly lower Schott Spacing values compared to MOSA, indicating superior distribution uniformity along the Pareto Front Approximation (PFA). Within the MTSA configurations, no single variant consistently achieves the lowest spacing across all instance groups, suggesting that configurations from MTSA-3 to MTSA-6 deliver comparable solution distribution performance. The marginal differences in spacing among MTSA configurations reinforce the findings from the HV analysis, confirming that MTSA is not only effective in generating high-quality solutions but also robust and reliable across different configurations.

In summary, the results from both the hypervolume and Schott Spacing analyses validate the proposed MTSA framework's effectiveness in delivering high-quality, well-distributed PFAs, which is essential for practical, multi-objective decision-making in real-world logistics applications.

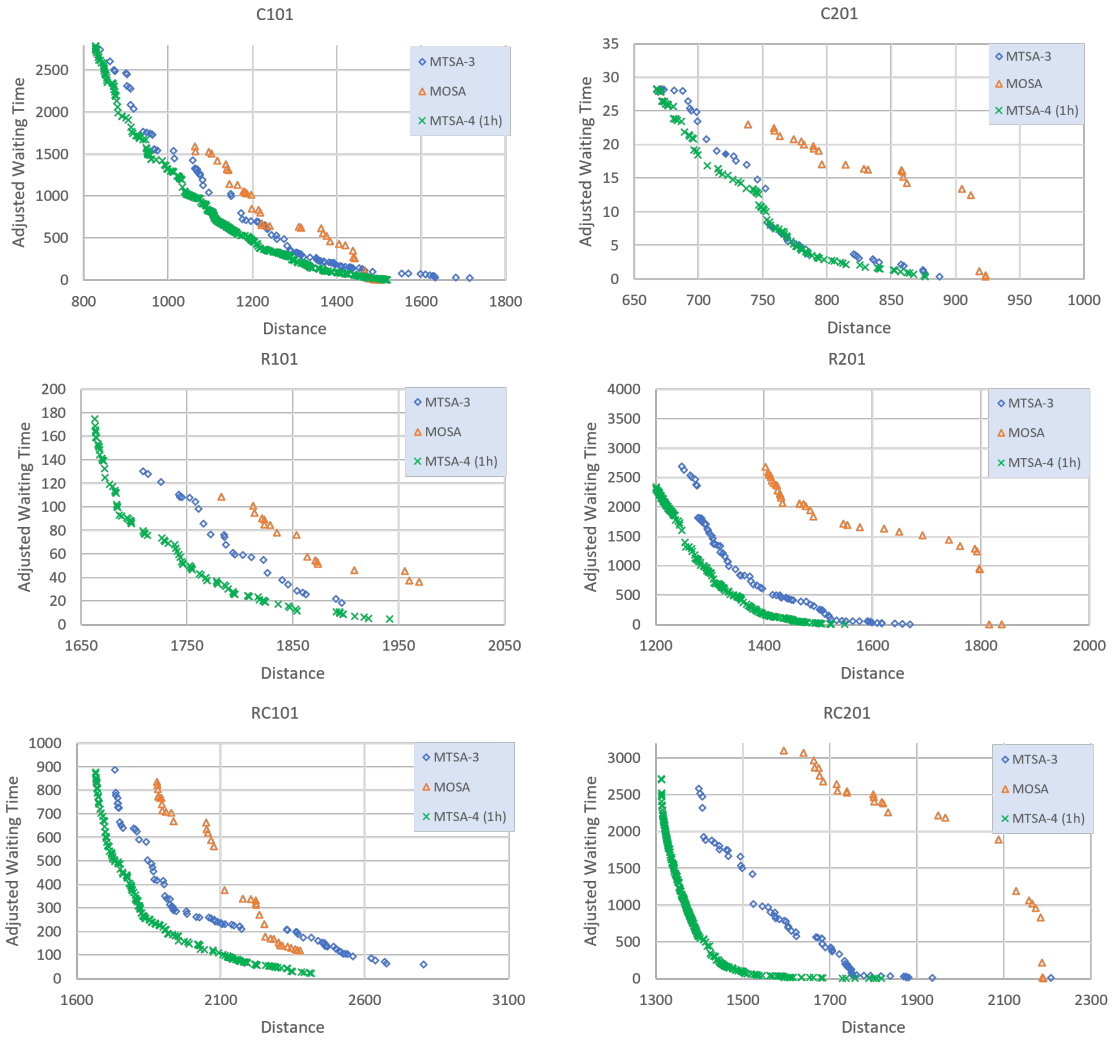

Figure 4.7: Best PFA obtained from 120 s for MTSA-3 and MOSA and 1-hour MTSA-4

Table 4.1: MTSA and MOSA results from the fixed 120-sec runtime

Instance	MTSA-3			MTSA-4			MTSA-5			MTSA-6			MOSA		
	Avg(HV)	*Max(HV)*	*Range*	*Avg(HV)*	*Max(HV)*	*Range*	*Avg(HV)*	*Max(HV)*	*Range*	*Avg(HV)*	*Max(HV)*	*Range*	*Avg(HV)*	*Max(HV)*	*Range*
c101	95.14	96.16	1.41	94.55	94.98	1.72	95.01	95.50	0.97	94.56	96.12	2.41	89.56	91.14	4.17
c102	90.64	91.45	1.56	90.73	91.50	1.50	90.22	90.65	0.93	89.43	91.38	2.97	85.80	86.87	2.34
c103	88.34	89.01	1.44	87.54	88.14	1.67	87.60	88.30	1.21	86.96	87.60	1.32	81.77	84.29	5.05
c104	82.12	82.74	1.36	80.24	81.30	2.27	80.88	81.78	2.29	81.06	81.87	1.61	72.34	76.73	8.20
c105	94.77	95.86	2.06	94.72	95.73	3.53	94.61	96.03	3.65	94.23	95.94	3.57	90.02	92.77	5.84
c106	94.45	95.42	1.51	94.68	95.69	1.85	94.58	95.31	1.31	93.86	94.90	2.19	91.31	92.35	2.77
c107	96.69	97.01	0.74	96.49	96.94	1.01	96.23	96.90	1.40	96.31	96.82	0.94	90.41	92.57	4.45
c108	93.28	94.24	2.16	93.30	94.30	2.49	93.44	94.78	2.00	93.37	93.63	0.70	90.13	92.14	4.90
c109	92.94	93.65	1.30	92.97	93.71	1.72	92.29	93.13	1.59	92.39	92.72	0.50	88.18	90.57	4.61
c201	98.97	99.25	1.35	98.94	99.23	1.41	98.18	99.22	2.03	98.40	99.25	1.85	94.51	95.80	2.46
c202	92.37	93.35	2.25	91.46	93.12	3.12	90.85	92.48	3.61	90.06	90.43	0.96	82.38	84.88	8.03
c203	89.79	90.11	0.59	88.66	89.78	1.78	88.22	88.92	1.47	87.50	88.57	2.14	71.64	76.23	9.79
c204	87.62	88.30	1.28	86.45	87.26	1.30	85.50	86.05	2.02	85.04	85.84	1.49	61.34	67.34	11.52
c205	97.68	98.65	1.66	97.93	98.55	1.24	97.71	98.81	3.61	97.97	98.44	1.37	91.11	94.70	8.93
c206	96.76	97.59	1.67	96.24	97.22	2.49	96.80	97.50	1.55	96.52	97.15	1.52	89.24	91.73	9.25
c207	95.18	96.70	5.43	95.44	96.49	1.90	95.82	96.40	1.26	95.88	96.11	0.50	85.84	88.03	4.28
c208	97.27	97.86	1.46	97.43	97.79	1.21	97.16	97.54	1.22	97.19	97.86	1.49	88.35	91.49	6.78
r101	92.91	96.38	6.78	93.17	94.27	1.51	91.86	93.72	3.38	93.62	94.84	4.35	91.37	92.81	3.91
r102	89.95	91.34	2.34	88.34	89.93	3.31	88.69	89.93	3.37	88.59	90.46	4.19	83.31	87.09	7.10
r103	81.55	82.68	2.10	79.71	81.45	4.20	80.45	81.22	1.44	80.42	81.45	2.10	74.21	78.26	8.15
r104	76.98	77.40	0.62	75.90	77.21	4.15	76.57	77.08	1.06	76.42	76.74	0.62	70.60	71.50	1.92
r105	90.41	91.66	2.90	90.72	91.75	2.16	89.90	90.77	2.72	90.34	91.06	1.74	85.77	86.89	3.30
r106	86.86	87.19	0.96	86.82	87.43	1.44	86.46	87.28	2.51	86.09	87.06	1.97	81.47	82.43	1.50
r107	79.51	80.09	1.18	78.96	79.30	0.90	78.90	80.15	2.16	78.62	80.00	1.91	72.98	73.74	1.05
r108	76.09	76.60	1.29	75.47	75.90	1.07	75.66	76.17	1.06	75.45	76.39	1.28	70.63	71.58	2.01
r109	90.71	91.22	1.18	90.10	91.31	2.21	90.70	91.74	1.80	90.32	91.06	1.47	84.79	86.69	4.31
r110	86.09	86.67	1.81	86.01	87.39	2.34	85.45	86.49	2.55	85.69	86.93	2.12	78.78	79.90	3.04

Continued on next page

Table 4.1 (continued)

Instance	MTSA-3			MTSA-4			MTSA-5			MTSA-6			MOSA		
	Avg(HV)	*Max(HV)*	*Range*	*Avg(HV)*	*Max(HV)*	*Range*	*Avg(HV)*	*Max(HV)*	*Range*	*Avg(HV)*	*Max(HV)*	*Range*	*Avg(HV)*	*Max(HV)*	*Range*
r111	89.12	89.66	1.71	88.90	89.96	2.61	88.21	89.24	2.55	87.41	88.13	1.22	79.59	80.77	2.54
r112	81.70	82.33	1.01	81.65	82.35	1.49	81.44	82.13	1.41	81.84	82.60	1.37	72.80	75.06	4.30
r201	92.24	94.54	6.87	92.56	93.90	4.06	93.36	94.04	2.08	92.10	93.40	3.26	85.60	86.39	1.57
r202	89.55	93.94	8.06	92.80	93.67	1.93	92.60	93.77	5.35	91.10	91.42	0.60	80.16	80.73	1.37
r203	92.84	93.73	2.07	92.65	93.61	2.00	89.90	91.61	7.16	91.19	92.44	3.04	79.99	81.88	4.67
r204	91.62	92.35	1.98	90.43	90.87	0.84	90.28	90.53	0.53	89.47	90.56	1.86	79.07	80.20	2.31
r205	91.56	92.92	2.91	90.23	92.59	6.59	90.46	92.16	3.13	90.87	91.38	1.40	78.41	80.65	7.38
r206	87.32	89.29	5.55	88.77	90.46	3.63	89.12	89.90	2.35	85.23	89.17	9.92	73.16	74.28	2.91
r207	88.68	91.42	7.27	88.74	90.84	3.75	86.85	89.66	9.33	86.69	88.86	6.42	72.61	75.01	5.49
r208	90.38	91.49	2.29	89.08	89.94	1.68	88.97	89.74	2.17	88.46	89.27	1.37	73.11	76.47	5.70
r209	91.45	92.45	1.83	89.95	92.06	7.50	90.01	90.89	1.62	89.66	90.49	1.92	77.76	79.98	5.49
r210	92.96	93.89	2.21	92.87	93.64	2.42	91.39	93.23	3.89	90.21	92.41	9.04	78.69	81.82	8.79
r211	89.42	90.46	2.14	88.54	89.58	1.91	88.16	89.36	2.53	88.11	89.75	3.24	67.64	68.63	2.42
rc101	90.58	91.77	2.54	90.33	92.33	3.60	90.51	91.46	2.41	89.53	90.16	1.30	81.77	85.58	7.20
rc102	82.66	84.70	5.76	84.34	85.07	1.55	83.60	84.53	2.54	82.81	84.53	3.51	76.50	78.76	4.46
rc103	*77.93*	*78.26*	*0.68*	*78.05*	*78.91*	*1.19*	*77.01*	*78.65*	*4.64*	*77.92*	*78.27*	*1.14*	*73.25*	*74.32*	*2.71*
rc104	73.31	73.98	1.70	73.45	74.64	1.97	73.32	74.17	1.59	73.85	74.71	1.87	70.04	71.18	2.62
rc105	90.96	92.84	3.61	92.33	93.71	3.72	91.82	92.67	1.91	92.60	93.61	2.43	84.38	87.18	6.19
rc106	89.21	89.80	1.20	89.71	90.22	1.93	89.50	90.18	1.41	89.03	89.48	0.90	85.21	86.49	3.49
rc107	86.14	86.72	1.37	87.28	87.91	1.33	86.59	87.06	0.95	85.31	86.91	3.63	81.91	83.12	1.82
rc108	83.09	84.03	2.21	83.44	83.79	0.58	83.40	84.14	1.40	80.96	81.92	1.97	77.05	79.10	3.66
rc201	93.72	94.37	1.86	94.06	95.54	3.71	93.33	95.17	4.10	91.98	93.06	2.98	84.66	85.07	1.17
rc202	90.29	93.77	7.07	91.98	93.58	4.96	92.50	94.02	2.86	89.35	92.94	5.67	80.52	87.16	12.91
rc203	89.66	91.52	6.40	91.45	91.99	1.07	89.44	91.01	5.09	89.96	90.93	1.85	80.78	83.00	4.88
rc204	88.29	89.82	4.59	89.24	89.78	1.61	89.00	89.50	0.69	86.71	89.09	7.94	67.71	71.54	6.68
rc205	93.03	96.03	8.47	93.28	94.21	3.33	91.04	92.19	2.43	91.27	94.03	6.08	85.28	87.35	6.06
rc206	91.63	92.44	2.88	90.61	91.55	2.45	90.76	92.71	3.33	88.80	89.80	2.84	80.12	82.30	3.68
rc207	93.11	94.11	2.04	92.73	94.37	3.15	93.00	94.70	2.79	90.99	91.85	1.60	79.22	81.21	5.00
rc208	87.09	88.79	4.22	86.99	88.64	2.57	86.38	87.26	2.06	82.80	84.03	3.90	64.40	67.18	4.78

