

## Article

# Balancing Staff Finishing Times vs. Minimizing Total Travel Distance in Home Healthcare Scheduling

Payakorn Saksuriya <sup>1</sup>  and Chulin Likasiri <sup>2,\*</sup> 

<sup>1</sup> International College of Digital Innovation, Chiang Mai University, Chiang Mai 50200, Thailand; payakorn.s@cmu.ac.th

<sup>2</sup> Department of Mathematics, Faculty of Science, Chiang Mai University, Chiang Mai 50200, Thailand

\* Correspondence: chulin.l@cmu.ac.th

**Abstract:** Cost reduction and staff retention are important optimization objectives in home healthcare (HHC) systems. Home healthcare operators need to balance their objectives by optimizing resource use, service delivery and profits. Minimizing total travel distances to control costs is a common routing problem objective while minimizing total finishing time differences is a scheduling objective whose purpose is to enhance staff satisfaction. To optimize routing and scheduling, we propose mixed integer linear programming with a bi-objective function, which is a subset of the vehicle routing problem with time windows (VRPTWs). VRPTWs is a known NP-hard problem, and optimal solutions are very hard to obtain in practice. Metaheuristics offer an alternative solution to this type of problem. Our metaheuristic uses the simulated annealing algorithm and weighted sum approach to convert the problems to single-objective problems and is equipped with operators including swapping, moving, path exchange and ruin and recreate. The results show, firstly, that the algorithm can effectively find the Pareto front, and secondly, that minimizing total finishing time differences to balance the number of jobs per caretaker is an efficient way to tackle HHC scheduling. A statistical test shows that the algorithm can obtain the Pareto front with a lower number of weighted sum problems.

**Keywords:** vehicle routing problem with time windows; minimizing total distance; minimizing total finishing time differences; home healthcare; bi-objective function



**Citation:** Saksuriya, P.; Likasiri, C. Balancing Staff Finishing Times vs. Minimizing Total Travel Distance in Home Healthcare Scheduling. *Appl. Sci.* **2024**, *14*, 7381. <https://doi.org/10.3390/app14167381>

Academic Editor: Suchao Xie

Received: 18 July 2024

Revised: 17 August 2024

Accepted: 19 August 2024

Published: 21 August 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

This research tackles two objectives in home healthcare (HHC) routing and scheduling, namely minimizing total travel distance and minimizing total finishing time differences. Minimizing total travel distance, a typical objective in a routing problem, attempts to minimize the total cost of the routing, while minimizing total finishing time differences is more meaningful for a general service problem where workload balance is important. The optimal utilization of resources, ensuring efficient service delivery and enhancing profits are all vital objectives for a home healthcare operator and managing all three is tantamount to balancing various objectives. HHC attracts interest due to a variety of factors including aging populations, the rising number of the chronically ill people and a preference for receiving care at home. Currently, those over 65 years of age make up 10% of the global population and this figure is expected to increase to 16% in 2050 [1]. In order for HHC providers to efficiently deliver essential services, they need to optimize the problem, and this requires optimizing both routing and scheduling simultaneously. The difficulties of HHC optimization come from its multi-objective nature, which sets it apart from traditional routing and scheduling problems. Optimizing multiple goals, e.g., minimizing travel time or distance while balancing workload allocation among caretakers, requires customized optimization techniques.

An HHC system is defined as an extensive network of medical experts, caregivers and support services that provides healthcare services to patients in their homes or designated

locations. Home healthcare administrators need to make optimum decisions regarding the route plans for all caretakers. In typical HHC operations, the operator receives information about a group of patients distributed across different locations who need to be visited by one caretaker. Home healthcare managers determine the routes for caregivers, who subsequently travel to provide services to customers following the predetermined routes. Typically, home healthcare managers try to make decisions that minimize total operational costs when creating route plans. To ensure competitiveness in the HHC business, both patient satisfaction and caregiver satisfaction need to be considered as factors that can significantly impact operators' profits. This notion motivated us to study a two-objective home healthcare routing and scheduling problem.

