Universiteit Leiden

Master Computer Science

A Hybrid Multi-Objective Evolutionary Algorithm for the Vehicle Routing Problem with Time Windows

Name:	Wenqian Hu
Student ID:	S2909278
Date:	23/08/2023

Specialisation: Artificial Intelligence

1st supervisor: Yingjie Fan
2nd supervisor: Mitra Baratchi
3rd supervisor: Thomas Bäck

Master’s Thesis in Computer Science

Leiden Institute of Advanced Computer Science
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

The Vehicle Routing Problem with Time Windows (VRPTW) is an important class of transportation problems with significant applications in real-world logistics and transportation. This paper introduces a novel approach known as the hybrid multi-objective evolutionary algorithm, which integrates similarity measurement, local search methods, and a non-dominated sorting selection operator to tackle this issue. Its performance is tested on two datasets, one being the benchmark dataset Solomon's benchmark instance for VRPTW, and the other consisting of 45 instances from the real-world dataset 2021 Amazon Last Mile Routing Research Challenge. The outcomes showcase that our algorithm outperforms state-of-the-art approaches, underscoring its suitability for real-world applications. Our code is available on GitHub[1].
KEYWORDS: Vehicle routing problem with time windows, Multiobjective optimization, Evolutionary algorithms

[1] https://github.com/hwqdddddd/MasterThesis

Contents

Chapter 1

Introduction

The vehicle routing problem (VRP) has emerged as a paramount and extensively investigated combinatorial optimization problem over the last several decades. Its pertinence is underscored by many real-world applications across domains such as supply chain management, transportation, production management, and so on [17, 19]. This intricate problem involves efficiently delivering specified merchandise to clientele, endeavoring to minimize expenses while accommodating diverse objectives and constraints. The journey commences and concludes at a singular depot location. The Vehicle Routing Problem (VRP) is categorized as an $\mathcal{NP}$-hard problem, stemming from its relationship with the traveling salesman problem [7].

The VRP has several well-known variants, each introducing specific complexities and constraints to the basic problem. For example, Capacitated VRP (CVRP) is the variant that each vehicle has a limited capacity to carry goods. Multi-Depot VRP (MDVRP) is the variant in that multiple depots are available, and vehicles must be assigned to these depots while optimizing routes and satisfying customer demands. The Vehicle Routing Problem with Time Windows (VRPTW) is a specific variation of the VRP where customers have specified time windows and vehicles have limited capacity. This variant is highly relevant to real-life applications [23]. As an extension of the classic VRP, the VRPTW introduces an additional temporal dimension that encapsulates the time constraints associated with the delivery or service windows at customer locations. This augmentation reflects the realistic constraints faced by modern logistics and distribution systems, where punctuality in deliveries and services is imperative.

The VRPTW has garnered substantial attention within the academic and industrial domains due to its profound implications for enhancing operational efficiency, reducing transportation costs, and ameliorating environmental impacts. The aim of VRPTW is to identify a collection of routes that collectively incur the lowest expenses, encompassing factors such as the number of vehicles used, overall travel distance, and etc. This cost calculation also encompasses penalties associated with instances where vehicles arrive excessively early or late. This is particularly pertinent since our approach considers time windows as flexible constraints, permitting slight violations while incurring penalties. The overarching goal is to minimize all these costs simultaneously. Therefore, the VRPTW could formulate as a Multi-objective Optimization Problem (MOP). Castro-Gutierrez et al. [4] defined 5 common objectives for the multi-objective VRPTW problem, which are the objective for this paper. The set of objectives comprises the following five facets: minimizing the count of vehicles/routes, the cumulative travel distance covered by all vehicles, the makespan, which represents the longest travel time among all routes, the overall waiting time attributed to early arrivals, and the collective delay time resulting from late arrivals.

The multi-objective VRPTW problem, commonly denoted as MOVRPTW, addresses the need to furnish a spectrum of solutions that capture the intricate trade-offs between objectives, diverging from the standard approach of producing a solitary solution [13]. Despite the existence of various meta-heuristic methods proposed to tackle this intricate problem, this study primarily focuses on multi-objective evolutionary algorithms (MOEAs). MOEAs are known for their effectiveness in handling Multi-Objective Problems (MOPs), as delineated by He et al. [16]. This choice is based on the fundamental characteristics of MOEAs, which rely on a population-based nature. This approach enables a comprehensive and diverse approximation of the Pareto Front (PF), which is a crucial aspect in MOP. MOEAs are one type of evolutionary algorithm (EA), which is inspired by the principles of Darwinian evolution and natural selection. The evolutionary process employs these mechanisms to select and propagate a new ensemble of individuals, distinguished by their enhanced quality to the preceding generation.

Despite the effectiveness of MOEAs in addressing complex optimization problems, they are not immune to certain challenges. One key challenge lies in maintaining population diversity throughout the evolutionary process. As MOEAs evolve solutions toward the Pareto Front, there is a risk of convergence to a limited set of solutions, neglecting potentially valuable areas of the solution space. Moreover, MOEAs can sometimes struggle with premature convergence, where the algorithm gets trapped in local optima and fails to explore promising regions that might lead to better solutions.

To mitigate these challenges and enhance the performance of MOEAs in the context of the Multi-Objective Vehicle Routing Problem with Time Windows (MOVRPTW), this study introduces several contributions.

- A new mechanism called similarity measurement [13] is added to MOEA to maintain diversity for the population.
- After evolution, a local search method is used to find a better solution that prevents to stuck in the local optimal.

The rest of this paper is structured as follows: Chapter 2 presents the background of this problem, which includes previous studies in this field, the definitions of VRPTW, and the preliminary concept of the multi-objective problem. Chapter 3 then explains the proposed hybrid multi-objective evolutionary algorithm for the VRPTW problem and outlines the experimental design of this paper. The results of the experiments are provided in Chapter 4. Finally, Chapter 5 concludes with the findings and future work in this area.

Chapter 2

Background

2.1 Related Work

This section offers a concise summary of the proposed methodologies for tackling the Vehicle Routing Problem with Time Windows (VRPTW).

Numerous investigations have approached the VRPTW as a single-objective problem, often focused on minimizing the total distance. For instance, Brandão de Oliveira et al. [3] utilized simulated annealing, while Ursani et al. [30] employed a genetic algorithm to reduce overall travel distance. However, confining the problem to a solitary objective overlooks the intricate intricacies inherent in practical real-world situations. Moreover, a subgroup of researchers has expanded their investigations to encompass dual objectives within the VRPTW context, aiming to minimize both the number of vehicles and the total travel distance simultaneously. Noteworthy in this regard is the work of Gong et al. [15], which introduces a discrete particle swarm optimization methodology employing a weighted sum approach to combine these objectives into a singular weighted objective function. Nonetheless, the efficacy of such techniques is impeded by the challenge of judiciously assigning weights to objectives.

As previously mentioned, it is more advantageous to view VRPTW as a multi-objective challenge, given the nature of its complexity. In existing literature, numerous studies have employed either generic or hybrid Multi-Objective Evolutionary Algorithms (MOEAs) to address the MOVRPTW and its diverse iterations. Tan et al. [28] introduced a hybridized multi-objective evolutionary algorithm, with a particular emphasis on addressing a problem involving only two objectives. Another notable contribution comes from Rahoual et al. [24], who present a multi-criteria genetic algorithm, derived from the principles of NSGA (Non-dominated Sorting Genetic Algorithm) [26], tailored to tackle the VRPTW. Jozefowiez et al. [18] proposed parallel and hybrid models combined with Elitist Diversification, tabu search, and evolutionary algorithm to tackle the VRP. Similarly, Garcia-Najera et al. [13] introduced a multi-objective evolutionary algorithm. This approach, notable for its incorporation of solution similarity measurement techniques, was designed for the Capacitated VRPTW (CVRPTW). However, they considered three objectives, including the number of vehicles, the total time, and the total distance. Baños et al. [2] contributed a novel multi-objective framework, employing a multi-start simulated annealing strategy named multi-start multi-objective evolutionary algorithm with simulated annealing (MMOEASA) to unravel the complexities of the Multi-Objective VRPTW (MOVRPTW). Most of these works used the well-known benchmark data set called Solomon's instance [25]. Nonetheless, Castro-Gutierrez et al. [4] claim that Solomon's dataset may not ideally align with the requisites of the MOPVRPTW, due to the emergence of relatively weak dependence relationships among the distinct objectives. So we not only tested our algorithm on Solomon's instance but also conducted experiments on the latest data set from Amazon Last Mile Routing Research Challenge [21]. The data provided by Amazon are all real paths in real life, which can better reflect the real situation.