Table 4.2: Schott Spacing Index across problem groups for MTSA and MOSA

Instance	MTSA-3	MTSA-4	MTSA-5	MTSA-6	MOSA
C1	80.98	87.43	89.32	82.64	114.82
C2	267.76	244.80	255.79	265.20	429.10
R1	23.01	18.50	19.36	23.33	32.55
R2	82.19	105.95	106.25	116.18	156.51
RC1	18.51	17.68	16.51	17.65	22.35
RC2	83.86	77.14	72.92	75.08	153.81

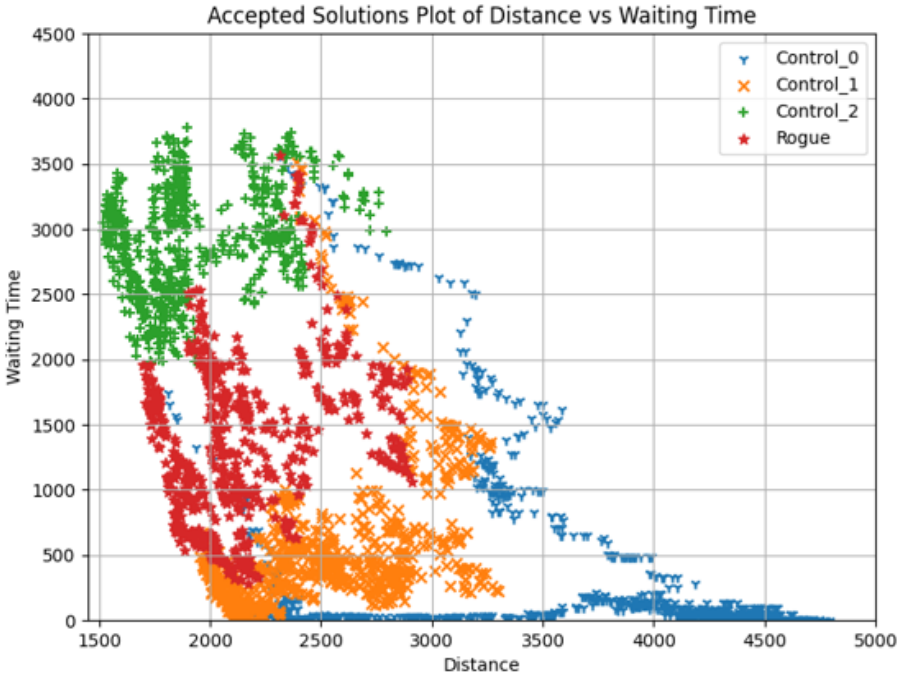

Figure 4.8: Accepted solution for MTSA-4 for solving RC201 within 45 seconds

To illustrate the efficacy of the proposed control and rogue thread mechanisms within MTSA, Figure 4.8 presents the search area coverage during the optimization process on instance RC201 over a 45-second runtime. The scatter plot of accepted solutions obtained from MTSA with four threads demonstrates how each control thread systematically explores distinct regions of the Pareto Front Approximation (PFA), thereby enhancing the breadth of the search space. Meanwhile, the rogue thread complements this process by strategically targeting the gaps between the regions explored by the control threads, effectively expanding the overall coverage of the search front.

In comparison, Figure 4.9 depicts the results for MOSA under the same computational conditions. It is evident that MOSA exhibits limitations in its exploitation capabilities within specific regions of the PFA, notably in the distance range

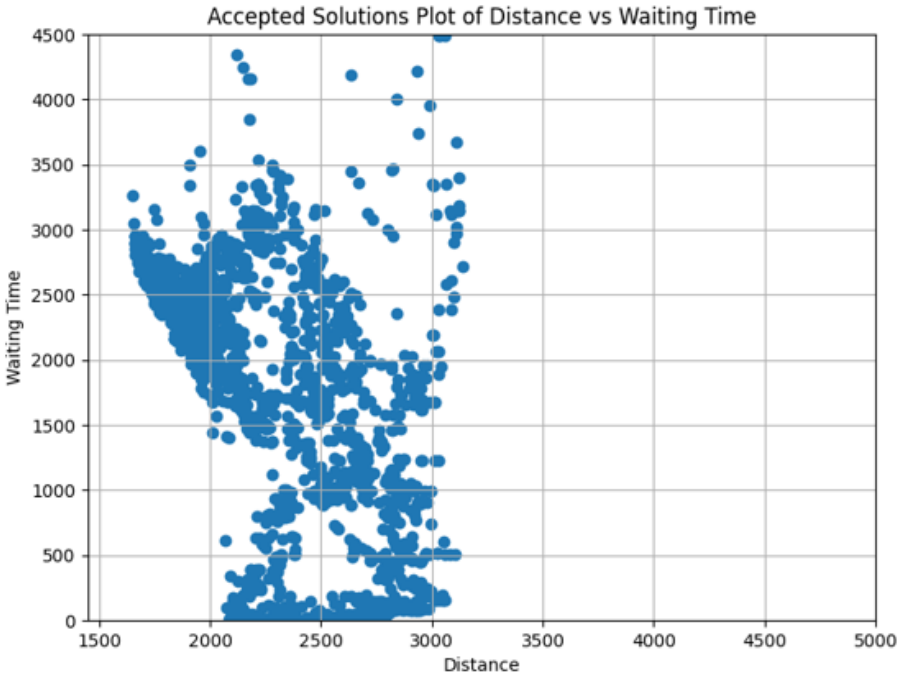

Figure 4.9: Accepted solution for MOSA for solving RC201 within 45 seconds

of 1500 to 2000, where it fails to effectively refine and explore solutions. Unlike MTSA, which leverages a combination of acceptance criteria, including weighted objectives and non-dominance conditions, to efficiently exploit targeted areas while maintaining expansion, MOSA primarily emphasizes exploration across the PFA without targeted exploitation. However, it is worth noting that given extended computation time, MOSA can improve its exploration and exploitation capabilities to generate a more comprehensive PFA.

In conclusion, the PFA achieved by MTSA-4 within the 45-second runtime clearly outperforms that of MOSA under identical computational conditions, as shown in Figure 4.10. This advantage is attributed to the structured and robust search strategies implemented within MTSA, which include weighted objective-based guidance and return-to-point policies that enable effective control over the exploration and exploitation balance during the optimization process.

4.3 Discussion and Practical Implications

The experimental results demonstrate the effectiveness of the proposed Multi-Thread Simulated Annealing (MTSA) algorithm in generating high-quality Pareto Frontier Approximations (PFAs) within practical computational timeframes. Across

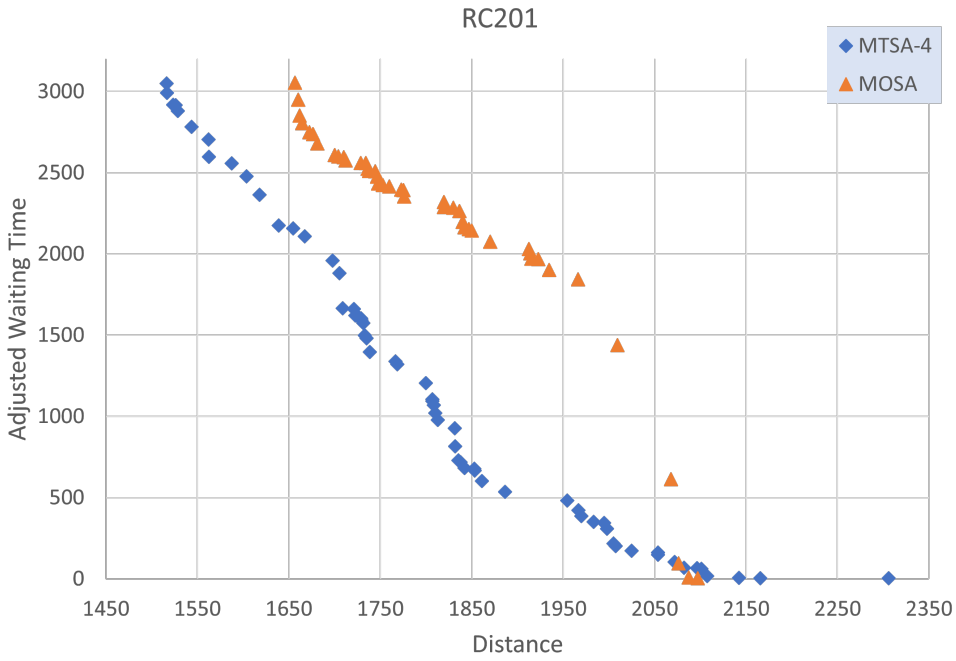

Figure 4.10: Best PFA obtained for RC201 by MTSA-4 and MOSA in 45 seconds

all benchmark evaluations, MTSA consistently outperforms the modified Multi-Objective Simulated Annealing (MOSA) algorithm in terms of both hypervolume (HV) and solution consistency.

Among the tested configurations, MTSA with three threads (MTSA-3) achieves the highest average HV values and the best maximum HV across the majority of instances, indicating a strong balance between exploration and exploitation under this setting. This performance can be attributed to MTSA's use of weighted objective functions, which enhance the algorithm's exploitation capability by focusing search efforts toward reducing specific objective values. At the same time, the use of multiple threads allows the algorithm to maintain effective exploration across the Pareto frontier, preventing premature convergence and supporting comprehensive front coverage.

In contrast, MOSA, operating with a single-thread structure, exhibits limitations in its ability to exploit and expand the Pareto front effectively within the same computational budget. Unlike population-based algorithms such as NSGA-II, which inherently leverage evolutionary operators for front progression, MOSA's single-thread structure restricts its capacity for concurrent exploration and exploitation. The introduction of multi-threading within MTSA addresses this limitation, enabling significant improvements in the algorithm's ability to discover well-distributed, high-quality solutions.

An additional advantage of the MTSA framework lies in its performance consistency. The observed range of HV values across five independent runs for each MTSA configuration is notably smaller than that of MOSA, indicating reliable performance across repeated trials. This consistency is critical in practical applications, where decision-makers require stable and predictable algorithm behavior under time-constrained environments.

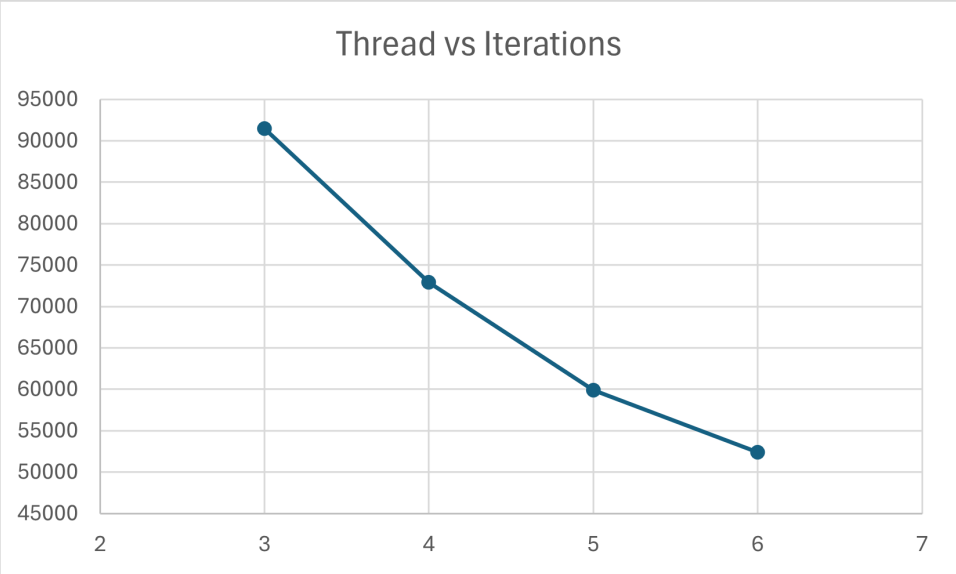

Figure 4.11: Relationship between the number of threads and the number of iterations executed within a fixed computational time.

Compared to population-based multi-objective evolutionary algorithms (MOEAs), MTSA exhibits lower structural complexity, as it does not rely on population-wide genetic operators. Its primary computational cost arises from neighborhood search operations executed independently within each thread, causing the computational effort to increase approximately linearly with the number of threads. This behavior is empirically illustrated in Fig. 4.11, where the number of iterations decreases proportionally as the number of threads increases under a fixed time budget, reflecting the independent execution of neighborhood searches.

In the computational experiments, Solomon benchmark instances are treated as large-scale problems, and the results demonstrate that MTSA can solve these instances effectively within a reasonable computation time, indicating good scalability under commonly used benchmark settings. For larger problem instances, computational complexity increases with the number of nodes, since neighborhood search operations dominate the runtime and their cost grows with problem size. Consequently, the scalability of MTSA is primarily governed by the number of threads and the instance size, with computational effort increasing proportionally in both

dimensions.

From a practical perspective, the MTSA algorithm provides a scalable and flexible tool for addressing multi-objective routing problems where both efficiency and service quality are critical. By effectively balancing exploration and exploitation through its multi-threaded design and weighted objective guidance, MTSA supports decision-makers in evaluating trade-offs between competing objectives, enabling the generation of well-distributed PFAs that align with practical logistics planning needs.