HHC systems can be optimized if healthcare providers treat them as routing and scheduling problems, allowing for the optimization of factors like travel time, distance, vehicle capacity and patient priorities. Optimization can result in reduced travel costs, minimized travel time, enhanced patient satisfaction and improved overall efficiency in healthcare service delivery. However, it is not possible to achieve all these objectives simultaneously, as optimizing one objective requires sacrificing others. In this scenario, using the typical routing objective of total distance minimization alone may not be appropriate due to the time-sensitive nature of both patients' and caretakers' needs. Focusing on distance as the primary goal could lead to unequal distribution of tasks among caregivers. This study aims to introduce a new objective of minimizing the total difference between each pair of caretakers' finishing times, an approach that has never been studied in the literature. The problem will be formulated based on the vehicle routing problem with time windows. Additionally, a metaheuristic approach relying on simulated annealing (SA) is proposed as a solution for handling a large-scale problem. The SA algorithm has been demonstrated to be effective for solving NP-hard problems. The algorithm has the ability to avoid local optima by employing a probabilistic approach.

Since vehicle routing problems are known to be NP-hard, we propose an efficient approach for multiple objectives. In this study, however, we focus only on bi-objective function. In this case, minimizing total distance will result in reducing the overall cost and minimizing total finishing time differences will lead to reduced differences in the number of tasks assigned to each caretaker. The approach starts by converting the problems from multiple objective to single objective and then the problem is solved using a metaheuristic algorithm. This study's main contributions are as follows:

1. The creation of a bi-objective model for home healthcare routing and scheduling, featuring total distance minimization and minimization of the total finishing time difference between each pair of caretakers; and
2. The development of an efficient algorithm for multiple-objective optimization problems based on a metaheuristic (simulated annealing algorithm) together with local search operators (swapping, moving, path exchanging, ruining and recreating) with the pruning conditions (with proof) to prevent unnecessary searching.

The remainder of this paper is organized as follows: The next section presents a review of literature related to various aspects of vehicle routing problems. Section 3 proposes model formulation based on vehicle routing problem with time windows. The solution is presented in Section 4. The results, along with a discussion and analysis, are presented in Section 5. And finally, the conclusion and future research suggestions are presented in Section 6.

## 2. Literature Review

In this section, we will focus on works related to the development of models and objective functions for use in HHC systems and the development of solutions. Since the problem can basically be formulated as a vehicle routing problem, model formulations and solution approach themselves are also reviewed.

Modeling for HHC starts with an allocation of resources (nurses), after which routing is considered. This was first studied by [2], who proposed a model for grouping nurses

into a region by considering the average travel distance. Later, Hindle et al. [3] provided a model for resource allocation that utilized spatial and geographical information to take into account distance and time, with a case study in Northern Ireland. When routing is added to the system, the problem becomes a routing and scheduling problem in home healthcare and can be formulated as a vehicle routing problem. Modeling for HHC resource allocation and routing was first proposed by Eveborn et al. [4]. In contrast to prior works, Bredström and Rönnqvist [5] added temporal, precedence and synchronization constraints (called synchronization and precedence constraints) to the routing and scheduling problem to allow more than one nurse to visit a patient to provide multiple services at the same time. The authors modified the original model by duplicating the node which required more than one visit. Rasmussen et al. [6] expanded on the previous work to feature five different temporal conditions in home healthcare and solved the problem by converting it into a set partitioning problem. Later, Ait Haddadene et al. [7] improved the model with synchronization and precedence constraints to contain a lower number of nodes instead of duplicating the nodes and proposed the greedy randomized adaptive search procedure for large-scale problems. Mankowska et al. [8] proposed modeling to support double services at each patient location, called interdependent services, with each service having temporal-dependent conditions. The authors proposed an adaptive variable neighborhood search with neighborhood search techniques for the temporal constraints. The reader should refer to [9] for a review of the home healthcare routing and scheduling literature and to Mascolo et al. [10] for a bibliometric analysis.