Some researchers tried to combine MOEA with local search methods to obtain better results. Zhou and Wang [33] introduced a novel algorithm based on local search for the MOVRPTW, they developed 5 local search methods to tackle each objective, and they are applied to randomly generated solutions iteratively. Subsequently, Srivastava et al. [27] present an algorithm based on NSGA-II with objective-specific variation operators to deal with MOVRPTW. Zhang et al. [32] develop multi-objective local search (MOLS) algorithms, and enhance MOLS to MOLS+ for the VRP with outsourcing and profit balancing.

Our work contrasts through the introduction of distinctive elements. Specifically, our approach integrates multi-objective evolutionary algorithms with a novel similarity measurement mechanism, which contributes to preserving population diversity. We extend the optimization process by integrating a post-evolution local search method designed to prevent premature convergence and refine solutions. Moreover, our study capitalizes on the latest data set from Amazon [21] to conduct comprehensive empirical assessments. In doing so, we bridge the gap between theoretical methodologies and real-world applications, leveraging authentic path data to more accurately mirror practical contexts within the realm of vehicle routing.

2.2 Preliminary

In this section, we present the formal definition of the VRPTW and introduce some preliminary concepts related to Multiple-Objective Problems (MOP).

2.2.1 Definition of VRPTW

This thesis uses the same formulation and conventions as presented in [13] and [33]. The VRPTW can be modeled as a graph G, there is a set of vertices $\mathcal{V} = \{0, ..., N\}$, each vertex signifies a customer location, where vertex 0 is the depot. Edges $\mathcal{E} = \{(0, i), ..., (i, j)\}$ connecting the vertices correspond to potential routes that vehicles can take to reach customers. Every customer i is situated at coordinates (x_i, y_i), accompanied by a distinct demand for goods denoted as $g_i > 0$, and subjected to a time window constraint $[b_i, e_i]$ that dictates when service should occur. This temporal constraint mandates that the respective customer's requirements are met within the stipulated time window. Additionally, each customer involves a service time s_i requisite for unloading deliveries. The depot is the central station where every vehicle departs and returns with the same capacity $\mathcal{C} \geq \max\ \{g_i : i \in \mathcal{V}\}$, is located at (x_0, y_0), possessing a demand $g_0 = 0$, and a time window $[0, e_0 \geq \max\ \{e_i : i \in \mathcal{V}\}]$. Let d_{ij} and t_{ij} denote distance and travel time from customer i to customer j, respectively.

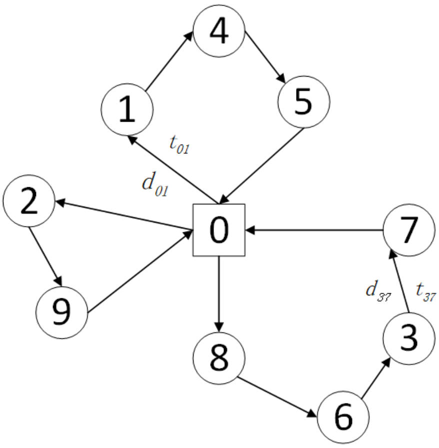

Figure 2.1: Routes representation in VRPTW.

The primary goal of the VRPTW involves determining a collection of m routes, denoted as $\mathcal{R} = \{r_1, ..., r_m\}$, that collectively incur the minimal cost. These routes should adhere to specific conditions: every customer must be serviced exactly once by a single vehicle, and each individual route r_j must initiate and terminate at the designated depot. Let $r_j =< c(1,j),...,c(N_j,j) >$ denote the sequence of customer that visited in j^{th} route which has N_j customers, and $c(i,j)$ is the i^{th} customer in j^{th} route. Notably, the cumulative demand across this route must not surpass the capacity of the respective vehicle. Moreover, the depot represents $c(0,j) = c(N_j + 1, j) = 0$. Figure 2.1 shows an example while $m = 3$ routes of $\mathcal{R} = \{r_1, r_2, r_3\}$. Therefore, the distance traveled, denoted as D_j, for the j^{th} route is defined as follows:

$$D_j = \sum_{i=0}^{N_j} d_{c(i,j)c(i+1,j)} \tag{2.1}$$

In this context, the notation $d_{c(i,j)c(i+1,j)}$ symbolizes the distance from customer i to the subsequent customer $i + 1$ within j^{th} route. To compute the relevant cost, it's essential to take into account $a_{c(i,j)}$ as the arrival time and $l_{c(i,j)}$ as the departure time of vehicle j at customer i. The calculation of vehicle j's arrival time at customer i follows this equation:

$$a_{c(i,j)} = l_{c(i-1,j)} + t_{c(i-1,j),c(i,j)} \tag{2.2}$$

where $l_{c(0,j)}$ is set to 0, as it signifies the vehicle's departure from the depot at time 0. Additionally, $t_{c(i-1,j),c(i,j)}$ signifies the time taken for travel from customer $i - 1$ to customer i.

When the vehicle reaches the customer before the designated service time, it is required to wait, resulting in waiting time. The waiting time experienced by vehicle j at customer i can be expressed as follows:

$$w_{c(i,j)} = \begin{cases} 0, & \text{if } a_{c(i,j)} \geq b_{c(i,j)} \\ b_{c(i,j)} - a_{c(i,j)}, & \text{if } a_{c(i,j)} < b_{c(i,j)} \end{cases} \tag{2.3}$$

Subsequently, the departure time of vehicle j from customer i is

$$l_{c(i,j)} = a_{c(i,j)} + w_{c(i,j)} + s_{c(i,j)}. \tag{2.4}$$

The overall travel time for route r_j can be calculated as:

$$T_j = \sum_{i=0}^{N_j} (t_{c(i,j),c(i+1,j)} + w_{c(i+1,j)} + s_{c(i+1,j)}) \tag{2.5}$$

where $s_{c(i+1,j)}$ signifies the service time of customer $i+1$ on route j, and it's worth noting that $w_{c(N_j+1,j)} = s_{c(N_j+1,j)} = 0$, as the depot's service and waiting times are both 0. The cumulative waiting time for route r_j is then computed as:

$$W_j = \sum_{i=1}^{N_j} w_{c(i,j)}. \tag{2.6}$$

Two distinct variations of VRPTW exist. The initial variant enforces hard time windows, where the vehicle's arrival after the customer's time window closure is prohibited [13, 28]. However, this strict constraint may not accurately represent real-world scenarios. In reality, factors beyond control, such as road conditions and customer-related variables, can significantly impact arrival times [29]. The second variant involves VRP with soft time windows, permitting deviations from the specified time windows and finding utility in numerous real-world contexts, as evidenced by [11, 22]. Notably, the adoption of soft time windows can lead to feasible solutions where the application of a hard time windows approach may falter [11]. In this study, our focus is on VRP with soft time windows, where vehicle arrival after the designated time window conclusion could result in delays. In this context, the delay time of vehicle j at customer i is calculated as:

$$delay_{c(i,j)} = \begin{cases} 0, & \text{if } a_{c(i,j)} \leq e_{c(i,j)} \\ a_{c(i,j)} - e_{c(i,j)}, & \text{otherwise.} \end{cases} \tag{2.7}$$

and the overall delay time for the route r_j is

$$Delay_j = \sum_{i=1}^{N_j} delay_{c(i,j)}. \tag{2.8}$$

The objectives of VRPTW problem can be defined as follows, each of which should be minimized:

f_1 (number of vehicles):

$$f_1 =| \mathcal{R} |= m; \tag{2.9}$$

f_2 (total travel distance):

$$f_2 = \sum_{i=1}^{m} D_j; \tag{2.10}$$

f_3 (longest travel time among all routes):

$$f_3 = \max\{T_j \mid j = 1, ..., m\}; \tag{2.11}$$

f_4 (accumulated waiting time resulting from early arrivals):

$$f_4 = \sum_{i=1}^{m} W_j; \tag{2.12}$$

f_5 (accumulated delay time caused by late arrivals):

$$f_5 = \sum_{i=1}^{m} Delay_j; \tag{2.13}$$

The constraint in VRPTW is that the combined demand of all customers in a route r_j must not exceed the capacity of the vehicle:

$$\sum_{i=1}^{N_j} g_{c(i,j)} \leq \mathcal{C} \quad \forall j = 1, ..., m. \tag{2.14}$$