Overall, the results confirm that the proposed MTSA framework enhances the capabilities of traditional MOSA, offering a robust and efficient approach for solving complex multi-objective vehicle routing problems under realistic computational constraints.

4.4 Chapter Remarks

This chapter introduced the Multi-Thread Simulated Annealing (MTSA) algorithm as a scalable and effective method for generating high-quality Pareto Frontier Approximations (PFAs) for the Multi-Objective Vehicle Routing Problem with Time Windows and Demand Priority (MO-VRPTWDP). By integrating parallelism, adaptive weight adjustment, and soft priority objectives, MTSA successfully addresses the exploration-exploitation trade-offs commonly encountered in SA-based methods while preserving strong local search capabilities essential for high-quality solution refinement.

The experimental results demonstrated that MTSA consistently outperforms the modified MOSA baseline across multiple benchmark instances, achieving superior solution quality and maintaining stable performance across repeated runs. The ability to control the weight configurations of each thread within MTSA enables targeted exploration of specific regions of the Pareto frontier, allowing decision-makers to emphasize particular trade-offs relevant to operational needs.

This capability lays the groundwork for the next advancement in this dissertation: the development of the Reinforcement Learning Multi-Thread Simulated Annealing (RL-MTSA) framework, which extends MTSA to enable preference-aware, user-guided optimization. By leveraging the weight control mechanism established in MTSA, RL-MTSA introduces adaptive preference learning into the multi-objective optimization process, facilitating dynamic exploration of user-defined regions within the Pareto frontier. This innovative approach will be detailed in Chapter 5.

Chapter 5

Reinforcement Learning Multi-Thread Simulated Annealing (RL-MTSA)

In multi-objective vehicle routing problems, conventional optimization algorithms such as Multi-Objective Simulated Annealing (MOSA) and its parallelized variant, Multi-Thread Simulated Annealing (MTSA), aim to approximate the entire Pareto frontier through uniform search distribution. While effective in maintaining solution diversity, such methods may fall short in aligning with the practical needs of real-world decision-makers, who are often interested in only specific regions of the trade-off space. Uniform exploration usually leads to unnecessary computational effort in less relevant areas.

To overcome this limitation, this chapter proposes the **Reinforcement Learning Multi-Thread Simulated Annealing (RL-MTSA)** framework, a hybrid algorithm that integrates reinforcement learning (RL) into the MTSA structure to enable preference-aware, region-targeted search. The key innovation of RL-MTSA lies in embedding a learning agent that continuously observes the progress of the search and dynamically adjusts the weight configurations of individual SA threads. This adaptive control steers the multi-thread search process toward user-specified regions of the Pareto frontier, effectively concentrating computational resources where they are most valuable.

By aligning search behavior with stakeholder preferences, RL-MTSA not only improves the relevance of generated solutions but also enhances convergence speed and decision support quality. The framework is particularly suited for complex, large-scale vehicle routing scenarios where decision-makers require focused exploration of specific cost-service trade-offs. Through its combination of adaptive learn-

ing and structured parallel search, RL-MTSA represents a significant advancement in preference-aware multi-objective optimization.

The remainder of this chapter is organized as follows. Section 5.1 presents the detailed methodology of the proposed RL-MTSA framework, including its integration of reinforcement learning with multi-thread simulated annealing. Section 5.2 describes the experimental setup used to evaluate the algorithm's effectiveness. Section 5.3 discusses the results, with particular attention to performance, adaptability, and practical implications. Finally, Section 5.4 summarizes the key findings and concludes the chapter.

5.1 RL-MTSA Methodology

5.1.1 Framework Overview

The Reinforcement Learning Multi-Thread Simulated Annealing (RL-MTSA) framework extends the capabilities of the original MTSA algorithm by introducing an adaptive, learning-based mechanism for preference-aware search. While MTSA distributes multiple Simulated Annealing (SA) threads across the objective space using fixed weight configurations, RL-MTSA introduces reinforcement learning (RL) to dynamically steer the search toward user-specified regions of interest on the Pareto frontier.

At the core of the RL-MTSA framework is a reinforcement learning agent that operates in tandem with multiple SA threads. This agent observes the performance of each thread in approximating the desired target region and adjusts their objective weight configurations accordingly. Specifically, the user specifies a **target point** in the objective space, typically representing a preferred trade-off between conflicting objectives, such as total distance and priority-weighted waiting time. The RL agent then learns to allocate weights that drive each thread's search trajectory toward this region, enhancing solution relevance and decision support quality.

The framework maintains a total of four threads: three are RL-Driven Threads whose weights are updated throughout the optimization process based on feedback from the learning agent, and one is a Fixed Thread with a fixed weight configuration. The fixed thread serves two purposes: (1) to maintain coverage of extreme regions on the Pareto frontier, and (2) to preserve solution diversity across the entire front, ensuring that the algorithm does not overly narrow its focus around the target point.

This combination of adaptive learning and structured parallel search enables

RL-MTSA to strike a balance between exploration and preference-driven exploitation. By focusing computational effort on the most relevant portions of the Pareto frontier, the framework improves convergence speed and enhances the practical applicability of multi-objective optimization in complex vehicle routing problems.

5.1.2 User-Specified Target Point

A key feature of the RL-MTSA framework is its ability to focus the optimization process on a user-defined region of interest along the Pareto frontier. This mechanism addresses the common practical need in multi-objective decision-making where stakeholders are not interested in the entire Pareto set but instead prioritize solutions that lie within specific trade-off zones between competing objectives.

In this study, the reference for guiding the search is based on the Pareto *knee point*—a region representing the most balanced trade-off between objectives. Users can specify their preferred region relative to this knee point by providing a scalar input value within the range $[-1, 1]$. A value of zero corresponds exactly to the knee point. Positive values (e.g., 0.1) indicate a preference for solutions positioned below the knee point, prioritizing shorter waiting times even if travel distances increase. Conversely, negative values (e.g., -0.3) shift the focus toward solutions above the knee point, favoring reduced travel distance at the expense of higher waiting times.

The identification of the user-specified *Target Point* proceeds by counting the number of solutions above or below the knee point along the non-dominated front. Suppose there are n solutions below the knee point, ordered from nearest to farthest relative to it. A user-defined value of 0.1 would guide the search to the first solution below the knee point if there are ten solution below the knee point. A value of 0.5 would correspond to the midpoint between the knee point and the most extreme solution in that direction—i.e., the solution at index $\lfloor 0.5 \times n \rfloor$. Similarly, a value of -0.4 would select the fourth solution above the knee point if ten such solutions exist, as illustrated in Figure 5.1.

This user-guided selection mechanism provides a flexible and interpretable means for narrowing the focus of the search algorithm. Rather than uniformly distributing effort across the entire Pareto frontier, RL-MTSA concentrates search resources around the user-indicated region. This is accomplished through reinforcement learning, where the agent continuously updates thread weight configurations to enhance exploration and exploitation in the vicinity of the Target Point.

By enabling users to express preferences in a simple scalar form, RL-MTSA bridges the gap between high-dimensional Pareto approximation and practical decision-making. The framework ensures that computational effort is aligned with stake-

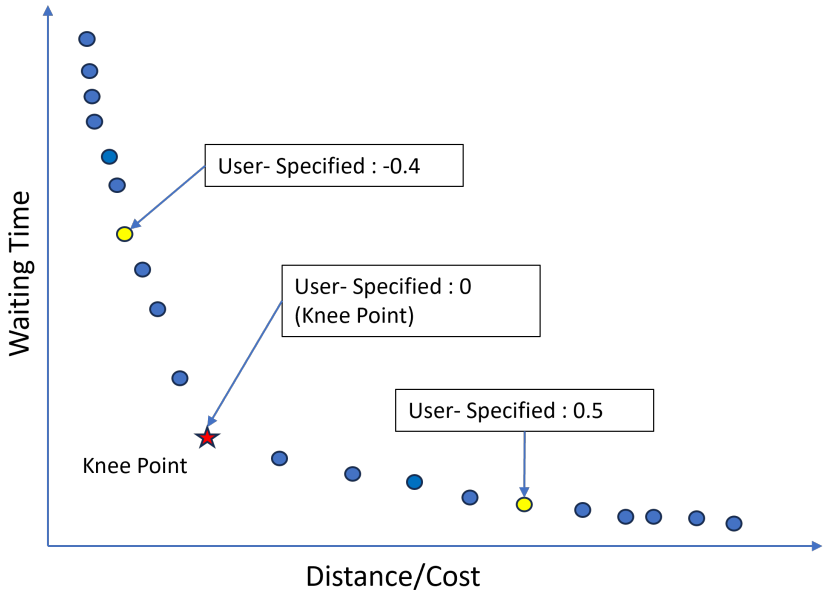

Figure 5.1: Illustration of User-Specified Target Point Selection

holder priorities, improving both the efficiency of the optimization process and the quality of the solutions generated in terms of decision relevance.

5.1.3 RL-MTSA Structure

The architecture of RL-MTSA is depicted in Figure 5.2. The algorithm begins by generating an initial solution set and assigning predefined weights to each thread. It then performs neighborhood search operations throughout a designated warm-up phase. Once this phase concludes, the RL agent begins to actively adjust the weights of the RL-driven threads based on their observed search behavior and performance. The algorithm proceeds iteratively, conducting further neighborhood searches, and evaluating performance at the end of each batch. Upon batch completion, the reward is calculated and the RL agent updates the corresponding weight configurations. This adaptive loop continues until the algorithm satisfies a predefined stopping criterion.

Initialization of Solutions and Weights

To initialize the MO-VRPTWDP, the algorithm applies the Time Window Insertion Heuristic (TWIH), a commonly used strategy for generating feasible initial solutions in VRPTW settings. For weight initialization, the following values are assigned to the threads: $[0.9, 0.7, 0.5, 1.0]$. The first three weights correspond to

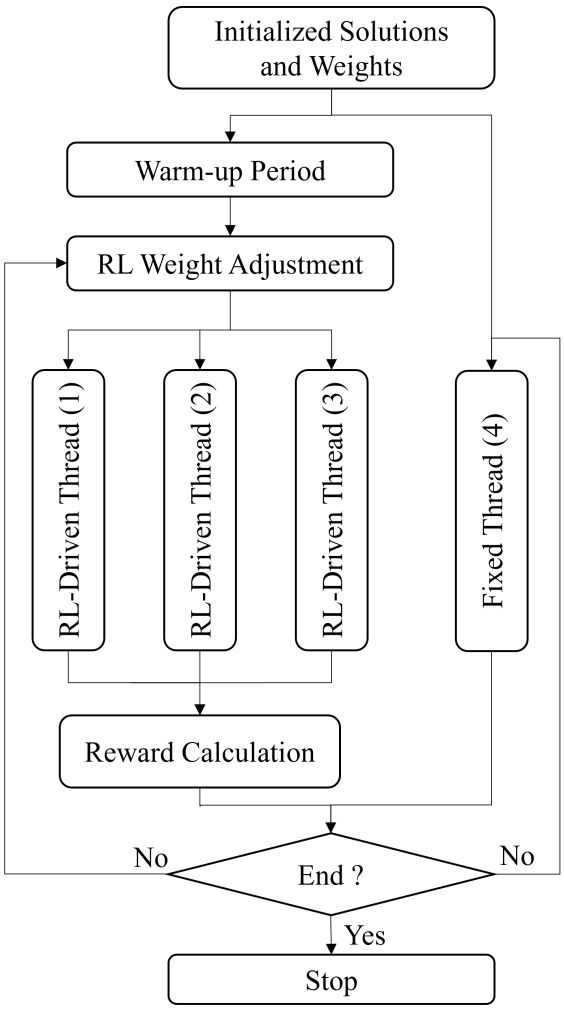

Figure 5.2: Overview of the RL-MTSA framework

RL-driven threads whose weights will be adjusted dynamically during search, while the fourth weight is assigned to the Fixed thread, which is responsible for exploring the boundary regions of the Pareto frontier and promoting diversity.

Warm-Up Phase and Weight Adjustment

Following initialization, the algorithm executes a warm-up phase, during which each thread independently performs neighborhood search operations. This phase allows the RL agent to gather informative feedback about the search trajectory and solution quality under the initial weight settings. Using this feedback, the RL agent then refines the weights of the RL-driven threads, guiding the search effort toward the user-defined region of interest, thereby aligning exploration with the intended trade-off preferences.

RL-Driven and Fixed Threads

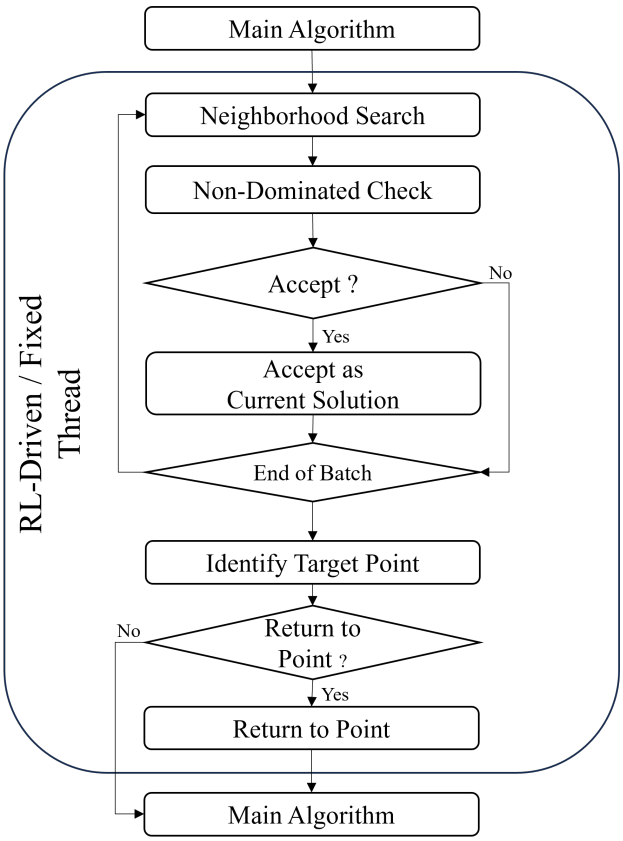

Figure 5.3: Structure of RL-Driven and Fixed Threads

The core structure of RL-driven and Fixed threads is consistent, differing only in their weight update policies. While Fixed threads maintain constant weights, RL-driven threads adapt their weights dynamically under the guidance of the RL agent. The architecture of each thread is shown in Figure 5.3.