Most home healthcare optimization objectives usually follow the classical objective: the minimization of either total travel distance or time [11]. In previous work, the objective functions for home healthcare routing and scheduling focused on costs related to distance or time, waiting time, over time, etc. Akjiratikar et al. [12] proposed a metaheuristic based on particle swarm optimization with embedded local improvement procedures to solve home healthcare routing and scheduling problems by minimizing total distance when capacity and time windows need to be satisfied. Dohn et al. [13] maximized the total number of assigned tasks for a manpower allocation problem, an approach called node covering. The study focused on the cooperation of teams to perform the same task under the conditions of time and working hours, and the problem was solved using the Danzig–Wolfe decomposition [14]. Allaoua et al. [15] used the Danzig–Wolfe decomposition technique and branch and price algorithm to solve a home healthcare problem formulated using a combination of routing and rostering problems. Their objective was to minimize the total number of caretakers. Saksuriya and Likasiri [16] minimized the total completion time for each patient node in a home healthcare problem formulated as a vehicle routing problem with time windows. To make the model more suitable for HHC systems, the chosen objective functions were the sum of total distance or cost and other aspects related to HHC. Trautsamwieser et al. [17] proposed a model with time windows for working time and break time to minimize the sum of traveling times, waiting times and the dissatisfaction levels of clients and nurses. Yuan et al. [18] proposed home healthcare routing and scheduling with time windows, stochastic service time and hierarchical skill levels. The objective function, to minimize total travel cost and penalty cost for violating the time windows, was solved using the Danzig–Wolfe decomposition and the branch and price algorithm. de Aguiar et al. [19] proposed a model to minimize the sum of total travel time and service time, working time, and penalty cost. The model was based on a vehicle routing problem with time windows. The authors considered more than one aspect as the main objective function; most of the objectives were related to minimizing costs, which may include distance and time. Braekers et al. [20] considered the bi-objectives of minimizing total costs including routing and overtime costs and minimizing inconvenience costs including qualifications and working time constraints. Later research addressed climate change by considering carbon emissions as one of the objectives. Niakan and Rahimi [21] proposed multiple objectives for the inventory and distribution problem in HHC. The objective functions used were total cost minimization, carbon emission minimization and customer satisfaction

maximization. Fathollahi-Fard et al. [22] proposed a bi-objective model to minimize total cost and carbon emissions. Their metaheuristic was based on the simulated annealing algorithm and the salp swarm algorithm that can efficiently find the Pareto optimum.

Since home healthcare routing and scheduling can be formulated as a vehicle routing problem, which is NP-hard, the studied problem also falls within NP-hard territory. Heuristics and metaheuristics are appropriate approaches to find the best solution for these problems in a reasonable amount of time [11]. Bertels and Fahle [23] proposed metaheuristics including simulated annealing and tabu search to find an initial solution and improved solution with improvement heuristics including swapping, moving and removing operators to solve HHC problems. They minimized the total weighted sum of cost and penalties. Maya Duque et al. [24] proposed a two-phase approach for vehicle routing problem to solve bi-objective functions including minimizing total distance and maximizing the coverage of service level to patients. The algorithm first optimized the service level and then improved total distance. Çakırgil et al. [25] also proposed a two-phase approach, which first clusters and then improves using a variable neighborhood search to solve the model with the objective of minimizing the total cost and total weighted completion time of multi-skill routing problems. Kordi et al. [26] proposed a variable neighborhood search to deal with multi-objective problems (minimizing total cost, greenhouse gas releases, maximum time spent by each caretaker and maximizing the matching level between nurses and patients) based on the epsilon approach. Cui et al. [27] proposed an adaptive genetic algorithm and a mathematical model for the routing problem with the customer satisfaction function. The satisfaction function is derived from the soft time window violation, the data for which is collected via a questionnaire. The reader should refer to [28] for a literature review related to various objective functions and algorithms for solving problems in HHC.

To summarize, Table 1 presents the objective functions and solution methods taken into account in the HHC context.

**Table 1.** Summary of previous review studies.

Reference	Objective Function	Solution Method
Bertels and Fahle [23]	Minimized the total weighted sum of cost and penalties	Simulated annealing and tabu search
Eveborn et al. [4]	Minimizing cost for assigning nurse to schedule	Set partitioning and matching algorithm
Akjiratikar et al. [12]	Minimizing total distance	Particle swarm optimization
Bredström and Rönnqvist [5]	Minimizing weighted sum of preferences, traveling time and one balancing variable	Exact
Dohn et al. [13]	Maximized total assigned tasks	Node covering
Trautsamwieser et al. [17]	Minimize the weighted sum of traveling times, waiting times, and the dissatisfaction levels of clients and nurses.	Variable neighborhood search
Rasmussen et al. [6]	Minimizing weighted sum of cost, visit preference, and uncovered visits	Branch and price
Allaoua et al. [15]	Minimize the total number of caretakers	Danzig-Wolfe decomposition technique and branch and price algorithm
Mankowska et al. [8]	Minimizing weighted sum of traveling time, late time, and maximum lateness	Adaptive neighborhood search
Maya Duque et al. [24]	Bi-objectives for minimizing total distance and maximizing the coverage of service level to patients	Two-phase approach, first optimized the service level and then improved total distance

Table 1. Cont.