Therefore, the VRPTW with five objectives could be summarized as

$$\text{min} f = \{f_1, f_2, f_3, f_4, f_5\} \tag{2.15}$$

subject to Equation 2.14.

2.2.2 Multi-objective optimization problems

A MOP can be described as:

$$\arg\ \min_x F(x) = \{f_1(x), ..., f_n(X)\} \quad x \in \mathcal{X} \tag{2.16}$$

subject to constraints, where $\mathcal{X}$ denotes the decision variable space. $F : \mathcal{X} \to R^n$ consist of n objective functions that conflict with each other. Let $x, y \in \mathcal{X}$, $x \prec y$ represents solution x dominates solution y, iif $f_i(x) \leq f_i(y) \forall i \in \{1, ..., n\}$, and $f_i(x) < f_i(y) \exists i \in \{1, ..., n\}$. A solution $x^\star$ is Pareto optimal if there is no solution $x \in \mathcal{X}$

dominates $x^\star$. The collection of all optimal solutions denoted by $x^\star$ forms the Pareto set, while the array of objective vectors corresponding to these solutions is referred to as the Pareto front [5].

Assessing the effectiveness of algorithms in a MOP can be quite challenging. Because MOPs are not like single-objective problems that can be judged according to one indicator, MOPs need to consider a whole set of solutions, which means is essential to find appropriate performance metrics. Many studies proposed metrics to overcome this issue, from which *hypervolume* [35] and *coverage* [34] are used in this paper.

Chapter 3

Experiment

3.1 The hybrid multi-objective evolutionary algorithm

In this section, we introduce our hybrid multi-objective evolutionary algorithm (HMOEA) as a solution strategy for addressing the MOVRPTW. The algorithm we propose is founded on the framework of Evolutionary Algorithms (EAs), which belong to a category of optimization methods inspired by the principles of natural evolution and genetics. These EAs establish an analogy between biological evolution and the optimization process, employing concepts such as selection, reproduction, mutation, and recombination to guide the evolution of a population of potential solutions toward improved outcomes across successive generations.

Our proposed methodology addresses the MOVRPTW challenge by seamlessly integrating Multi-Objective Evolutionary Algorithms (MOEAs) with an innovative similarity measurement mechanism and a local search approach. This integration is poised to optimize the delicate trade-offs among competing objectives, ultimately aiming to augment the efficiency of route planning while maintaining adherence to stringent vehicle capacity and time windows constraints. The sequential workflow of the proposed algorithm is visually depicted in Figure 3.1, while the precise operational details are outlined in the algorithmic representation provided in Algorithm 1.

The algorithm commences its journey with the initiation and evaluation of a population, embarking on a cyclic loop that continues until a designated number of generations is reached. Operating within this loop, the algorithm harnesses a hybrid methodology that combines selection, crossover, mutation, local search, and similarity measurement. This orchestration iteratively evolves a diverse population of prospective solutions tailored to the intricate contours of the MOVRPTW challenge.

The forthcoming subsections will delve into comprehensive elucidations of these intricacies, furnishing a thorough understanding of the algorithm's inner workings and its efficacy in solving the MOVRPTW challenge.

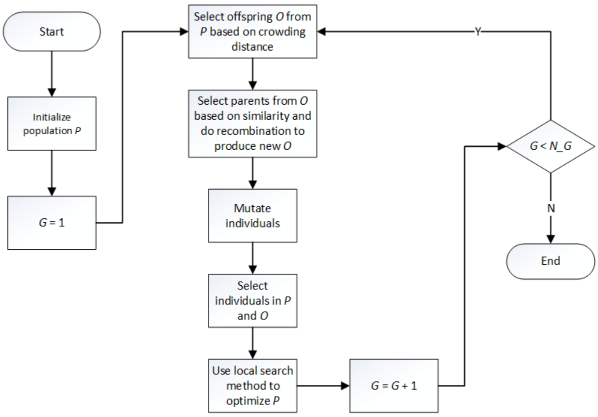
Start
Initialize population *P*
G = 1
Select offspring *O* from *P* based on crowding distance
Select parents from *O* based on similarity and do recombination to produce new *O*
Mutate individuals
Select individuals in *P* and *O*
Use local search method to optimize *P*
G = *G* + 1
G < *N_G*
Y
N
End

Figure 3.1: The process of the HMOEA.

3.1.1 Initial population

Population initialization marks the initial stage of EAs. The population is constituted by a collection of individuals, representing potential solutions to the problem at hand. Each individual comprises a randomly generated sequence of visited customers along a route. This can be visualized in Figure 3.2, where the length of the individual aligns with the number of customers present on the route.

2	6	1	4	7	9	5	3	8

Figure 3.2: Individual representation.

According to the capacity of the vehicle, we start from the depot, and from each added customer's demand, when the load exceeds the capacity, this sub-route is finished. We repeat until we reach the end of the sequence of customers, then the sequence divides into several sub-sequence, as shown in Figure 3.3. The count of sub-routes corresponds to the number of vehicles.

Algorithm 1: Hybrid multi-objective evolutionary algorithm

```
Input: Population size $p$, Crossover rate $\epsilon$, Number of generation $N_G$
$population \leftarrow$ initialize($p$);
$fitness \leftarrow$ evaluate($population$);
$G \leftarrow 0$;
while $G < N_G$ do
    $offspring \leftarrow$ selection($population$)
    for $ind_1 \in offspring$ do
        $simi \leftarrow 1$;
        for $i \in$ random.sample(offspring, 5) do
            temp $\leftarrow$ similarityMeasurement($ind_1, i$);
            if $temp < simi$ then
                simi = temp;
                $ind_2 \leftarrow i$;
            end
        end
        if $random < \epsilon$ then
            crossover($ind_1, ind_2$);
        end
        mutate($ind_1$);
        mutate($ind_2$);
    end
    2-Opt($population$);
    $population \leftarrow$ selection($population + offspring$);
    $G \leftarrow G + 1$;
end
return $population$
```

3.1.2 Fitness evaluation

It's crucial in EA to evaluate every individual's fitness in each generation. Unlike single-objective problems, the fitness function could be straightforwardly calculated by one objective like total distance, in multi-objective problems, we first need to compute each objective including the number of vehicles m, the entire distance $\sum_{i=1}^{m} D_j$, the longest travel time among all routes $\max\{T_j \mid j = 1, ..., m\}$, the total waiting time $\sum_{i=1}^{m} W_j$, and the total delay time $\sum_{i=1}^{m} Delay_j$ and store them in a tuple. Non-dominance sorting criterion of Deb et al. [8] proposed an approach to make a trade-off between all the objectives. This is the chosen selection approach employed by this algorithm. By leveraging this criterion, MOEAs can effectively balance convergence and diversity, explore the Pareto Front more comprehensively, and produce high-quality

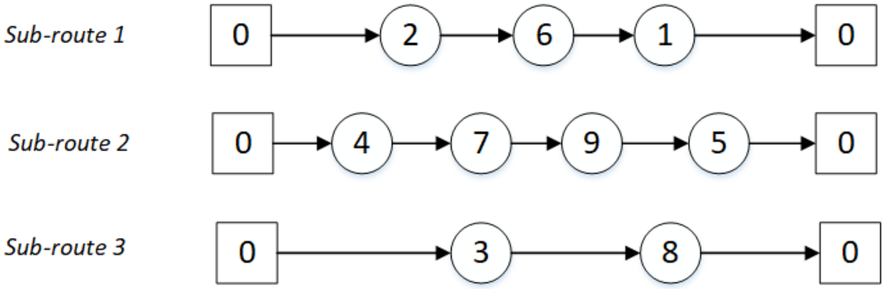

Figure 3.3: Route representation.

solutions that cater to a wide range of objectives and preferences.