Each thread contains three critical components: a neighborhood search module, a non-dominated solution checking module, and an acceptance condition module.

- **Neighborhood Search Module:** This module generates new candidate solutions by applying swap and insertion heuristics to the current solution. These operations facilitate local improvement and exploration of the solution space.

- **Non-Dominated Checking Module:** After generating a new solution, this module evaluates whether it dominates any existing solutions in the Pareto archive. Unlike population-based methods such as NSGA-II, which maintain a fixed-size archive, RL-MTSA keeps an unlimited archive of non-dominated solutions, enabling more comprehensive and diverse frontier coverage.

- **Acceptance Condition Module:** A new solution is accepted as the current solution if it satisfies at least one of the following:

  1. It dominates at least one existing solution on the Pareto frontier.
  2. It has a lower weighted sum objective value than the current solution.
  3. It is accepted probabilistically, with the acceptance probability α defined by its distance to the nearest non-dominated solution:

$$\alpha = \begin{cases} 0, & \text{if } D > 100 \\ \frac{100-D}{2000}, & \text{otherwise} \end{cases} \tag{5.1}$$

     where D is the Euclidean distance between the new solution and the nearest solution on the current Pareto frontier.

Each thread iterates this process until the current batch concludes. Upon batch completion, the Target Point is determined to guide further search. To identify this point, the Pareto frontier is smoothed using a moving average filter with a window size of $k = 5$, after which the KneeLocator Python package is used to detect the knee point.

Target Point Identification and Return-to-Point Strategy

Once the knee point is identified, the algorithm locates the user-specified Target Point. If the current solution deviates significantly from this point, the Return-to-Point mechanism is activated. This mechanism reassigns the current solution to the Target Point, reinforcing search activity in the preferred trade-off region. By periodically identifying and returning to the Target Point, RL-MTSA sustains focused, user-aligned exploration of the Pareto frontier while maintaining adaptability.

5.1.4 Reinforcement Learning Framework

The integration of reinforcement learning (RL) into the RL-MTSA framework enables the optimization algorithm to focus the search process dynamically based on decision-maker preferences. Unlike traditional multi-objective metaheuristics, which aim to approximate the entire Pareto frontier uniformly, RL-MTSA introduces an adaptive layer of intelligence that selectively intensifies the search in regions of interest. This is particularly valuable in practical applications where stakeholders are often concerned with only a subset of trade-offs rather than the entire frontier.

The RL framework in RL-MTSA is responsible for learning how to adjust the weights of the RL-driven threads during the optimization process. These weights influence the aggregation of multiple objectives and determine the direction of each thread's search. By continuously observing the progress of the optimization and receiving rewards based on performance relative to a user-defined target point on the Pareto frontier, the RL agent learns which weight adjustments yield the most desirable outcomes.

This framework operates in cycles, with each cycle consisting of thousands of neighborhood search iterations conducted independently by each thread. At the end of each cycle, the RL agent evaluates the search performance, updates its policy, and selects new weight adjustment actions for the next batch of iterations. The feedback-driven loop of state observation, action selection, and reward evaluation allows the RL agent to progressively refine its strategy for guiding the optimization process toward solutions that are not only efficient but also preference-aligned.

The following subsection introduces the RL environment in detail, including its architecture and interaction flow. It elaborates on three fundamental components of the environment—Action Space, Observation Space, and Reward Function—and explains how they operate together to facilitate targeted and adaptive multi-objective optimization. The training strategy is also discussed in this section.

RL Environment

The reinforcement learning (RL) environment embedded within the RL-MTSA framework is designed to dynamically steer the search process by adjusting thread weights based on user-specified preferences. Its primary function is to ensure that optimization remains aligned with targeted trade-offs along the Pareto frontier while continuously adapting to feedback from the search dynamics.

The RL process begins with a warm-up phase, during which each thread performs neighborhood searches to improve its initial solution. This phase provides a baseline for solution quality and generates valuable information for initializing the learning process. After the warm-up period concludes, the RL agent takes an active role in directing the search. Each learning cycle spans 3000 iterations, during which neighborhood search operations are carried out across all RL-Driven threads. At the end of each cycle, the RL agent evaluates the performance of the search and updates the weight configurations of the threads to better align the search direction with the user-specified target region.

The RL environment is composed of three core components: the *Action Space*, *Observation Space*, and *Reward Function*. These components interact in a closed

feedback loop to continuously guide the learning and optimization process.

- **Action Space:** This defines the set of discrete actions available to the RL agent, primarily focused on adjusting the weight configurations of the RL-driven threads. Through these actions, the agent modulates the emphasis placed on different objectives, thereby altering the direction of search across the trade-off landscape.

- **Observation Space:** The observation vector provides the RL agent with real-time feedback on the current state of the optimization process. It includes metrics such as thread weight distributions, progress toward the user-specified target point, and historical adjustment patterns. This high-dimensional input enables the agent to evaluate the context of previous decisions and plan subsequent actions accordingly.

- **Reward Function:** The reward mechanism assesses the effectiveness of the selected actions based on how well the generated solutions match the preferred trade-off region. Positive rewards are granted for solutions that move closer to the target point and contribute to improved hypervolume and diversity. Conversely, penalties are applied when the search drifts away from the specified region or fails to maintain diversity. This reward shaping promotes a balanced trade-off between convergence and exploration.

These components operate in an iterative feedback loop. The Observation Space captures the evolving search state, which informs the RL agent's decision-making. Based on the observed state, the agent selects an action from the Action Space to adjust thread weights. After search updates are performed, the Reward Function evaluates the outcome and feeds back performance metrics, thereby reinforcing successful strategies and discouraging ineffective ones.

Through this interactive learning process, the RL environment enables RL-MTSA to progressively refine its optimization strategy. This results in improved convergence toward high-quality solutions concentrated around the user-specified target, enhancing the algorithm's utility in multi-objective decision-making contexts.

Action Space

The action space in the RL-MTSA framework defines how the reinforcement learning (RL) agent adapts the behavior of each RL-Driven Thread by modifying their weight configurations. Since MTSA operates with multiple threads exploring the

Pareto frontier simultaneously, the RL agent must determine how much to alter the weights of these threads during each decision cycle. These adaptive weight changes are critical for steering the search process toward user-defined regions of interest while maintaining a diverse and well-distributed solution set.

In this study, the action space is formulated as a *MultiDiscrete* space, enabling the RL agent to select a discrete weight adjustment action for each RL-Driven Thread independently. The framework comprises three RL-Driven Threads, each associated with a set of nine possible discrete actions. These actions define how the corresponding thread's current weight is adjusted, allowing for both fine-grained and more substantial changes.

The set of discrete adjustments available to the agent is given by:

$$\text{Adjustments} = \{0.2,\ 0.1,\ 0.05,\ 0.025,\ 0,\ -0.025,\ -0.05,\ -0.1,\ -0.2\}$$

Each value in the set represents an incremental change to the thread's weight for the distance-minimizing objective. Positive values shift the thread's focus more strongly toward minimizing total distance, while negative values reduce the emphasis on distance, thereby implicitly increasing attention on the waiting time objective. The inclusion of a zero-change action permits the thread to maintain its current weight when no modification is deemed beneficial.

Given the three independent RL-Driven Threads, each capable of selecting from 9 adjustment options, the total number of possible action combinations per decision step is $9^3 = 729$. This large and structured action space empowers the RL agent to flexibly control the multi-threaded search process.

The availability of multiple adjustment magnitudes plays a vital role in achieving a balance between exploration and exploitation. Larger step sizes promote exploration by broadening the search coverage, helping to discover new promising regions of the Pareto frontier. Conversely, smaller step sizes support exploitation by enabling the refinement of promising areas already near the target region. This capability allows the RL-MTSA framework to respond dynamically to ongoing search conditions and user preferences, ultimately contributing to the generation of high-quality, preference-aligned solution sets.

Observation Space

The observation space in RL-MTSA provides the reinforcement learning agent with essential information about the current and past states of the optimization process. This data-driven awareness enables the agent to adaptively adjust the search di-

rection by modifying the objective weight configurations of each RL-driven thread. In RL-MTSA, the optimization problem is modeled as a bi-objective minimization, where the total objective is expressed as a weighted sum:

$$\text{Objective} = w_{TD} \cdot TD + w_{WT} \cdot WT$$

Since the weights are constrained such that $w_{TD}+w_{WT} = 1$, only the distance weight w_{TD} needs to be explicitly adjusted. The waiting time weight w_{WT} is automatically derived from this, i.e., $w_{WT} = 1 - w_{TD}$.

The observation space is defined as a 19-dimensional vector implemented using a continuous `Box` space bounded in the range $[-100, 100]$. This setup provides sufficient flexibility to encode a wide spectrum of search dynamics and optimization histories.

The 19 observation features are structured as follows:

- **Current Thread Weights (3 inputs):** These represent the current values of w_{TD} for each RL-driven thread, providing insight into the present search orientation with respect to minimizing travel distance versus waiting time.
- **Previous Weight Sets (6 inputs):** These consist of the w_{TD} values used in the two most recent reinforcement learning cycles (3 threads $\times$ 2 past steps). Including this history allows the agent to identify trends in weight adjustments and their impact on search performance.
- **Current Normalized Distance Values (3 inputs):** These values indicate the relative deviation of each thread's current solution from the Target Point in terms of travel distance, computed as:

  $$\text{Normalized Distance} = \frac{D_{\text{Cur}} - D_{\text{Target}}}{D_{\text{Target}}} \tag{5.2}$$

  where D_{Cur} is the total distance of the thread's current solution and D_{Target} is the distance value of the user-defined Target Point. A value near zero indicates successful focus on the desired trade-off region, while positive or negative values suggest deviation above or below the target.
- **Previous Normalized Distance Values (6 inputs):** These correspond to the normalized deviations from the Target Point observed in the two preceding reinforcement learning cycles. They enable the agent to assess longer-term patterns in convergence behavior.

- **Exploration Weight of Target Point (1 input):** This input stores the w_{TD} value used by the thread that discovered the current Target Point. This contextual reference allows the agent to associate effective weight configurations with search success.

The observation space is designed to include the current state and two historical data points, forming three consecutive observations. By capturing both immediate feedback and short-term historical behavior, this structure allows the agent to infer the direction and trend of the search relative to the user-specified target point, enabling informed and strategic weight updates. Two historical observations are sufficient to determine whether the search is moving toward or away from the target region while keeping the observation space compact.

Although incorporating additional history is possible, doing so would increase the dimensionality of the observation space and the complexity of the policy network, potentially introducing noise and reducing training stability. Moreover, since the observation space directly defines the network input, altering the number of historical observations would require retraining the policy with a different architecture, making fair ablation difficult. Therefore, the use of two historical data points represents a balanced and stable design choice that improves convergence and enhances the practical relevance of the generated solutions.

Reward Function

The reward function in RL-MTSA is designed to guide the reinforcement learning (RL) agent toward producing solutions that closely align with the user-specified Target Point, which represents a preferred trade-off between total travel distance and customer waiting time. This function evaluates performance in each batch of iterations and influences the learning process by assigning rewards or penalties based on proximity to the Target Point, improvements in hypervolume, and solution diversity.

The reward function consists of three main components:

1. **Proximity-Based Reward (R_P):** This component measures how closely each RL-Driven thread's solution approaches the Target Point. For each RL-Driven Thread i, the proximity-based reward $R_P = \sum_{i=1}^{3} R_i$ is computed as follows:

    - *High Proximity Reward:* A solution receives a high reward when:

$$\left| \frac{D_{\text{Cur},i} - D_{\text{Target}}}{D_{\text{Target}}} \right| < 0.5$$

In such cases, the reward is:

$$R_i = R_i + 0.5$$

- *Moderate Proximity Reward:* A moderate reward is granted if:

$$\left| \frac{D_{\text{Cur},i} - D_{\text{Target}}}{D_{\text{Target}}} \right| < 0.75$$

The associated reward is:

$$R_i = R_i + 0.25$$

- *Penalty for Deviation:* If a solution strays significantly from the Target Point, a penalty is applied:

$$R_i = R_i - \left(0.1 + 0.5 \cdot \left| \frac{D_{\text{Cur},i} - D_{\text{Target}}}{D_{\text{Target}}} \right| \right)$$

2. **Hypervolume Improvement Reward (R_H):** To encourage convergence and global quality, the improvement in hypervolume over the course of a batch is rewarded. This is computed as:

$$R_H = 0.01 \cdot (HV_{\text{cur}} - HV_{\text{init}})$$

where HV_{cur} and HV_{init} represent the hypervolume at the end and start of the batch, respectively.