Reference	Objective Function	Solution Method
Niakan and Rahimi [21]	Multi-objective of minimizing total cost, carbon-emission minimization, and maximizing customer-satisfaction	Two-phase approach, the equivalent auxiliary crisp model and the efficient fuzzy multi-objective method
Yuan et al. [18]	Minimize total travel cost and penalty cost for violating the time windows	Danzig-Wolfe decomposition and the branch and price algorithm
Ait Haddadene et al. [7]	Minimizing weighted sum of non-preference and total traveling time	Greedy randomized heuristic
Braekers et al. [20]	Bi-objectives of minimizing total cost including routing and overtime cost and minimizing inconvenience cost	Multi-directional local search
Fathollahi-Fard et al. [22]	Bi-objectives for minimizing total cost and carbon emissions.	Simulated annealing algorithm and the salp swarm algorithm
Çakırgil et al. [25]	Minimize total cost and total weighted completion time for multi-skill routing problems	Two-phase approach, first clusters and then improves using variable neighborhood search
Saksuriya and Likasiri [16]	Minimized the total completion time for each patient node	Particle swarm optimization
de Aguiar et al. [19]	Minimize the weighted sum of total travel time and service time, working time, and penalty cost	Exact
Kordi et al. [26]	Multi-objective for minimizing total cost, greenhouse gas releases, maximum time spent by each caretaker, and maximizing the matching level between nurses and patients	Variable neighborhood search
Cui et al. [27]	Minimizing cost function derived from the customer satisfaction	Adaptive genetic algorithm

### 3. Mathematical Modeling

As previously mentioned, the home healthcare routing and scheduling problem can be described as a vehicle routing problem with time windows and mixed integer programming. Caretakers have to start their route at the medical center and proceed to visit patient nodes. Only one caretaker can visit a patient node to provide healthcare services. Note that services can only begin after their earliest time window. If caretakers arrive early, they must wait until the specified time is reached. In addition, caregivers must begin before the end of the time window or the latest starting time, a restriction known as hard time windows. After completing their service at the final patient node, the caregiver must return to the medical center. There are two main objective functions in this case: minimizing the total finishing time difference and minimizing the total travel distance. Minimizing the total finishing time difference can help achieve a balanced workload among caretakers, as they will complete their work very close to each other. As a result, the number of tasks per caretaker is generally similar. Note that the total finishing time difference means the sum of the absolute value which is not in a linear form.

The required parameters for our problem are listed below.

Parameters:

$N = \{0, 1, 2, \dots, \mathcal{N}\}$  a set of patient nodes where node 0 represents origin node (health center or depot);

$K = \{1, 2, \dots, \mathcal{K}\}$  a set of caretakers;

$e_j$  = the earliest starting time for patient node  $j$ ;

$l_j$  = the latest starting time for patient node  $j$ ;

$q_j$  = demand for patient node  $j$ ;

$Q$  = maximum capacity for each caretaker;  
 $D = [d_{ij}]$  = traveled time matrix where  $d_{ij}$  is the traveled time from node  $i$  to  $j$ ;  
 $s_j$  = service time for service  $j$ ;  
 $r_{jk} = 1$  if the vehicle  $k$  can visit patient  $j$ ;  
 $M$  = A large number.

For variables, there is a binary variable  $x_{ij}^k$  representing the decision of vehicle (caretaker)  $k$  to travel on the route. The starting time variable  $t_j$  represents time that the caretaker starts the healthcare services.  $u_j$  represents the cumulative demand when the caretaker visits the patient node. The variable  $CMAX_k$  represents the calculated finishing time for caretaker  $k$  which is the time that caretaker visits and finishes their service at the last patient node on the route. We need this variable to calculate the total finishing time difference which is the sum of the absolute value of every pair of finishing times. The details of the variables are listed below.