3.1.3 Similarity Measurement

Maintaining diversity within the population is a critical aspect of evolutionary algorithms. Success hinges on preventing premature convergence by striking a balance between exploration and exploitation [10], these concepts play crucial roles in finding optimal solutions efficiently and effectively in complex search spaces. And for MOPs, population diversity is of utmost importance. Ensuring that the final population encompasses the entire Pareto front, rather than just a portion of it, is a crucial goal [13].

To tackle this challenge, Garcia-Najera et al. [13] introduced a similarity measurement mechanism aimed at preserving population diversity. Their approach has been demonstrated to exhibit favorable performance outcomes. The concept is rooted in Jaccard's similarity coefficient, which quantifies the proportion of common elements between the sets relative to the total number of distinct elements present in both sets. [12]. The similarity between set A and set B is

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}. \tag{3.1}$$

To apply this measure in the context of VRPTW, each individual is regarded as a set of edges $(c(i,j), c(i+1,j))$. As a result, the similarity between solution X and solution Y is determined by the proportion of common edges to the total number of edges in the two solutions.

$$S_{XY} = \frac{\sum_{i,j\in\mathcal{V}}(i,j)_X \cdot (i,j)_Y}{\sum_{i,j\in\mathcal{V}} \operatorname{sign}((i,j)_X + (i,j)_Y)}, \tag{3.2}$$

where $(i,j)_X$ denote the edge (i,j) of solution X, $(i,j)_X = 1$ means the arc exists, otherwise $(i,j)_X = 0$, (i,j) and (j,i) consider as two different edges. The *sign* function returns 1 if the sum is positive, -1 if the sum is negative, and 0 if the sum is zero. Thus if X and Y have every edges in common, $S_{XY} = 1$, while if X and Y have nothing in common, $S_{XY} = 0$.

3.1.4 Parents selection

Some traditional methods in EA to select parents are tournament selection, keep best selection, and proportional roulette selection. Similarity measurement is used to select parents for every individual in the population. For each individual, we randomly sample five individuals to do similarity matching and chose the least similar individual as its mating parent. Doing so can enable each individual to mate with the one that is least similar to them, resulting in a greater diversity of offspring, thereby making the entire population more heterogeneous.

3.1.5 Recombination

Recombination is the process in evolutionary algorithms whereby offspring are generated through the crossover of parent individuals, with a certain recombination probability ϵ of occurrence. For the recombination method, we used ordered crossover (OX) [14] since this method could always provide valid and feasible outcomes. This could help us calculate the fitness of the individual without checking if is valid or not.

The OX (Order Crossover) process is shown in Figure 3.4 involves initially having two parents. Two crossover points a, b are randomly selected to divide them into three parts: left, middle, and right. $Offspring2$ inherits the middle part [2,4,9] from $parent1$, while the left and right parts are inherited from $parent2$. This is accomplished by removing [2, 4, 9] from $parent2$ and filling in the remaining numbers, starting from location $b+1$. The same operation is performed on $offspring1$.

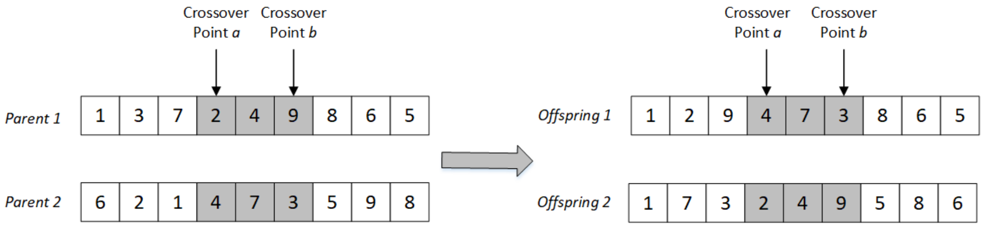

Figure 3.4: Illustration of OX operator.

3.1.6 Mutation

Mutation is another method used to increase the diversity of a population after offspring generation, it occurs with a certain mutation probability σ. And in this paper, we employed the swap method. Specifically, the operation involves exchanging two random customers, the example is shown in Figure 3.5.

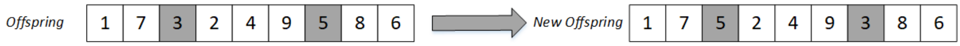

Figure 3.5: Illustration of swap mutation operator.

3.1.7 Selection

The final step of each evolution round is the selection process, where we aggregate the populations of both parents and offspring to select p individuals using the non-dominated sorting approach and the crowding distance comparison from NSGA-II [8]. The concept of non-dominated sorting organizes individuals into distinct fronts according to their dominance relationships. The first front comprises individuals that aren't dominated by any other members of the population. The second front encompasses individuals dominated by the first front, yet they themselves do not dominate any other individuals. The sorting continues until all individuals are assigned. And crowding distance is calculated within each front and represents how crowded or sparse the solutions are in the objective space. During selection, solutions with greater crowding distances are favored, as they indicate less congested regions and offer a more accurate representation of the Pareto front [8].

3.1.8 Local search method

An approach to integrate the local search algorithm with the evolutionary algorithm involves applying it to individuals within the population subsequent to the processes of recombination and mutation [20]. Regarding the local search methods, we experimented with 2-opt, 3-opt, relocate, and Tabu search. Among them, the 3-opt, relocate, and Tabu search demonstrated suboptimal performance. Therefore, we ultimately selected the 2-opt method.

The original 2-opt algorithm [6] would compare every possible solution to find the best, which is time-consuming. In this thesis, we employed a modified version of 2-opt, where a random individual from the final evolved population was selected. The specific procedure involved randomly choosing two customers within two sub-routs and performing an exchange. This exchange process was iterated five times to identify the individual with the highest fitness. The process is depicted in Figure 3.6.

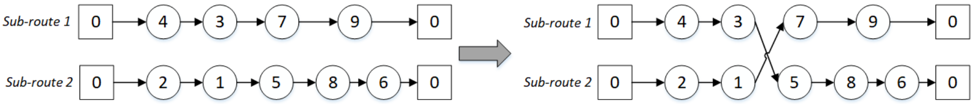

Figure 3.6: Illustration of 2-opt local search method.

3.2 Experiments design

This section describes the experimental strategy used to assess the effectiveness of the algorithm. It covers aspects such as the utilized datasets, performance evaluation metrics, and the established baseline.

The Python implementation of the HMOEA was executed on a personal computer. The system utilized for conducting all experiments consisted of a 2.30 GHz Intel Core i5 processor and 16 GB of RAM. The experiments are tested on two data sets, the first one is the VRPTW benchmark instance called Solomon's instance [25], and the other one is the real-world routing data set provided by Amazon [21]. The experimental configurations encompassed specific parameter settings: the number of generation N_G is 100, the Population size p is 100, the mutation rate σ is 0.1, and the crossover rate ϵ is 0.85. Subsequent to a series of experiments, it was determined that these parameter values yielded improved outcomes.