3. **Diversity Reward (R_S):** To ensure that the search does not converge prematurely and maintains spread along the Pareto front, a diversity reward is introduced. It is triggered when the normalized distance spread across current solutions exceeds a defined threshold:

$$R_S = \begin{cases} 0.25 & \text{if } \max\left(\frac{D_{\text{Cur},i} - D_{\text{Target}}}{D_{\text{Target}}}\right) - \min\left(\frac{D_{\text{Cur},i} - D_{\text{Target}}}{D_{\text{Target}}}\right) > 0.15 \\ 0 & \text{otherwise} \end{cases}$$

The total reward at the end of each batch is calculated by summing the proximity reward, hypervolume reward, and diversity reward:

$$R_{\text{Final}} = R_P + R_H + R_S$$

This structured reward formulation allows RL-MTSA to adaptively steer the optimization process toward solutions that are both high quality and well-aligned with user-defined trade-offs. The balance between proximity, convergence, and diversity in the reward function was fine-tuned through empirical analysis, and its impact is further evaluated through ablation studies presented in the experimental section.

Training Strategy

The reinforcement learning (RL) agent in RL-MTSA is trained using two state-of-the-art deep RL algorithms: Proximal Policy Optimization (PPO) (Schulman et al., 2017) and Advantage Actor-Critic (A2C) (Mnih et al., 2016). To ensure broad generalization and avoid overfitting, the training process uses only the odd-numbered Solomon benchmark instances, comprising a total of 29 problem cases. During training, user-specified points are sampled uniformly from the range $[-0.5, 0.5]$ in increments of 0.1, allowing the agent to learn effective search behavior across a range of trade-off preferences.

PPO is a robust policy gradient method that stabilizes learning by employing a clipped surrogate objective function, preventing large, destabilizing updates. This characteristic makes it particularly well-suited for complex multi-objective optimization problems like MO-VRPTWDP. In RL-MTSA, PPO is implemented with a multi-layer perceptron (MLP) policy network using ReLU activation functions. The actor network comprises four hidden layers with 64 neurons each, while the critic network includes three hidden layers with 128 neurons each. A learning rate of 0.00125 and an entropy coefficient of 0.0175 are used to encourage sufficient exploration. The PPO model is trained for 120,000 timesteps to allow the agent to learn reliable search strategies.

A2C, another policy-based approach, adopts a synchronous actor-critic structure, where the actor updates the policy and the critic estimates state values to support efficient learning. Although A2C lacks PPO's clipping mechanism—which may result in higher variance during policy updates—it offers strong learning performance in multi-objective contexts. The A2C configuration in RL-MTSA follows the same network structure as PPO, with four hidden layers of 64 neurons in the actor and three hidden layers of 128 neurons in the critic. The learning rate is set to 0.00075, and the entropy coefficient is again set at 0.0175. The A2C model is also trained for 120,000 timesteps to match the PPO training horizon.

The configuration details for both algorithms are summarized in Table 5.1. These settings ensure the RL agent is capable of learning adaptive strategies for

weight adjustment that effectively direct the search toward the user-specified regions of the Pareto frontier.

Table 5.1: Training configurations for PPO and A2C in RL-MTSA

Parameter	**PPO Value**	**A2C Value**
Policy Network Architecture	[64, 64, 64, 64]	[64, 64, 64, 64]
Value Network Architecture	[128, 128, 128]	[128, 128, 128]
Activation Function	ReLU	ReLU
Learning Rate	0.00125	0.00075
Entropy Coefficient	0.0175	0.0175
Training Timesteps	120,000	120,000
Input Shape	(19,)	(19,)
Output Shape	(3,)	(3,)

The input shape corresponds to the 19-dimensional observation space described in Section 5.1.4, capturing the state of each RL-driven thread, historical weights, and proximity to the Target Point. The output shape reflects the three discrete actions used to adjust thread weights. A comparative analysis of the performance of PPO and A2C is presented in the following experimental results section.

5.2 Computational Experiments

This section presents the evaluation of the RL-MTSA framework against the baseline Multi-Thread Simulated Annealing (MTSA) through **dominance-based comparison** and **localized hypervolume analysis**. The objective is to determine whether reinforcement learning enhances the search process toward higher-quality trade-offs that are difficult to achieve with traditional simulated annealing alone.

The computational experiments are conducted on the well-known Solomon benchmark set (Solomon, 1987), which contains 56 VRPTW instances consisting of 100 customers and one depot. Each customer is associated with a service time, demand, and time window. These instances are categorized into six groups: C1 and C2 (clustered customers), R1 and R2 (random distribution), and RC1 and RC2 (a mix of clustered and random). To ensure comparability, all experiments are performed on a 12th Gen Intel(R) Core(TM) i7-12650H 2.30 GHz processor under a fixed runtime of 120 seconds.

In addition, an ablation study on the reward function is conducted to verify that the reward design effectively guides the search toward the user-specified region of the Pareto frontier.

5.2.1 Dominance-Based Comparison

To evaluate the contribution of reinforcement learning to solution quality, RL-MTSA is directly compared against the standard MTSA algorithm using a dominance-based analysis. Both algorithms are executed under identical computational conditions, with a fixed runtime of 120 seconds per run, ensuring a fair comparison of their capabilities.

This evaluation is conducted across all 56 Solomon benchmark instances, which are widely used in the literature for assessing VRPTW algorithms. To examine RL-MTSA's ability to target specific trade-off regions, each instance is tested with five different user-specified target points: 0.5, 0.25, 0, -0.25, and -0.5. These values correspond to positions along the Pareto frontier relative to the knee point and represent different preferences between total distance and waiting time. As a result, the experiment comprises 280 runs in total (56 instances $\times$ 5 preference levels), providing a thorough assessment of the framework's preference-aware performance.

The core of this analysis lies in determining whether the solutions generated by RL-MTSA dominate, are dominated by, or remain neutral to the Pareto frontier produced by MTSA. If a large proportion of RL-MTSA solutions dominate those of MTSA, this would demonstrate the effectiveness of reinforcement learning in producing high-quality, preference-aligned solutions that go beyond what can be achieved with traditional metaheuristics.

The results of this dominance comparison are presented in Table 5.2. For the PPO-based RL-MTSA, the algorithm dominates MTSA in 225 out of 280 runs (80.36%), while being dominated in only 43 runs (15.36%). The remaining 12 runs (4.29%) result in mutually non-dominated solutions. These findings highlight PPO's strength in guiding the search toward superior trade-offs concentrated around user-specified points on the Pareto frontier.

The A2C-based RL-MTSA also demonstrates strong performance, with 200 runs (71.43%) yielding solutions that dominate those of MTSA. However, it is outperformed by MTSA in 61 cases (21.79%) and yields neutral results in 19 cases (6.79%). While A2C remains effective, the results suggest greater variability in its learning behavior and search efficiency, likely due to the lack of stabilization mechanisms such as those used in PPO.

These findings underscore the advantage of PPO in preference-aware multi-objective optimization, attributed to its clipped policy updates and stable convergence properties. In contrast, the higher variance observed in A2C outcomes may stem from its synchronous update strategy, which can lead to less consistent performance.

Table 5.2: Dominance Comparison between RL-MTSA and MTSA

Dominance Category	**Count (PPO)**	**Percent (PPO)**	**Count (A2C)**	**Percent (A2C)**
MTSA Dominates	43	15.36%	61	21.79%
Neutral	12	4.29%	19	6.79%
RL-MTSA Dominates	225	80.36%	200	71.43%

In summary, RL-MTSA consistently produces solutions that dominate those of MTSA, validating the effectiveness of incorporating reinforcement learning into simulated annealing for targeted optimization. PPO demonstrates higher robustness and performance stability compared to A2C, making it a more reliable choice for guiding search behavior in preference-sensitive multi-objective vehicle routing scenarios.

5.2.2 Localized Hypervolume Analysis

The Hypervolume (HV) metric is a widely adopted performance indicator in multi-objective optimization, measuring the volume of objective space dominated by a set of non-dominated solutions with respect to a defined reference point. Higher HV values imply broader coverage and better trade-off quality. In conventional evaluations, HV is computed over the entire Pareto frontier. However, such global assessments may not accurately reflect the ability of an algorithm to concentrate search efforts on user-specified preference regions.

To better capture the preference-awareness capability of RL-MTSA, we employ a localized HV computation method. This approach focuses exclusively on the region surrounding the Target Point, enabling a more precise evaluation of search effectiveness in the specified trade-off region.

The localized HV region is defined by a threshold around the distance component of the Target Point. Let $(D_{\mathrm{RL}}, W_{\mathrm{RL}})$ denote the total travel distance and total waiting time of the solution suggested by RL-MTSA. The bounds for localized HV inclusion are defined as:

$$\text{Lower Threshold} = D_{\mathrm{RL}} \times (1 - \delta) \tag{5.3}$$

$$\text{Upper Threshold} = D_{\mathrm{RL}} \times (1 + \delta) \tag{5.4}$$

where $\delta = 0.05$ is the threshold level used in this study. This value corresponds to the reward proximity used in the RL agent's training, making it a suitable radius for evaluating search precision.

Only solutions with travel distances within the defined range are included in

the localized HV calculation. All others are excluded, ensuring that the metric reflects performance strictly in the vicinity of the user-defined region. Figure 5.4 illustrates this concept, where blue triangles denote RL-MTSA solutions, orange diamonds represent MTSA results, and red squares mark the RL-suggested point. The vertical dashed lines delineate the inclusion thresholds for localized HV.

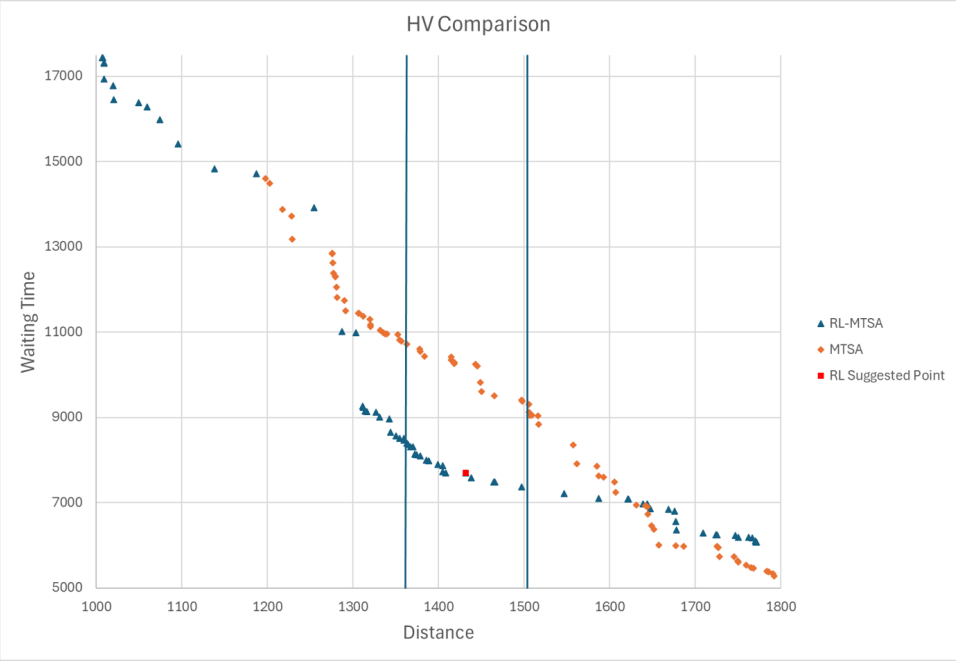

Figure 5.4: Illustration of Localized HV Region for Comparison

To evaluate RL-MTSA's ability to focus on preference regions, five user-specified target values were selected: 0.5, 0.25, 0, -0.25, and -0.5, reflecting varying trade-off preferences between waiting time and travel distance. Each configuration was applied to all 56 Solomon benchmark instances. The performance of RL-MTSA (under both PPO and A2C agents) was compared against baseline MTSA using the localized HV metric.

Table 5.3 presents the results, demonstrating that PPO-based RL-MTSA consistently outperforms MTSA across all target levels and instance categories. These findings confirm PPO's strong ability to focus the search within user-preferred regions and maintain solution quality.

While A2C-based RL-MTSA also shows competitive performance, often outperforming MTSA, it exhibits slightly less consistent results than PPO. For instance, under the target level 0.5, improvements over MTSA are marginal, and in some cases, MTSA performs comparably or slightly better. This variability suggests that PPO offers greater learning stability and more reliable preference alignment than A2C.