Variables:

$x_{ij}^k$                       1 if vehicle  $k$  travels from patient node  $i$  to patient node  $j$ ;  
 $t_j$                          Starting time at patient node  $j$ ;  
 $u_j$                          Variable for the subtour elimination;  
 $CMAX_k$                     Finishing time (Maximum completion time) for caretaker  $k$ .

Given a collection of patient nodes  $N = \{1, \dots, \mathcal{N}\}$  and a collection of caretakers  $K = \{1, 2, \dots, \mathcal{K}\}$ , the objective of the problem is to determine the optimal routes for each caregiver beginning and ending at the health center. Each caretaker must arrive at the patient node within the specified time frame. The problem can be formulated as mixed integer programming (MIP) problem with the incorporation of routing constraints and scheduling constraints (constraints related to starting time and time window).

Model formulation

$$\text{Minimize } \left\{ \sum_{k_1, k_2 \in K \times K, k_1 \neq k_2} |CMAX_{k_1} - CMAX_{k_2}|, \sum_{i, j \in N, i \neq j, k \in K} x_{ij}^k d_{ij} \right\}$$

$$\sum_{j=1}^n x_{0j}^k = 1, \forall k \in K \tag{1}$$

$$\sum_{j=1}^n x_{j0}^k = 1, \forall k \in K \tag{2}$$

$$\sum_{k=1}^m \sum_{i=0}^n x_{ij}^k = 1, \forall j \in N \setminus \{0\} \tag{3}$$

$$\sum_{k=1}^m \sum_{j=0}^n x_{ij}^k = 1, \forall i \in N \setminus \{0\} \tag{4}$$

$$\sum_{i=0}^n x_{ij}^k - \sum_{l=0}^n x_{jl}^k = 0, \forall j \in N \setminus \{0\}, \forall k \in K \tag{5}$$

$$\sum_{i=0}^n x_{ij}^k \leq r_{jk}, \forall j \in N, \forall k \in K \tag{6}$$

$$u_i - u_j + q_j \leq Q(1 - x_{ij}^k), \forall i, j \in N \setminus \{0\}, i \neq j, \forall k \in K \tag{7}$$

$$q_j \leq u_j \leq Q, \forall j \in N \setminus \{0\} \tag{8}$$

$$t_i + s_j + d_{ij} - M(1 - x_{ij}^k) \leq t_j, \forall i \in N, i \neq j, \forall j \in N, \forall k \in K \tag{9}$$



$$e_j \leq t_j \leq l_j, \forall j \in N \tag{10}$$

$$t_i + s_i - M(1 - x_{i0}^k) \leq CMAX_k, \forall i \in N, \forall k \in K \tag{11}$$

$$t_i + s_i + M(1 - x_{i0}^k) \geq CMAX_k, \forall i \in N, \forall k \in K \tag{12}$$

$$x_{ij}^k \in \{0, 1\}, \forall i, j \in N, i \neq j, \forall k \in K \tag{13}$$

The objective function is to simultaneously minimize the total finishing time difference and total distance. Constraints (1) and (2) ensure that each caretaker’s route begins and ends at the origin, given that the objective function is to minimize the difference in finishing times of each pair of caretakers and minimize the total distance. Constraints (3) and (4) limit the number of caretaker visits and departures per patient to one. Additionally, constraint (5) ensure that a caregiver moves on to the next patient in the sequence after providing care to the present one. The compatibility constraint (6) ensures that only a caregiver who is compatible with the patient will travel to provide care for them. Constraints (7) and (8) are subtour elimination constraints and capacity constraints, respectively. Constraint (9) becomes active when caregiver  $k$  transitions from patient  $i$  to patient  $j$  ( $x_{ij}^k = 1$ ). The completion time for patient  $i$  (starting time plus service duration plus travel time) must be less than that for patient  $j$ . Constraint (10) involves time windows.  $M$  in constraints (11) and (12) is a sufficiently large number used to calculate the maximum completion time for each caretaker. Constraint (13) guarantees variable integrality.