The baseline and the HMOEA have each been run ten times on each instance independently in order to produce accurate data.

3.2.1 Data sets

The conducted experiments were applied to two distinct datasets: the initial dataset being Solomon's instance, a recognized benchmark for the VRPTW [25], and the second dataset being a real-world routing dataset furnished by Amazon [21].

1. Solomon's benchmark instances

   Solomon's standard public benchmark set [25], which includes 56 instances of size N = 100 available from Solomon's website[1]. They are well-established and

[1] http://w.cba.neu.edu/ msolomon/problems.htm

widely recognized in the research community. They have been extensively studied, which allows for direct comparisons between different algorithms and methods across a common set of problem instances. These instances have been classified into six distinct categories: C1, C2, R1, R2, RC1, and RC2. The 'C' class pertains to problems with clustered data, implying that customers are grouped either based on geographic proximity or time windows. The 'R' class represents problems with uniformly random data distribution. Finally, the 'RC' class encompasses instances that combine characteristics from both the 'C' and 'R' classes. Additionally, Category 1 has a vehicle with a small capacity that only serves a small number of clients, whereas Category 2 has a vehicle with a big capacity that serves more consumers in a single sub-route.

2. Amazon Data Set

   2021 Amazon Last Mile Routing Research Challenge: Data Set [21]. The Challenge provides two datasets. The training data set has 6,112 routes, and the evaluation set has 3,052 routes. Each route contains a set of 50 to 250 stops. The dataset provided by Amazon consists exclusively of real-life routes, offering a more accurate representation of actual operational scenarios.

   Since this data set is designed for the Travel Salesman Problem instead of the VRPTW problem, we need to apply preprocessing steps. The data set contained route information, stop information and package information. In route information, each route has the capacity of the vehicle, departure time, and each stop in the route. Stop information specified the coordinates of each stop, packages delivered at each stop, and the estimated transit time to every other stop on the route. As for the package information, it has the service time, dimension, and time window for each package. Please note the time window is not applicable for every package, since some customers do not mention it. For packages that do not specify a time window, we set his time window to a whole day. A route represents an individual instance, where each stop is considered as a customer to be serviced. The sum of package sizes to be delivered at each stop is treated as the customer's load, while the earliest and latest time windows within the stop's package collection define the time window. Considering that the capacity of the vehicle in the provided dataset accommodates all packages within each instance, and as only one vehicle operates on a given route, we opted to diminish the vehicle's capacity from 3,313,071 to 1,313,071 and from 4,247,527 to 2,247,527. The distances and estimated times between each stop are assumed to be equal, as in reality, distance and time are positively correlated, and calculating precise distances in the real world is challenging. As Amazon did not provide the location of the depot, the required time between the depot and each stop is unknown. In this study, we consider the first stop of each route as the depot.

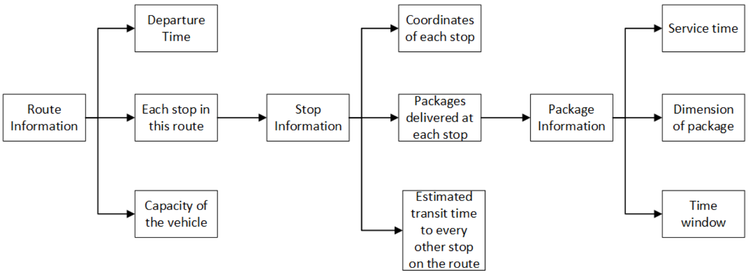

Figure 3.7: Structure of Amazon data set.

Categorized by the count of customers (stops), we organized the problems into three distinct groups: A1 encompassing instances with fewer than 100 customers, A2 comprising instances with 100 to 200 customers, and A3 involving instances with over 200 customers. Within each problem category, a selection of 15 instances was randomly drawn from the training dataset, and each route has an 80% probability of being selected. Each of these instances includes the corresponding route ID present in the original dataset. We did not run all instances due to the primary purpose of the complete training set being model development. With 45 examples, we deemed it sufficient to assess the testing algorithm's performance.

Table 3.1: Details of A1 and vehicle capacity in cm^3 for instances.

Instance	Number of customer	Vehicle capacity
A100	59	1313071
A101	78	1313071
A102	80	1313071
A103	78	1313071
A104	91	1313071
A105	72	2247527
A106	96	1313071
A107	83	1313071
A108	74	1313071
A109	82	2247527
A110	82	1313071
A111	38	2247527
A112	85	1313071
A113	82	1313071
A114	69	1313071

Table 3.2: Details of A2 and vehicle capacity in cm^3 for instances.

Instance	Number of customer	Vehicle capacity
A200	119	1313071
A201	106	2247527
A202	128	2247527
A203	142	1313071
A204	155	1313071
A205	190	2247527
A206	161	1313071
A207	155	2247527
A208	177	1313071
A209	129	1313071
A210	185	1313071
A211	173	1313071
A212	161	1313071
A213	130	1313071
A214	176	1313071

Table 3.3: Details of A3 and vehicle capacity in cm^3 for instances.

Instance	Number of customer	Vehicle capacity
A300	202	1313071
A301	222	2247527
A302	205	2247527
A303	204	2247527
A304	203	1313071
A305	203	1313071
A306	203	1313071
A307	204	1313071
A308	203	1313071
A309	203	1313071
A310	206	2247527
A311	204	2247527
A312	203	1313071
A313	204	1313071
A314	219	1313071

3.2.2 Performance metrics

Assessing the performance of algorithms in Multiple-Objective Problems (MOPs) involves evaluating both convergence and diversity aspects. As concluded by Zitzler et al [37], relying on a single performance metric isn't sufficient to provide a holistic measure of an MOP algorithm's performance. Therefore, it is advisable to utilize multiple indicators. In this study, we employed widely-used metrics in MOPs such as the Hypervolume (HV) and the Coverage metric (C-metric).

1. Hypervolume (HV): HV metric is proposed by Zitzler and Thiele [35]. The hypervolume metric calculates the total amount of space enclosed by a set of solutions, which indicates how well the solutions cover the desired objective space. A higher hypervolume value indicates a more diverse and better-distributed set of solutions, which is desirable in multi-objective optimization. To calculate HV, we need to define the reference point, which is a set of points with extreme values in each objective dimension. In this paper, we define as [1.01, 1.01, 1.01, 1.01, 1.01].

2. Coverage metric (C-metric): C-metric is proposed by Zitzler and Thiele [36]. This metric is extensively employed for comparing two sets of nondominated solutions, labeled as X and Y. The value of $C(X, Y)$ reflects the proportion of solutions in set Y that are dominated by at least one solution in set X. A value

of 1 for $C(X,Y)$ implies that all nondominated solutions in Y are dominated by solutions in X, while a value of 0 indicates that all nondominated solutions in X are dominated by solutions in Y. It's important to note that the sum of $C(X,Y)$ and $C(Y,X)$ doesn't always equate to 1, owing to the potential scenario where some solutions within sets X and Y do not dominate each other.

A collection of solutions with a higher HV can be interpreted as a superior approximation to the true Pareto Front. Given the varying ranges of objectives in each instance, we have normalized all objective values to compute HV and C-metric.

3.2.3 Baseline

To evaluate the performance of our algorithm, we used NSGA-II [8] as the baseline. It is a state-of-art multi-objective evolutionary algorithm that has proven to have good performance and offers different choices and trade-offs for decision-makers facing complex problems with multiple objectives. The NSGA-II implementation retained an identical solution representation and population initialization, along with consistent utilization of crossover, mutation, selection operators, and parameter configuration as employed in the hybrid multi-objective evolutionary algorithm. The disparities lie in the parent selection procedure and the subsequent integration of the local search approach following the evolution process.