Table 5.3: Localized hypervolume comparison results

Method	Instance	Target (0.5)		Target (0.25)		Target (0)		Target (-0.25)		Target (-0.5)	
		RL-MTSA	MTSA	RL-MTSA	MTSA	RL-MTSA	MTSA	RL-MTSA	MTSA	RL-MTSA	MTSA
PPO	C1	**79.57**	78.34	**79.14**	78.55	**79.24**	77.43	**79.28**	74.92	**75.52**	71.54
	C2	**78.59**	78.27	**83.08**	79.35	**80.88**	79.49	**85.02**	84.10	**84.93**	77.83
	R1	**76.76**	75.60	**77.37**	75.77	**77.54**	75.55	**77.68**	74.89	**77.13**	71.98
	R2	**77.02**	76.19	**81.71**	80.42	**82.55**	81.21	**83.41**	79.99	**82.46**	77.64
	RC1	**75.97**	74.28	**76.64**	75.14	**76.95**	75.09	**76.12**	74.51	**75.38**	73.10
	RC2	**77.40**	76.04	**82.34**	80.75	**83.33**	81.53	**83.90**	81.88	**81.68**	80.89
	Total	**77.55**	76.46	**80.05**	78.33	**80.08**	78.38	**80.90**	78.38	**79.52**	75.50
A2C	C1	**78.53**	77.60	**79.41**	78.03	**79.33**	78.41	**77.93**	75.56	**74.63**	69.35
	C2	75.80	**76.04**	**81.74**	81.60	**83.64**	83.48	**83.92**	82.55	**80.97**	78.04
	R1	**75.85**	75.32	**77.07**	75.94	**77.22**	75.60	**77.27**	73.20	**76.54**	71.50
	R2	**77.42**	77.26	**81.00**	79.80	**82.92**	81.09	**82.83**	81.39	**81.81**	79.30
	RC1	73.64	**74.03**	**75.68**	74.95	**76.21**	74.94	**74.73**	72.67	**73.52**	72.34
	RC2	75.52	75.29	**83.01**	82.31	**83.81**	82.56	82.25	**82.37**	**81.89**	77.22
	Total	**76.13**	75.92	**79.65**	78.77	**80.52**	79.34	**79.82**	77.96	**78.23**	74.63

In conclusion, the localized hypervolume analysis demonstrates that RL-MTSA effectively concentrates the search around user-defined regions of interest, resulting in superior solution quality compared to traditional approaches. Among the tested variants, RL-MTSA trained with PPO consistently outperforms both MTSA across all target preferences, achieving better convergence and producing solutions that are more aligned with decision-maker priorities. These findings underscore the value of integrating reinforcement learning into preference-aware multi-objective optimization frameworks for vehicle routing problems.

5.2.3 Reward Function Analysis

To assess the role of each component in the reward function, an ablation study was conducted using six Solomon benchmark instances that were not included during training. Each instance was evaluated under five different user-specified preference levels, resulting in a total of 30 independent runs. The performance of the PPO-trained RL agent was compared against a baseline random policy. Figure 5.5 presents the average reward obtained by each component across all target levels.

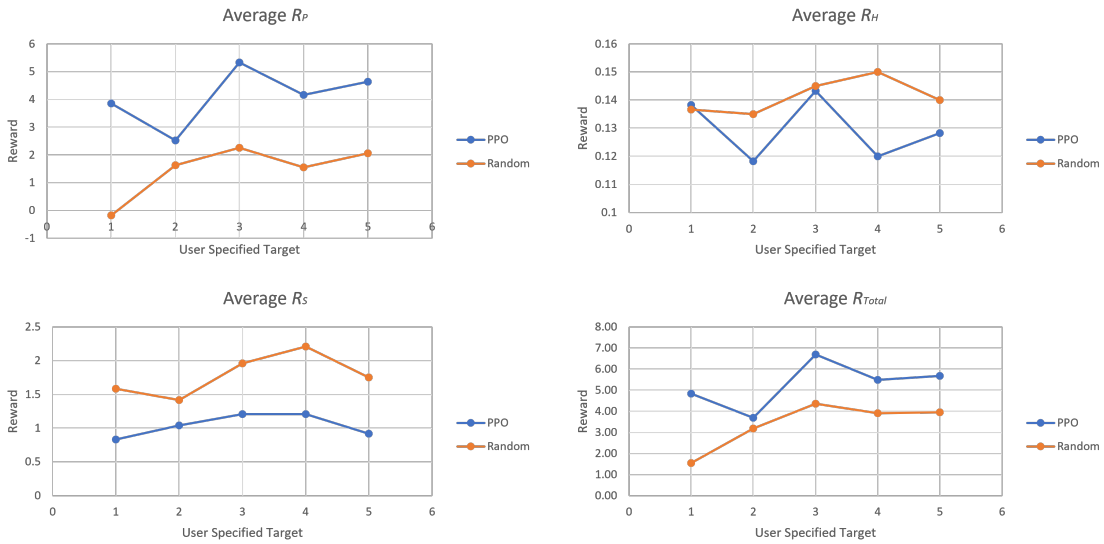

Figure 5.5: Average reward comparison between PPO and random policy

The results show that the proximity-based reward component (R_P) strongly favors the PPO policy over the random policy. This confirms that PPO effectively learns to adjust thread weights in a way that consistently guides the search toward the user-specified region of interest. As this reward directly reflects the algorithm's ability to align search outcomes with preference constraints, it plays a central role in RL-MTSA's functionality. Notably, R_P contributes approximately 77.79% of

the total average reward under the PPO policy, highlighting its critical role in the learning process.

Conversely, the hypervolume-based reward (R_H) was slightly higher for the random policy. This is expected since random weight changes lead to broader, less focused exploration, which can result in a more uniformly distributed Pareto front and increased global coverage. The PPO policy, in contrast, concentrates its search around the target region, leading to slightly lower global hypervolume. In terms of contribution, R_H accounted for only 2.46% of the PPO policy's total reward, indicating that this component plays a minimal role in directing the agent's behavior. This is consistent with the reward design, which intentionally downweights global spread in favor of alignment with user preferences. Elevating the influence of R_H could inadvertently shift the agent's focus away from targeted search, which would be counterproductive in preference-driven contexts.

For the diversity-based reward (R_S), the random policy again slightly outperformed PPO. This outcome is intuitive, as randomness tends to promote variation in weight adjustment and, therefore, greater dispersion of solutions. However, this diversity comes at the expense of proximity to the target, illustrating the fundamental trade-off between exploration and preference alignment. Nevertheless, R_S contributed 19.75% of the PPO policy's total reward on average, showing that diversity remains an important, though secondary, factor in guiding the search.

Overall, the total reward accumulated by PPO (5.27) was significantly higher than that of the random policy (3.39), underscoring the effectiveness of PPO in fulfilling the objectives encoded in the reward function. This superior performance is largely attributed to its ability to consistently satisfy the R_P component, which is the principal driver of preference-aware behavior.

In summary, the ablation analysis confirms the efficacy of the reward design in RL-MTSA. While solution diversity and overall hypervolume are still considered through R_S and R_H, the strong influence of R_P ensures that the algorithm remains focused on user-defined trade-off regions. This structured balance enables RL-MTSA to deliver targeted and practically relevant solutions in complex multi-objective routing scenarios.

5.3 Discussion and Results Analysis

The experimental results confirm the capability of RL-MTSA to effectively enhance multi-objective vehicle routing optimization by integrating reinforcement learning into the search process. The dominance-based comparison clearly demonstrates that

RL-MTSA consistently identifies solutions that outperform those generated by the baseline MTSA, validating the benefit of reinforcement-driven weight adjustment in guiding the search toward user-preferred trade-off regions.

Among the two reinforcement learning agents tested, Proximal Policy Optimization (PPO) exhibits the strongest and most consistent performance, achieving dominance in over 80% of experimental runs. Advantage Actor-Critic (A2C), while still outperforming MTSA in a majority of cases, shows lower stability with a dominance rate of around 70%. The superior performance of PPO suggests its robustness in maintaining an effective balance between exploration and exploitation, resulting in more reliable adaptation across diverse benchmark instances.

The relatively small proportion of cases where MTSA dominates RL-MTSA highlights that the reinforcement learning integration does not universally improve performance but instead excels in most preference-driven settings. This indicates that while RL-MTSA shifts the search toward more relevant solution regions, the quality of learning plays a decisive role in determining overall effectiveness. Importantly, neutral outcomes remain minimal, suggesting that reinforcement-guided search rarely converges to equivalent-quality solutions when compared to baseline MTSA.

The localized HV results show that RL-MTSA consistently outperforms MTSA across all target regions. For every instance group and every target level, RL-MTSA achieves higher localized hypervolume, with the strongest gains appearing near the user-specified preference points. These results indicate that RL-MTSA more effectively concentrates search effort in the desired regions of the Pareto frontier, producing higher-quality solutions where user preference matters most.

The reward analysis confirms that the proximity-based component (R_p) is the primary driver of learning, as PPO consistently achieves higher values than the random policy, effectively guiding the search toward the user-specified Pareto region. In contrast, the random policy attains higher hypervolume (R_h) and diversity (R_s) rewards due to its uniform exploration behavior. Despite this, PPO achieves a higher total reward (R_{total}), indicating that the reward design successfully prioritizes preference alignment over global exploration.

Overall, these findings reinforce the practical value of RL-MTSA for logistics and routing problems where decision-makers are not only concerned with broad Pareto coverage but also with obtaining high-quality solutions aligned with specific service-cost trade-offs. By embedding preference-awareness through reinforcement learning, RL-MTSA delivers improvements in both the relevance and competitiveness of solutions.

5.4 Chapter Remarks

This chapter presented **Reinforcement Learning Multi-Thread Simulated Annealing (RL-MTSA)**, an extension of MTSA that incorporates reinforcement learning to guide thread-level weight adjustments. The integration of learning enables RL-MTSA to align the optimization process with user-defined trade-off regions, thereby moving beyond uniform frontier exploration.

The dominance-based and localized hypervolume evaluation against baseline MTSA demonstrates that RL-MTSA substantially improves solution quality, particularly when using the PPO agent. A2C also provides competitive results but with less consistency, underscoring the importance of the choice of reinforcement learning algorithm within the framework.

Despite its advantages, RL-MTSA also presents certain limitations. In a small number of cases, MTSA solutions dominate those of RL-MTSA, indicating that reinforcement learning integration does not guarantee improvement in all scenarios. Furthermore, performance depends on the stability of the chosen learning algorithm, as demonstrated by the contrast between PPO and A2C.

In summary, RL-MTSA represents a significant advancement in preference-aware multi-objective optimization for vehicle routing problems. By embedding adaptive reinforcement learning into a scalable simulated annealing framework, it provides a practical, interpretable, and effective tool for decision-makers who require solutions concentrated near their strategic trade-off preferences.

Chapter 6

Discussion and Contributions

This chapter presents a comprehensive discussion of the research findings and the integrated methodological framework developed in this dissertation. The discussion is structured to emphasize (1) the interrelationship among the three research components including MO-VRPTWDP formulation, MTSA algorithm, and RL-MTSA framework; (2) the key findings derived from computational experiments; (3) the theoretical and methodological contributions of this work; (4) its broader contributions to knowledge science; and (5) practical implications for real-world routing system.

6.1 Integration Across Research Components

The research presented in this dissertation follows a structured and layered methodology, moving from problem formulation to algorithmic development and, ultimately, to adaptive, preference-aware optimization. Each component contributes uniquely to the goal of solving the Multi-Objective Vehicle Routing Problem with Time Windows and Demand Priority (MO-VRPTWDP), while also serving as a foundation for the subsequent component. The three research components, including MO-VRPTWDP formulation, MTSA algorithm, and RL-MTSA framework, are designed not as isolated modules but as a coherent, interdependent system.

MO-VRPTWDP Formulation: Mixed Integer Linear Programming Modeling

The first component introduces a novel formulation of the VRP that explicitly incorporates customer service priorities through a soft-priority model. By modeling customer waiting time as a weighted term in a secondary objective function, the

MO-VRPTWDP formulation enables the routing system to balance efficiency (measured as total travel distance) with responsiveness (captured by adjusted waiting time based on priority). This dual-objective structure reflects real-world trade-offs between cost and service equity. Importantly, the MILP model provides both a theoretical foundation and a basis for small-scale validation, offering a ground truth against which heuristic solutions can be assessed.

MTSA: Meta-Heuristics Algorithm Design

The second component, Multi-Thread Simulated Annealing (MTSA), is developed to handle the computational complexity introduced by the MO-VRPTWDP formulation. Traditional Simulated Annealing (SA), while powerful in local search, lacks scalability and diversity in high-dimensional multi-objective contexts. MTSA addresses these limitations by introducing a multi-thread structure where each thread explores the Pareto frontier from a different weight. Control threads focus on specific weighted sums of objectives, while rogue threads introduce variability and promote diversity. The use of adaptive weight reassignment and return-to-point policies ensures that MTSA can maintain broad coverage across the Pareto frontier while converging efficiently toward high-quality solutions. Thus, MTSA operationalizes the mathematical model from the first component by providing a scalable and efficient mechanism for approximating the Pareto frontier.

RL-MTSA: Adaptive and Preference-Aware Optimization

The final component, RL-MTSA, builds directly on the MTSA structure but augments it with reinforcement learning (RL) to incorporate user preferences. While MTSA explores the Pareto frontier uniformly, RL-MTSA enables the optimization process to be guided toward user-defined regions of interest, such as a preferred trade-off between travel distance and waiting time. The RL agent observes the performance of MTSA threads, interprets their alignment with a user-specified target point on the Pareto frontier, and adjusts the search weights of RL-driven threads accordingly. This dynamic adjustment transforms MTSA from a passive optimizer into an intelligent, learning-driven system that actively aligns its search behavior with strategic preferences. The RL environment includes a reward structure that balances proximity to the target point, contribution to hypervolume, and solution diversity, reinforcing behaviors that produce relevant and diverse solutions.

Together, these components form a complete pipeline from mathematical model formulation to practical decision support. The MO-VRPTWDP formulation cap-

tures complex service-level considerations. MTSA enables tractable and effective approximation of the solution space. RL-MTSA introduces user-preference adaptivity, making the framework not just efficient but also aligned with stakeholder values. This integration supports both strategic planning (through trade-off visualization and Pareto analysis) and operational decision-making (by focusing search on actionable regions). The proposed methodology signifies a conceptual shift from perceiving optimization as a process of mere minimization to one of alignment, wherein the decision-maker's preferences are systematically integrated into the core of the computational process.

6.2 Key Findings

This section synthesizes the major findings derived from the modeling, algorithm development, and experimental evaluation phases of this research. The results confirm the effectiveness of the proposed MO-VRPTWDP formulation and the performance superiority of the developed MTSA and RL-MTSA algorithms. The findings are structured into three subsections corresponding to the major components of the study.