Chapter 4

Results

We compared the results of the hybrid multi-objective evolutionary algorithm with NSGA-II by applying both algorithms to two data sets: Solomon's benchmark instances and the Amazon data set. Due to the stochastic nature of the experiments, all the results are reported as the average (AVG) and standard deviation (SD) obtained from ten repetitions of the experiment. The difference between the results obtained by the two algorithms was demonstrated through the utilization of the Wilcoxon signed-rank test [31, 9, 1] at 5% significance level.

4.1 Solomon's benchmark instances

Table 4.1 illustrates the results between HV and C-metric in the context of Solomon's instances. The initial column lists the instance names, followed by the second and third columns displaying the mean and standard deviation of HV, respectively. The fourth column showcases the mean of C-metric, delineated by C(NS, HM) for C(NSGA-II, HMOEA) and C(HM, NS) for C(HMOEA, NSGA-II). For each instance, superior values are highlighted in bold typeface. The concluding row within Table W/B provides an overview, indicating whether HMOEA is better or worse compared to NSGA-II.

The comparison of HV and C-metric results in Solomon's instances is presented in Table 4.1, where the first column is the name of instances, the second and third columns are the average and standard deviation of HV respectively, the fourth column is the average of C-metric, in which C(NS, HM) is C(NSGA-II, HMOEA), C(HM, NS) is the C(HMOEA, NSGA-II). The better values are represented in bold font for each instance.

Based on the results, we can observe that HMOEA outperforms NSGA-II overall. In terms of the average values, HMOEA demonstrates superiority over NSGA-II in the majority of cases, achieving higher Hypervolume (HV) scores. Specifically, out of the total instances, HMOEA outperforms NSGA-II in 47 instances concerning HV, while in 9 instances, it shows slightly worse performance.

An additional perspective to consider is the stability of the algorithms, as indicated by the standard deviation values. Notably, HMOEA showcases a slightly higher level of stability compared to NSGA-II, suggesting that it could yield more consistent results across various scenarios.

Furthermore, evaluating the algorithms using the C-metric reveals a similar pattern. HMOEA outperforms NSGA-II in 45 instances, showcasing its superior convergence and diversity capabilities. In contrast, there are 11 instances where HMOEA falls short compared to NSGA-II. This collectively underscores HMOEA's proficiency in achieving a balance between convergence and diversity, ultimately leading to enhanced overall performance.

4.2 Amazon Data Set

In Table 4.2, we delve into a comprehensive comparison between the outcomes of HV and C-metric analyses within the Amazon dataset. Much like Table 4.1, the table structure's initial columns remain consistent, including instance names, average HV values, standard deviations of HV, and average C-metric values. The difference is the final row of the table furnishes a comprehensive overview through a W/S/B format, embodying HMOEA Worse than, Similar to, and Better than comparisons, providing an encompassing vantage point to assess the performance of HMOEA in relation to NSGA-II.

Based on the results, a discernible pattern emerges—HMOEA exhibits a tendency to outperform NSGA-II across diverse metrics. This observation holds true when considering the average values, as HMOEA registers remarkable improvements over NSGA-II in 28 instances under the HV metric. Remarkably, 7 instances demonstrate inferior performance. A parallel trend is evident when scrutinizing the standard deviation of HV, with HMOEA showcasing superiority in 35 instances compared to NSGA-II. However, it is crucial to acknowledge the balance, as 10 instances exhibited less favorable outcomes under HMOEA's approach.

Furthermore, evaluating the performance through the lens of C-metric reaffirms HMOEA. It triumphs over NSGA-II in 31 instances, reinforcing its potential for enhancing optimization. Nonetheless, it is essential to acknowledge that in 13 instances, HMOEA registers comparatively unfavorable results, while 1 instance exhibits similarity. This comprehensive examination underscores HMOEA's overall effectiveness in addressing the challenges posed by the Amazon dataset and highlights its potential to outperform traditional methods such as NSGA-II across a range of performance metrics.

Table 4.1: The AVG and SD of the hypervolume (HV) and Coverage metric (C-metric) results for the NSGA-II and HMOEA algorithms in Solomon's instances experiment, superior values are highlighted in bold typeface.

Instance	HV-AVG		HV-SD		C-metric	
	NSGA-II	HMOEA	NSGA-II	HMOEA	C(NS,HM)[a]	C(HM,NS)[b]
C101	0.87958	**0.90172**	**0.06982**	0.07237	0.30600	**0.49800**
C102	0.86834	**0.90226**	0.11097	**0.06584**	0.21500	**0.42200**
C103	0.63655	**0.66850**	**0.27252**	0.36602	0.27575	**0.57812**
C104	0.54264	**0.65106**	**0.24458**	0.31427	0.23442	**0.34109**
C105	0.82921	**0.93097**	0.17592	**0.04990**	0.24345	**0.41100**
C106	0.77889	**0.86833**	0.24850	**0.16987**	0.34646	**0.70519**
C107	**0.77462**	0.52132	**0.17227**	0.31481	**0.41900**	0.28700
C108	0.60628	**0.67702**	0.30326	**0.26934**	**0.15100**	0.05900
C109	0.53529	**0.71507**	0.30204	**0.20037**	**0.29650**	0.04200
C201	0.88378	**0.88662**	0.13022	**0.08078**	0.32364	**0.45791**
C202	**0.92057**	0.90441	**0.06328**	0.06890	**0.41655**	0.40800
C203	0.89519	**0.89825**	0.12161	**0.09327**	0.33631	**0.58341**
C204	0.86843	**0.89738**	0.18798	**0.11490**	0.37030	**0.38473**
C205	0.86557	**0.89096**	0.08176	**0.05793**	0.38366	**0.54000**
C206	0.91744	**0.92185**	**0.04235**	0.04800	0.42997	**0.44367**
C207	0.84325	**0.87873**	0.11196	**0.09296**	0.36836	**0.47100**
C208	0.84891	**0.86623**	**0.05422**	0.07679	0.39411	**0.63917**
R101	0.78533	**0.90256**	0.15136	**0.04844**	0.31958	**0.56300**
R102	0.62589	**0.67018**	**0.19728**	0.30173	0.34167	**0.36722**
R103	0.39709	**0.53129**	**0.33428**	0.36308	0.22825	**0.46106**
R104	0.51277	**0.83305**	0.44549	**0.29046**	0.39530	**0.70700**
R105	**0.84239**	0.82943	0.10114	**0.09326**	0.29358	**0.42864**
R106	0.52118	**0.64363**	**0.16323**	0.28768	0.32979	**0.64959**
R107	0.41975	**0.72374**	0.38092	**0.37601**	0.15567	**0.73800**
R108	0.47103	**0.53545**	**0.38491**	0.42521	0.43556	**0.65039**
R109	**0.70372**	0.68592	0.25304	**0.20160**	**0.42696**	0.17218
R110	0.51957	**0.73568**	0.36283	**0.31908**	0.28062	**0.28132**
R111	0.46222	**0.68079**	**0.27516**	0.30121	0.23335	**0.61608**
R112	0.58190	**0.61109**	**0.22639**	0.32315	0.46506	**0.52098**
R201	0.80888	**0.83020**	**0.07382**	0.08384	0.34849	**0.50223**
R202	0.85166	**0.88930**	0.19246	**0.10067**	0.37687	**0.39458**
R203	0.85305	**0.85508**	**0.09041**	0.12225	0.36263	**0.50939**
R204	0.83364	**0.93550**	**0.14666**	0.17240	0.37713	**0.57346**
R205	0.81602	**0.85562**	**0.06481**	0.06519	0.32182	**0.34900**
R206	0.78864	**0.87685**	0.11284	**0.07384**	0.20984	**0.57700**
R207	0.86209	**0.91063**	0.09317	**0.06980**	**0.38032**	0.20441
R208	0.82041	**0.94091**	0.21951	**0.14514**	0.26052	**0.76657**
R209	0.74908	**0.87182**	0.17510	**0.11799**	0.44754	**0.62111**
R210	0.77128	**0.87682**	0.11594	**0.11397**	0.31399	**0.51159**
R211	**0.80845**	0.77864	0.13500	**0.12151**	**0.45687**	0.43407
RC101	0.71507	**0.75804**	**0.12841**	0.17219	0.24570	**0.49539**
RC102	**0.72006**	0.69056	0.27107	**0.23676**	0.35919	**0.52578**
RC103	**0.57744**	0.55803	**0.20177**	0.27974	0.30477	**0.46033**
RC104	0.33824	**0.60497**	**0.29957**	0.32761	**0.41993**	0.38564
RC105	0.72960	**0.76846**	**0.14213**	0.14267	0.30672	**0.42895**
RC106	0.61252	**0.73261**	**0.29394**	0.29835	0.35920	**0.36284**
RC107	0.32382	**0.66470**	**0.21939**	0.28484	0.21545	**0.41084**
RC108	0.53267	**0.56231**	0.37447	**0.32953**	**0.40156**	0.17900
RC201	0.83554	**0.87407**	**0.04711**	0.08122	0.35383	**0.56731**
RC202	0.82720	**0.87507**	0.09412	**0.08599**	0.28490	**0.43903**
RC203	**0.90734**	0.87613	**0.07533**	0.10475	0.37706	**0.49389**
RC204	0.90420	**0.91358**	0.11442	**0.10357**	0.36725	**0.42433**
RC205	0.85685	**0.86617**	0.09441	**0.08844**	**0.50537**	0.43789
RC206	0.84195	**0.86147**	0.08616	**0.06663**	0.34683	**0.44317**
RC207	**0.85281**	0.82646	**0.07680**	0.10967	**0.51361**	0.34412
RC208	0.78807	**0.86862**	0.25738	**0.08813**	0.25799	**0.51741**
W/B	9/47		26/30		11/45	

[a] C(NSGA-II, HMOEA)
[b] C(HMOEA, NSGA-II)

Table 4.2: The AVG and SD of the hypervolume (HV) and Coverage metric (C-metric) results for the NSGA-II and HMOEA algorithms in the Amazon data set experiment, superior values are highlighted in bold typeface.

Instance	HV-AVG		HV-SD		C-metric	
	NSGA-II	HMOEA	NSGA-II	HMOEA	C(NS,HM)[a]	C(HM,NS)[b]
A100	0.61629	**0.77360**	0.18942	**0.13732**	0.15375	**0.56003**
A101	0.61881	**0.76852**	0.43729	**0.29651**	0.58100	**0.67778**
A102	0.58329	**0.69294**	**0.25277**	0.27965	**0.39705**	0.32404
A103	**0.75308**	0.71688	**0.19567**	0.31870	0.22133	**0.69581**
A104	0.34472	**0.75924**	**0.33208**	0.34784	0.36268	**0.89800**
A105	0.50596	**0.89580**	0.41244	**0.26994**	0.12778	**0.74575**
A106	0.67709	**0.71784**	0.29332	**0.27759**	0.49495	**0.68019**
A107	0.60565	**0.71507**	0.30774	**0.28945**	0.43423	**0.58625**
A108	0.56203	**0.79259**	0.31258	**0.17518**	0.29498	**0.76207**
A109	0.45747	**0.85067**	0.47158	**0.16125**	0.35000	**0.51133**
A110	0.72675	**0.74702**	**0.25786**	0.27261	0.29800	**0.64553**
A111	0.65709	**0.88109**	0.40161	**0.23465**	0.40400	**0.54107**
A112	0.68732	**0.89933**	0.29869	**0.14262**	0.29491	**0.59872**
A113	**0.79636**	0.68320	0.42744	**0.42427**	**0.74700**	0.40110
A114	0.69161	**0.77379**	0.32452	**0.17715**	**0.51890**	0.37838
A200	0.70759	**0.91315**	0.36133	**0.15607**	0.18727	**0.66657**
A201	0.46221	**0.74321**	0.44636	**0.41121**	0.30300	**0.69600**
A202	0.54070	**0.75209**	**0.37239**	0.40471	0.25020	**0.66395**
A203	0.87150	**0.92448**	0.14442	**0.13946**	0.39798	**0.62807**
A204	**0.90681**	0.83556	**0.30542**	0.34668	**0.72675**	0.44737
A205	0.72025	**0.80092**	0.45308	**0.42765**	**0.77368**	0.71229
A206	0.43173	**0.81716**	0.48779	**0.37201**	0.39800	**0.80000**
A207	0.74902	**0.88667**	0.38326	**0.31346**	0.42000	**0.67750**
A208	0.87923	**0.97376**	0.33440	**0.24296**	0.70000	**0.80000**
A209	0.63986	**0.92724**	0.47420	**0.32153**	0.60000	**0.90000**
A210	0.67488	**0.68435**	0.47519	**0.44840**	**0.70000**	0.50200
A211	0.51573	**0.80215**	0.41645	**0.18574**	0.49098	**0.73000**
A212	0.73440	**0.86661**	0.49304	**0.34182**	**0.70000**	0.60000
A213	0.58580	**0.70402**	0.49956	**0.43913**	0.66267	**0.69400**
A214	0.78701	**0.80919**	**0.41884**	0.42021	0.60000	**0.69778**
A300	0.85814	**0.90646**	0.21686	**0.17860**	0.50427	**0.59836**
A301	**0.95689**	0.87669	**0.29762**	0.36065	**0.90000**	0.70000
A302	0.64478	**0.89499**	0.43879	**0.29552**	0.30300	**0.82208**
A303	0.68346	**0.73785**	0.41172	**0.37673**	**0.51667**	0.49600
A304	**0.89804**	0.73631	**0.34524**	0.48741	**0.89800**	0.60000
A305	**0.88220**	0.86925	0.36655	**0.33777**	**0.80000**	0.60000
A306	**0.70107**	0.67705	0.48248	**0.38612**	**0.77818**	0.40909
A307	0.67429	**0.74108**	0.46544	**0.43158**	**0.50204**	0.49750
A308	0.65628	**0.81657**	**0.35765**	0.42364	0.39593	**0.75577**
A309	0.66648	**0.76833**	0.40148	**0.37126**	0.55219	**0.67829**
A310	0.78847	**0.94798**	0.44131	**0.32581**	0.70000	**0.90000**
A311	0.64871	**0.97681**	0.47417	**0.17164**	0.50000	**0.73750**
A312	0.72445	**0.78651**	0.48115	**0.41237**	0.70000	0.70000
A313	0.63889	**0.95480**	0.53205	**0.29538**	0.60400	**0.80200**
A314	0.68368	**0.93550**	0.43706	**0.12584**	0.47778	**0.71380**
W/B/S	7/28/0		10/35/0		13/31/1	

[a] C(NSGA-II, HMOEA)
[b] C(HMOEA, NSGA-II)

Chapter 5

Conclusions and Future Work

5.1 Summary

This paper has presented a hybrid multi-objective evolutionary algorithm to address MOVRPTW. The algorithm follows the framework of the evolutionary algorithm with similarity measurement, local search method, and non-dominated sorting selection operator. Experimental evaluations were conducted on two datasets, Solomon's benchmark instance and 2021 Amazon Last Mile Routing Research Challenge: Data Set, revealing superior performance compared to conventional NSGA-II. The dataset provided by Amazon consists of real-world data, and our proposed method exhibits promising performance, particularly in scenarios with a large number of customers. This suggests the applicability of our algorithm to real-world settings, addressing a range of problems.

Moreover, effectively solving VRPTW can have the following impacts on society, e.g., improved transportation efficiency, energy savings, enhanced service quality, and promoted economic development. Optimizing vehicle routing can reduce the distance and travel time, leading to decreased fuel emissions. By properly arranging the vehicle routes and time windows, it is possible to have logistics services reach customers on time and meet their time window requirements. This will improve service quality and customer satisfaction. Furthermore, solving VRPTW can help businesses reduce logistic costs and strengthen supply chain coordination and management.