6.2.1 VRPTWDP Formulation

The Mixed-Integer Linear Programming (MILP) formulation of the MO-VRPTWDP provides a solid theoretical framework to model the complexity of real-world distribution problems where both efficiency and equity are essential. The key findings from the formulation are as follows:

1. **Trade-off Structure:** The dual-objective formulation, minimizing total travel distance and waiting time adjusted by demand priority, reveals a well-structured and interpretable trade-off surface, wherein solutions emphasizing distance minimization inevitably increase customer waiting times, whereas those prioritizing waiting time reduction result in longer travel distances.

2. **Soft-Priority Flexibility:** By weighting the waiting time objective according to customer priority, the model enables differentiated service without introducing rigid constraints. This formulation provides a flexible middle ground between strict priority queuing and uniform service, allowing planners to tune the balance based on context-specific requirements.

3. **Behavioral Sensitivity:** Empirical results show that changes in priority scores or time window constraints can cause shifts in route assignments. This highlights the practical utility of the model for simulating operational policies and assessing the impact of priority schemes before implementation.

4. **Ground Truth Benchmarking:** Although limited to small instances due to computational constraints, the MILP model serves as a ground truth validator, enabling performance benchmarking and approximation quality assessment for heuristic methods such as MTSA and RL-MTSA.

6.2.2 MTSA Performance

The Multi-Thread Simulated Annealing (MTSA) algorithm demonstrates significant improvements over traditional metaheuristics such as MOSA in both solution quality and diversity. The following key findings summarize MTSA's performance:

1. **High-Quality Frontier Approximation:** MTSA consistently produces Pareto frontiers with higher hypervolume (HV) values across the Solomon benchmark set, indicating that the algorithm is capable of generating solutions that dominate a larger portion of the objective space.

2. **Enhanced Diversity and Uniformity:** The algorithm maintains a well-distributed set of solutions along the trade-off curve, as measured by diversity metrics such as the Schott Spacing index. This ensures that decision-makers are presented with a comprehensive spectrum of trade-offs.

3. **Efficient Convergence via Thread Cooperation:** The multi-thread structure allows each thread to explore different weight vectors in parallel. Adaptive weight assignment and the use of a "return-to-point" strategy help guide threads back to promising regions, enhancing convergence without sacrificing diversity.

4. **Robustness Across Problem Types:** MTSA performs well across all instance categories in the Solomon benchmark (C, R, and RC), indicating its robustness in handling varying customer distributions and time window constraints.

6.2.3 RL-MTSA Effectiveness

The Reinforcement Learning MTSA (RL-MTSA) framework significantly improves upon the baseline MTSA by introducing intelligent, preference-aware search guid-

ance. The following findings highlight the effectiveness of this hybrid approach:

1. **Dominance Superiority of PPO-Guided Search:** The PPO-guided RL-MTSA variant dominates the MTSA-generated Pareto frontier in over 80% of benchmark runs. This demonstrates that the reinforcement learning agent is able to learn effective weight adjustment strategies that result in consistently superior solutions.

2. **Convergence to Preferred Solutions:** RL-driven thread adjustment significantly reduces the number of iterations required to discover high-quality solutions near the user-defined target point. This not only improves computational efficiency but also aligns the optimization process with decision-makers' operational goals.

3. **Generalization to Unseen Instances:** Despite being trained on a subset of problem instances, the RL agent generalizes well to unseen Solomon instances, indicating that the learned search strategies are transferable across similar problem structures.

6.3 Research Contributions

This dissertation presents a methodological advancement that bridges mathematical modeling, algorithmic design, and adaptive optimization in the context of multi-objective vehicle routing problem with time windows and demand priority constraints. The key contributions of this research are summarized as follows:

1. **Modeling Contribution:** This study introduces the MO-VRPTWDP, a novel formulation that integrates customer priority into the classical VRPTW through a soft-priority mechanism. The proposed model allows priority levels to influence waiting time without enforcing rigid service orders, thereby enhancing the realism of service differentiation. Computational analysis reveals a clear trade-off between total travel distance and customer waiting time, demonstrating how priority weighting can systematically shape routing outcomes.

2. **Algorithmic Contribution:** A new metaheuristic algorithm, MTSA, is developed to efficiently approximate the Pareto frontier of the MO-VRPTWDP. The design retains the strong exploitation capability of simulated annealing while enhancing exploration through a multi-thread structure. Unlike conventional single-thread SA or evolutionary approaches, MTSA employs multiple

parallel search threads, each initialized with distinct weight configurations to explore diverse regions of the trade-off space. Adaptive weight reassignment and a return-to-point mechanism further promote balanced search behavior by redirecting threads toward underexplored yet promising regions. Together, these mechanisms improve solution quality, diversity, and convergence speed, demonstrating the robustness of MTSA across varied instance types and trade-off scenarios.

3. **Preference-Aware Optimization:** The dissertation proposes *RL-MTSA*, a hybrid framework that integrates reinforcement learning with MTSA to enable preference-aware multi-objective optimization in vehicle routing problem. The framework allows the user to specify a target region on the Pareto frontier before optimization begins, and a reinforcement learning agent adaptively adjusts the thread-level search weights to focus on that region. This method moves beyond uniform Pareto frontier exploration and enables region-focused optimization aligned with stakeholder preferences. By embedding a user-specified target into the reward structure, RL-MTSA offers a scalable and generalizable approach to preference-aware decision support in complex logistics settings.

6.4 Contributions to Knowledge Science

Beyond its technical and domain-specific outcomes, this dissertation contributes to the broader field of knowledge science. The framework developed in this study offers insights that are applicable not only to operations research but also to interdisciplinary domains such as knowledge engineering, adaptive systems, and decision support. The key contributions to knowledge science are outlined below:

1. **Preference-Aware Optimization in Knowledge-Driven Systems:** This work demonstrates a principled way to embed user preferences directly into the search behavior of combinatorial optimization processes. Through the use of reinforcement learning agents trained to prioritize specific regions of the Pareto frontier, the RL-MTSA framework provides a pathway toward optimization systems that reflect contextual relevance.

2. **Integrative and Cross-Paradigm Design:** This work unifies mathematical modeling, multi-objective metaheuristic search, and reinforcement learning into an integrated framework. By connecting the formulation, the MTSA

algorithm, and the preference-aware RL extension, the study demonstrates how distinct research components can be systematically combined to create a decision support approach.

In summary, this dissertation contributes to knowledge science by modeling how complex optimization can be made adaptive and preference-aware through the integration of mathematical modeling, meta-heuristic algorithms, and reinforcement learning hybridization. It lays the groundwork for future studies that seek to bridge computational efficiency with human-centric relevance in intelligent decision-making systems.

6.5 Practical Implications

The proposed framework, encompassing the MO-VRPTWDP formulation, MTSA algorithm, and RL-MTSA framework, provides not only theoretical advances but also tangible tools for real-world deployment. This section elaborates on how the methods developed in this study can support operational decision-making in logistics and supply chain contexts, especially where competing objectives and user preferences are critical.

6.5.1 For Logistics Planners

In real-world transportation and logistics settings, planners often face the dual challenge of minimizing costs while ensuring timely and prioritized service to customers with varying levels of importance. The proposed framework offers several practical advantages for such decision-makers:

1. **Strategic Cost-Service Trade-offs:** The MO-VRPTWDP model allows planners to visualize and select routing solutions that balance operational cost (total travel distance) against customer satisfaction metrics (priority-weighted waiting time).

2. **Soft-Priority Enforcement Without Rigidity:** Traditional approaches often rely on hard constraints to enforce customer priority, which may lead to infeasibility or operational inefficiencies. The proposed soft-priority mechanism allows planners to respect priority hierarchies while retaining scheduling flexibility.

3. **Customizable Search Alignment:** The RL-MTSA's user-specified point functionality empowers planners to guide the optimization process toward regions of interest based on organizational goals.

4. **Sensitivity Analysis for Policy Tuning:** The formulation and solution framework support scenario testing by varying input weights, demand priorities, or time windows. This enables planners to conduct what-if analyses for capacity planning, resource allocation, or demand fluctuation management.

6.5.2 For Decision Support Systems

Beyond manual planning tasks, the developed methods offer robust capabilities for integration into algorithmic decision support systems (DSS) in logistics platforms, transportation management systems (TMS), or enterprise resource planning (ERP) systems:

1. **Preference-Aware Initialization:** The RL-MTSA model enables decision-makers to specify their trade-off preference or region of interest *prior to optimization*. This initial input, defined through a user-specified target point on the Pareto frontier, guides the reinforcement learning agent throughout the optimization process. Unlike conventional methods that explore the entire frontier indiscriminately, RL-MTSA concentrates its search efforts in the relevant region from the outset, enhancing efficiency and aligning the output with user-defined strategic goals. This preference specification offers significant practical value in structured planning environments where decision criteria are known in advance.

2. **Scalability for Large-Scale Deployment:** MTSA's multi-thread design and RL-MTSA's region-focused exploration reduce unnecessary search effort, allowing rapid convergence to relevant solutions. This is particularly useful in high-frequency, large-scale logistics networks where decisions must be made in minutes, not hours.

3. **Enhanced Responsiveness in Critical Applications:** In sectors such as healthcare logistics, vaccine delivery, emergency response, or just-in-time manufacturing, the ability to generate preference-aligned solutions rapidly can lead to significant performance gains. RL-MTSA enables DSS to suggest solutions that align with both operational constraints and real-time policy shifts.

In summary, the practical value of this research lies not only in its ability to produce high-quality solutions but also in its adaptability to context, preferences, and operational constraints, an essential capability for modern logistics systems.

Chapter 7

Conclusion

This chapter concludes the dissertation by summarizing the main contributions and findings of the research and outlining its limitations and potential avenues for future work. The chapter is structured into two sections: the first provides the overall concluding remarks that synthesize the results and contributions, while the second discusses limitations and proposes directions for extending this work.

7.1 Concluding Remarks

This dissertation presented a novel hybrid preference-aware optimization framework, Reinforcement Learning Multi-Thread Simulated Annealing (RL-MTSA), to address the Multi-Objective Vehicle Routing Problem with Time Windows and Demand Priority (MO-VRPTWDP). The research was motivated by the limitations of conventional multi-objective optimization techniques, which distribute computational effort across the entire Pareto frontier and lack mechanisms to focus the search on user-specified regions of interest. To address these challenges, the dissertation advances three interconnected components: a mathematical model that formulates MO-VRPTWDP, a metaheuristic (MTSA) that provides scalable Pareto frontier approximation, and an adaptive reinforcement learning extension (RL-MTSA) that directs the search toward preference regions. Together, these components form a methodology for targeted and preference-aware optimization in complex routing problems.

The development proceeds in stages, beginning with the formulation of the MO-VRPTWDP, a new problem variant that jointly minimizes travel distance and weighted customer waiting time. This formulation introduces a soft-priority mechanism, allowing service differentiation without imposing strict priority constraints. The weighted sum structure facilitates flexible trade-off analysis, making the model

suitable for diverse operational scenarios in logistics, where balancing cost-efficiency with customer satisfaction is essential.

To solve large-scale instances of MO-VRPTWDP efficiently, a Multi-Thread Simulated Annealing (MTSA) algorithm was developed. MTSA employs a cooperative thread structure in which each thread explores a different region of the Pareto frontier using distinct weight vectors. The framework includes mechanisms such as adaptive weight adjustment and return-to-point strategies that enhance solution quality, diversity, and convergence stability. Experimental results show that MTSA outperforms baseline metaheuristics, such as MOSA, in hypervolume and distribution metrics across benchmark instances.

Building on this foundation, the RL-MTSA framework incorporates reinforcement learning agents, trained using Proximal Policy Optimization (PPO) and Advantage Actor-Critic (A2C), to adjust the weight configuration of MTSA threads dynamically. Given a user-specified target point on the Pareto frontier, the RL agent guides the search toward that region, enabling the generation of solutions aligned with decision-maker preferences. Empirical evaluations demonstrate that PPO-guided RL-MTSA dominates the MTSA Pareto front in over 80% of benchmark runs and achieves superior localized hypervolume in the vicinity of target solutions.

The contributions of this dissertation are threefold: (1) the formulation of a new bi-objective VRP variant that integrates time windows and demand priority; (2) the development of the MTSA algorithm, a scalable and cooperative multi-thread simulated annealing framework for Pareto approximation; and (3) the proposal of RL-MTSA, the new reinforcement learning-enhanced simulated annealing method for preference-aware optimization in multi-objective routing. Collectively, these contributions advance both the theory and practice of multi-objective optimization by demonstrating how structured modeling, meta-heuristic search, and learning mechanisms can be effectively combined.

In conclusion, this dissertation provides a comprehensive and generalizable methodology for solving multi-objective vehicle routing problems with user-defined trade-offs. By layering mathematical modeling, metaheuristic search, and reinforcement learning, the proposed framework demonstrates how computational intelligence can be harnessed to create preference-aware and decision-aligned optimization tools. This work contributes not only to the vehicle routing literature but also to the broader field of intelligent decision support for preference-aware optimization systems.

7.2 Limitations and Future Directions

The limitations and future directions of this research can be discussed in three parts: problem formulation, metaheuristic development, and reinforcement learning extension.

Problem Formulation

For the mathematical formulation part, the main limitation lies in the simplifying assumptions made when formulating MO-VRPTWDP, particularly in its treatment of uncertainty and customer satisfaction. The current formulation assumes deterministic parameters such as travel times and demands, and it represents customer satisfaction only through waiting time with soft-priority adjustment. Future work could address these issues by extending the model to incorporate stochastic or dynamic elements, such as probabilistic travel times or real-time customer requests, and by considering additional service quality dimensions beyond waiting time, including reliability, fairness, or customer-specific service levels.