5.2 Future work

This paper holds potential for expansion from two distinctive perspectives. Firstly, the current approach initializes the population in a randomized manner and hasn't explored various recombination and mutation techniques. Addressing this could lead to an intriguing avenue of research. For instance, future investigations might involve devising heuristics tailored to each example during the population's initial establishment, potentially serving as a foundation for subsequent evolution. Exploring different recombination and mutation operators within the evolutionary algorithm could also yield enhanced performance.

Moreover, integrating the methodology outlined in this paper with artificial intelligence holds promise. Utilizing neural networks for individual optimization post each evolutionary round could be a fruitful endeavor. However, this would introduce challenges such as adapting neural network architectures, training protocols, and balancing computational efficiency. The resulting amalgamation could potentially bring forth improved optimization outcomes and unveil new insights into the intersection of evolutionary algorithms and artificial intelligence.

Bibliography

[1] Jesús Alcalá-Fdez, Luciano Sanchez, Salvador Garcia, Maria Jose del Jesus, Sebastian Ventura, Josep Maria Garrell, José Otero, Cristóbal Romero, Jaume Bacardit, Victor M Rivas, et al. Keel: a software tool to assess evolutionary algorithms for data mining problems. *Soft Computing*, 13:307–318, 2009.

[2] Raúl Baños, Julio Ortega, Consolación Gil, Antonio L Márquez, and Francisco De Toro. A hybrid meta-heuristic for multi-objective vehicle routing problems with time windows. *Computers & industrial engineering*, 65(2):286–296, 2013.

[3] Humberto César Brandão de Oliveira and Germano Crispim Vasconcelos. A hybrid search method for the vehicle routing problem with time windows. *Annals of Operations Research*, 180:125–144, 2010.

[4] Juan Castro-Gutierrez, Dario Landa-Silva, and José Moreno Pérez. Nature of real-world multi-objective vehicle routing with evolutionary algorithms. In *2011 IEEE International Conference on Systems, Man, and Cybernetics*, pages 257–264. IEEE, 2011.

[5] Shao-Wen Chen and Tsung-Che Chiang. Evolutionary many-objective optimization by mo-nsga-ii with enhanced mating selection. In *2014 IEEE Congress on Evolutionary Computation (CEC)*, pages 1397–1404. IEEE, 2014.

[6] Georges A Croes. A method for solving traveling-salesman problems. *Operations research*, 6(6):791–812, 1958.

[7] George B Dantzig and John H Ramser. The truck dispatching problem. *Management science*, 6(1):80–91, 1959.

[8] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

[9] Joaquín Derrac, Salvador García, Daniel Molina, and Francisco Herrera. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, 1(1):3–18, 2011.

[10] AE Eiben, JE Smith, AE Eiben, and JE Smith. Gray coding. *Introduction to Evolutionary Computing*, pages 265–265, 2003.

[11] Zhuo Fu, Richard Eglese, and Leon YO Li. A unified tabu search algorithm for vehicle routing problems with soft time windows. *Journal of the Operational Research Society*, 59(5):663–673, 2008.

[12] Abel Garcia-Najera and John A Bullinaria. Bi-objective optimization for the vehicle routing problem with time windows: Using route similarity to enhance performance. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 275–289. Springer, 2009.

[13] Abel Garcia-Najera and John A Bullinaria. An improved multi-objective evolutionary algorithm for the vehicle routing problem with time windows. *Computers & Operations Research*, 38(1):287–300, 2011.

[14] David Goldberg and Kumara Sastry. *Genetic algorithms: the design of innovation.* Springer, 2007.

[15] Yue-Jiao Gong, Jun Zhang, Ou Liu, Rui-Zhang Huang, Henry Shu-Hung Chung, and Yu-Hui Shi. Optimizing the vehicle routing problem with time windows: A discrete particle swarm optimization approach. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(2):254–267, 2011.

[16] Zhenan He and Gary G Yen. Comparison of many-objective evolutionary algorithms using performance metrics ensemble. *Advances in Engineering Software*, 76:1–8, 2014.

[17] Arild Hoff, Henrik Andersson, Marielle Christiansen, Geir Hasle, and Arne Løkketangen. Industrial aspects and literature survey: Fleet composition and routing. *Computers & Operations Research*, 37(12):2041–2061, 2010.

[18] Nicolas Jozefowiez, Frédéric Semet, and El-Ghazali Talbi. Parallel and hybrid models for multi-objective optimization: Application to the vehicle routing problem. In *International conference on parallel problem solving from nature*, pages 271–280. Springer, 2002.

[19] Nicolas Jozefowiez, Frédéric Semet, and El-Ghazali Talbi. Multi-objective vehicle routing problems. *European journal of operational research*, 189(2):293–309, 2008.

[20] Mark William Shannon Land. *Evolutionary algorithms with local search for combinatorial optimization.* University of California, San Diego, 1998.

[21] D Merchán, J Pachon, J Arora, K Konduri, M Winkenbach, S Parks, and J Noszek. Amazon last mile routing research challenge: Data set. *Transportation Sci*, 2021.

[22] Juliane Müller. Approximative solutions to the bicriterion vehicle routing problem with time windows. *European Journal of Operational Research*, 202(1):223–231, 2010.

[23] Eric Prescott-Gagnon, Guy Desaulniers, Michael Drexl, and Louis-Martin Rousseau. European driver rules in vehicle routing with time windows. *Transportation science*, 44(4):455–473, 2010.

[24] Malek Rahoual, Boubekeur Kitoun, Mohamed-Hakim Mabed, Vincent Bachelet, and Féthia Benameur. Multicriteria genetic algorithms for the vehicle routing problem with time windows. In *4th Metaheuristics International Conference*, pages 527–532. Citeseer, 2001.

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

[26] Nidamarthi Srinivas and Kalyanmoy Deb. Muiltiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary computation*, 2(3):221–248, 1994.

[27] Gaurav Srivastava, Alok Singh, and Rammohan Mallipeddi. Nsga-ii with objective-specific variation operators for multiobjective vehicle routing problem with time windows. *Expert Systems with Applications*, 176:114779, 2021.

[28] Kay Chen Tan, Yoong Han Chew, and Loo Hay Lee. A hybrid multiobjective evolutionary algorithm for solving vehicle routing problem with time windows. *computational optimization and applications*, 34:115–151, 2006.

[29] Duygu Taş, Nico Dellaert, Tom Van Woensel, and Ton De Kok. Vehicle routing problem with stochastic travel times including soft time windows and service costs. *Computers & Operations Research*, 40(1):214–224, 2013.

[30] Ziauddin Ursani, Daryl Essam, David Cornforth, and Robert Stocker. Localized genetic algorithm for vehicle routing problem with time windows. *Applied Soft Computing*, 11(8):5375–5390, 2011.

[31] Frank Wilcoxon. *Individual comparisons by ranking methods.* Springer, 1992.

[32] Zizhen Zhang, Hu Qin, and Yanzhi Li. Multi-objective optimization for the vehicle routing problem with outsourcing and profit balancing. *IEEE Transactions on Intelligent Transportation Systems*, 21(5):1987–2001, 2019.

[33] Ying Zhou and Jiahai Wang. A local search-based multiobjective optimization algorithm for multiobjective vehicle routing problem with time windows. *IEEE Systems Journal*, 9(3):1100–1113, 2014.

[34] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary computation*, 8(2):173–195, 2000.

[35] Eckart Zitzler and Lothar Thiele. Multiobjective optimization using evolutionary algorithms—a comparative case study. In *International conference on parallel problem solving from nature*, pages 292–301. Springer, 1998.

[36] Eckart Zitzler and Lothar Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE transactions on Evolutionary Computation*, 3(4):257–271, 1999.

[37] Eckart Zitzler, Lothar Thiele, Marco Laumanns, Carlos M Fonseca, and Viviane Grunert Da Fonseca. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on evolutionary computation*, 7(2):117–132, 2003.