Metaheuristic Development

For the metaheuristic development part, the main limitations concern the scope of evaluation and the limited cooperation among threads. MTSA has been tested only on the bi-objective VRPTWDP, and its performance in problems with more than two objectives remains uncertain. In addition, the current framework allows only limited information exchange between threads, which may restrict opportunities to leverage collective search experience for improved performance. Future research could therefore extend MTSA to many-objective optimization contexts, such as multi-objective scheduling or resource allocation, and design enhanced mechanisms for sharing and utilizing search information across threads to guide exploration more effectively.

Reinforcement Learning Extension

For the reinforcement learning extension part, the main limitations are partly inherited from MTSA, particularly the fact that the framework has so far been evaluated only in a bi-objective setting. Its performance and scalability in many-objective optimization contexts remain uncertain. In addition, while RL-MTSA is designed to concentrate the search around user-specified regions of the Pareto frontier, this focus can also increase the risk of converging to local optima. Striking the right balance

between exploiting the target region and maintaining sufficient exploration of the broader search space remains a challenge. Future research could therefore investigate mechanisms that dynamically adjust this balance, for example, by integrating adaptive exploration strategies or hybridizing RL-MTSA with diversity-preserving techniques. Moreover, extending RL-MTSA to many-objective problems would further strengthen its applicability and robustness in more complex optimization settings.

Bibliography

Avila-Torres, P. A., Arratia-Martinez, N. M., and Ruiz-y Ruiz, E. (2020). The inventory routing problem with priorities and fixed heterogeneous fleet. *Applied Sciences*, 10(10):3502.

Baños, R., Ortega, J., Gil, C., Fernández, A., and De Toro, F. (2013). A simulated annealing-based parallel multi-objective approach to vehicle routing problems with time windows. *Expert Systems with Applications*, 40(5):1696–1707.

Baradaran, V., Shafaei, A., and Hosseinian, A. H. (2019). Stochastic vehicle routing problem with heterogeneous vehicles and multiple prioritized time windows: Mathematical modeling and solution approach. *Computers & Industrial Engineering*, 131:187–199.

Barma, P. S., Dutta, J., Mukherjee, A., and Kar, S. (2023). A bi-objective latency based vehicle routing problem using hybrid grasp-nsgaii algorithm. *International Journal of Management Science and Engineering Management*, 18(3):190–207.

Beheshti, A. K., Hejazi, S. R., and Alinaghian, M. (2015). The vehicle routing problem with multiple prioritized time windows: A case study. *Computers & Industrial Engineering*, 90:402–413.

Bono, G., Dibangoye, J. S., Simonin, O., Matignon, L., and Pereyron, F. (2021). Solving multi-agent routing problems using deep attention mechanisms. *IEEE Transactions on Intelligent Transportation Systems*, 22(12):7804–7813.

Chen, J., Ding, J., Li, K., Tan, K. C., and Chai, T. (2024). A knee point driven evolutionary algorithm for multiobjective bilevel optimization. *IEEE Transactions on Cybernetics*.

Dantzig, G. B. and Ramser, J. H. (1959). The truck dispatching problem. *Management science*, 6(1):80–91.

Das, D. N., Sewani, R., Wang, J., and Tiwari, M. K. (2020). Synchronized truck and drone routing in package delivery logistics. *IEEE Transactions on Intelligent Transportation Systems*, 22(9):5772–5782.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197.

Doan, T. T., Bostel, N., and Hà, M. H. (2021). The vehicle routing problem with relaxed priority rules. *EURO Journal on Transportation and Logistics*, 10:100039.

El-Sherbeny, N. A. (2010). Vehicle routing with time windows: An overview of exact, heuristic and metaheuristic methods. *Journal of King Saud University-Science*, 22(3):123–131.

Eslamipoor, R. (2024). Direct and indirect emissions: a bi-objective model for hybrid vehicle routing problem. *Journal of Business Economics*, 94(3):413–436.

Fitzpatrick, J., Ajwani, D., and Carroll, P. (2024). A scalable learning approach for the capacitated vehicle routing problem. *Computers & Operations Research*, 171:106787.

Gao, M., Chen, Y., Zhang, Z., and Wahab, M. (2025). Learning-based column generation approach for the vehicle routing problem with release dates and incompatible loading constraints. *Computers & Operations Research*, page 107152.

Ghannadpour, S. F. (2019). Evolutionary approach for energy minimizing vehicle routing problem with time windows and customers' priority. *International Journal of Transportation Engineering*, 6(3):237–264.

Golmohammadi, A.-M., Abedsoltan, H., Goli, A., and Ali, I. (2024). Multi-objective dragonfly algorithm for optimizing a sustainable supply chain under resource sharing conditions. *Computers & Industrial Engineering*, 187:109837.

Guo, T., Mei, Y., Tang, K., and Du, W. (2023). A knee-guided evolutionary algorithm for multi-objective air traffic flow management. *IEEE Transactions on Evolutionary Computation*.

Hu, Y., Yao, Y., and Lee, W. S. (2020). A reinforcement learning approach for optimizing multiple traveling salesman problems over graphs. *Knowledge-Based Systems*, 204:106244.

Huang, T., Wang, S., and Li, K. (2024). Direct preference-based evolutionary multi-objective optimization with dueling bandits. *Advances in Neural Information Processing Systems*, 37:122206–122258.

Kalatzantonakis, P., Sifaleras, A., and Samaras, N. (2023). A reinforcement learning-variable neighborhood search method for the capacitated vehicle routing problem. *Expert Systems with Applications*, 213:118812.

Kuo, R., Edbert, E., Zulvia, F. E., and Lu, S.-H. (2023a). Applying nsga-ii to vehicle routing problem with drones considering makespan and carbon emission. *Expert Systems with Applications*, 221:119777.

Kuo, R., Luthfiansyah, M. F., Masruroh, N. A., and Zulvia, F. E. (2023b). Application of improved multi-objective particle swarm optimization algorithm to solve disruption for the two-stage vehicle routing problem with time windows. *Expert Systems with Applications*, 225:120009.

Li, J., Ma, Y., Gao, R., Cao, Z., Lim, A., Song, W., and Zhang, J. (2022a). Deep reinforcement learning for solving the heterogeneous capacitated vehicle routing problem. *IEEE Transactions on Cybernetics*, 52(12):13572–13585.

Li, K., Lai, G., and Yao, X. (2023). Interactive evolutionary multiobjective optimization via learning to rank. *IEEE Transactions on Evolutionary Computation*, 27(4):749–763.

Li, Y., Ye, C., Wang, H., Wang, F., and Xu, X. (2022b). A discrete multi-objective grey wolf optimizer for the home health care routing and scheduling problem with priorities and uncertainty. *Computers & Industrial Engineering*, 169:108256.

Lin, B., Ghaddar, B., and Nathwani, J. (2022). Deep reinforcement learning for the electric vehicle routing problem with time windows. *IEEE Transactions on Intelligent Transportation Systems*, 23(8):11528–11538.

Liu, W., Wang, R., Zhang, T., Li, K., Li, W., Ishibuchi, H., and Liao, X. (2023). Hybridization of evolutionary algorithm and deep reinforcement learning for multiobjective orienteering optimization. *IEEE Transactions on Evolutionary Computation*, 27(5):1260–1274.

Long, J., Sun, Z., Pardalos, P. M., Hong, Y., Zhang, S., and Li, C. (2019). A hybrid multi-objective genetic local search algorithm for the prize-collecting vehicle routing problem. *Information Sciences*, 478:40–61.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T. P., Harley, T., Silver, D., and Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning.

Nucamendi-Guillén, S., Flores-Díaz, D., Olivares-Benitez, E., and Mendoza, A. (2020). A memetic algorithm for the cumulative capacitated vehicle routing problem including priority indexes. *Applied Sciences*, 10(11):3943.

Pan, W. and Liu, S. Q. (2023). Deep reinforcement learning for the dynamic and uncertain vehicle routing problem. *Applied Intelligence*, 53(1):405–422.

Phiboonbanakit, T., Horanont, T., Huynh, V.-N., and Supnithi, T. (2021). A hybrid reinforcement learning-based model for the vehicle routing problem in transportation logistics. *IEEE Access*, 9:163325–163347.

Pugliese, L. D. P., Ferone, D., Festa, P., Guerriero, F., and Macrina, G. (2023). Combining variable neighborhood search and machine learning to solve the vehicle routing problem with crowd-shipping. *Optimization Letters*, pages 1–23.

Qin, W., Zhuang, Z., Huang, Z., and Huang, H. (2021). A novel reinforcement learning-based hyper-heuristic for heterogeneous vehicle routing problem. *Computers & Industrial Engineering*, 156:107252.

Schott, J. R. (1995). *Fault tolerant design using single and multicriteria genetic algorithm optimization*. PhD thesis, Massachusetts Institute of Technology.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms.

Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations research*, 35(2):254–265.

Srivastava, G., Singh, A., and Mallipeddi, R. (2021). Nsga-ii with objective-specific variation operators for multiobjective vehicle routing problem with time windows. *Expert Systems with Applications*, 176:114779.

Sun, J. and Wang, R. (2023). Multi-objective optimization of a sustainable two echelon vehicle routing problem with simultaneous pickup and delivery in construction projects. *Journal of Engineering Research*.

Suppapitnarm, A., Seffen, K. A., Parks, G. T., and Clarkson, P. (2000). A simulated annealing algorithm for multiobjective optimization. *Engineering optimization*, 33(1):59–85.

Toth, P. and Vigo, D. (2002). *The vehicle routing problem*. SIAM.

Wang, Y., Wang, L., Chen, G., Cai, Z., Zhou, Y., and Xing, L. (2020). An improved ant colony optimization algorithm to the periodic vehicle routing problem with time window and service choice. *Swarm and Evolutionary Computation*, 55:100675.

Wu, Y., Song, W., Cao, Z., Zhang, J., and Lim, A. (2022). Learning improvement heuristics for solving routing problems. *IEEE transactions on neural networks and learning systems*, 33(9):5057–5069.

Wu, Y. and Zeng, B. (2023). Dynamic parcel pick-up routing problem with prioritized customers and constrained capacity via lower-bound-based rollout approach. *Computers & Operations Research*, 154:106176.

Xu, J., Li, K., and Abusara, M. (2022). Preference based multi-objective reinforcement learning for multi-microgrid system optimization problem in smart grid. *Memetic Computing*, 14(2):225–235.

Xu, W. and Yu, X. (2023). A multi-objective multi-verse optimizer algorithm to solve environmental and economic dispatch. *Applied Soft Computing*, 146:110650.

Yu, X., Duan, Y., and Luo, W. (2022). A knee-guided algorithm to solve multi-objective economic emission dispatch problem. *Energy*, 259:124876.

Zhang, K., Shen, C., He, J., and Yen, G. G. (2021). Knee based multimodal multi-objective evolutionary algorithm for decision making. *Information Sciences*, 544:39–55.

Zhang, S., Liu, S., Xu, W., and Wang, W. (2022). A novel multi-objective optimization model for the vehicle routing problem with drone delivery and dynamic flight endurance. *Computers & Industrial Engineering*, 173:108679.

Zhang, Z., Wu, Z., Zhang, H., and Wang, J. (2023). Meta-learning-based deep reinforcement learning for multiobjective optimization problems. *IEEE Transactions on Neural Networks and Learning Systems*, 34(10):7978–7991.

Zhao, J., Mao, M., Zhao, X., and Zou, J. (2021). A hybrid of deep reinforcement learning and local search for the vehicle routing problems. *IEEE Transactions on Intelligent Transportation Systems*, 22(11):7208–7218.

Zhao, L., Ren, Y., Zeng, Y., Cui, Z., and Zhang, W. (2022). A knee point-driven many-objective pigeon-inspired optimization algorithm. *Complex & Intelligent Systems*, 8(5):4277–4299.

Zidi, I., Mesghouni, K., Zidi, K., and Ghedira, K. (2012). A multi-objective simulated annealing for the multi-criteria dial a ride problem. *Engineering Applications of Artificial Intelligence*, 25(6):1121–1131.

Zitzler, E., Laumanns, M., and Thiele, L. (2001). Spea2: Improving the strength pareto evolutionary algorithm. *Evolutionary Methods for Design, Optimization and Control*, pages 95–100.

Zou, Y., Hao, J.-K., and Wu, Q. (2024a). A reinforcement learning guided hybrid evolutionary algorithm for the latency location routing problem. *Computers & Operations Research*, 170:106758.

Zou, Y., Wu, H., Yin, Y., Dhamotharan, L., Chen, D., and Tiwari, A. K. (2024b). An improved transformer model with multi-head attention and attention to attention for low-carbon multi-depot vehicle routing problem. *Annals of Operations Research*, 339(1):517–536.

List of Publications

International Journal

- Thanapat Leelertkij, Jirachai Buddhakulsomsiri, and Van-Nam Huynh. A multi-thread simulated annealing for multi-objective vehicle routing problem with time windows and demand priority. *Computers & Industrial Engineering*, Vol. 207, 111253, 2025.
  DOI: 10.1016/j.cie.2025.111253

International Conferences

- Thanapat Leelertkij, Parthana Parthanadee, Jirachai Buddhakulsomsiri, and Van-Nam Huynh. A hybrid reinforcement learning and simulated annealing approach for user-specified optimization in multi-objective VRP. *International Conference on Mechanical Manufacturing and Industrial Engineering*, 26-29 August 2025, Hosei University, Tokyo, Japan. (Accepted)