Since the objective function is not in the form of a linear equation, new variables that represent the difference between a pair of finishing times are introduced,  $CD_{k_1, k_2}$  for  $k_1, k_2 \in K \times K$  and  $k_1 \neq k_2$ . The new constraints are described as:

$$CMAX_{k_1} - CMAX_{k_2} \leq CD_{k_1, k_2}, \forall k_1, k_2 \in K \times K, k_1 \neq k_2 \tag{14}$$

$$CMAX_{k_2} - CMAX_{k_1} \leq CD_{k_1, k_2}, \forall k_1, k_2 \in K \times K, k_1 \neq k_2 \tag{15}$$

The model can therefore be converted to mixed linear integer programming by adding constraints (14) and (15), and replacing the objective function with

$$\text{Minimize} \left\{ \sum_{k_1, k_2 \in K \times K, k_1 \neq k_2} CD_{k_1, k_2}, \sum_{i, j \in N, i \neq j, k \in K} x_{ij}^k d_{ij} \right\} \tag{16}$$

#### 4. Solution Method

Vehicle Routing with Time Windows is a well-known NP-hard problem whose complexity increases exponentially with the number of variables. Only in rare instances can the problem be precisely solved. Moreover, when the number of nodes is increased, the algorithm is unable to discover an exact solution in some instances. Consequently, metaheuristics are alternative techniques for locating solutions that are sufficient although not optimal.

The simulated annealing algorithm (SA) is a local search-based heuristic algorithm that can prevent becoming caught in a local optimum and help explore a larger portion of the search space. Inspired by the work of [29] on the annealing process, SA has been effectively used to solve several combinatorial problems, including location, vehicle and other related problems. The SA algorithm typically begins with an initial solution, which is subsequently improved with each iteration by neighborhood operators (or local search algorithms) based on the problem’s characteristics. Neighborhood operators such as swap, move, insert, reverse, ruin-and-recreate, path exchange, etc., are typically employed to address routing problems. Each algorithm containing SA that is used to solve a combinatorial optimization problem always reaches a local optimum. One method for escaping the local optimum is to allow the solution to relocate to a neighborhood solution that is worse than the current one,

and then apply the same neighborhood operators to bring the solution back to a new local optimum. In SA, probability is used to determine when the worst potential solution should be accepted. The probability is determined by the variable temperature, which decreases with every iteration. The higher the temperature, the greater the opportunity to switch to a worse solution. In this case, we use the Boltzmann function to calculate the probability of accepting the worst solution.

As mentioned earlier, our model was formulated using bi-objective mixed integer programming. Our proposed algorithm is based on simulated annealing to solve the problem by converting it to a single-objective problem using the weighted sum of two objectives. In this case, there are two weight coefficients associated with the objective functions where the sum of weights is equal to 1. There are multiple choices to choose from; for the best performance of the algorithm, we limited the set of weights to have only 11 choices:  $\{(w_i, w_j) | i, j = 0, 1, \dots, 10, w_i + w_j = 10\}$ . The approach aims to find the Pareto optimal set, which is a set of non-dominated solutions to each other but each is superior to the rest of the solutions in the searching space. Non-dominated solutions are those that have no solution that has a less objective value for all objectives. To optimize multiple objective functions, one objective needs to be sacrificed to improve another. The Pareto set for two objectives, if plotted two-dimensionally, will lie on the surface known as the Pareto front. To obtain the best Pareto set of solutions, we solved the problem using each weighted sum as the single objective function and selected only the Pareto solutions to remain in the set.

As mentioned above, the problem is an NP-hard problem. To find an optimal solution within a reasonable execution time, we needed to improve the algorithm's efficiency. Here, we provide feasible conditions for preventing the algorithm from searching in an unnecessary solution space. For convenience, the solution is represented in the single vector form. The vector will start with the node "0" which is the depot node, followed by the sequence of nodes and finally end at the depot. For example,  $[0, 1, 2, 3, 0, 4, 5, 0, 6, 7, 8, 0]$  represents a solution with three routes: the first route starts from depot and visits nodes 1, 2 and 3 before returning to the depot; the second route starts from depot, visits nodes 4 and 5 and then returns to depot; and the third route contains nodes 6, 7 and 8. For the feasibility test, we chose the conditions to ensure the feasibility of the algorithm. For convenience, the route  $i$  can be written as  $R_k = [0, \delta_k^1, \delta_k^2, \dots, \delta_k^{|R_k|}, 0]$  where each  $\delta_k^j$  represents the node that vehicle  $k$  visits in the  $j^{th}$  order.

**Definition 1.** The starting time for node  $\delta_k^j$  is equal to the maximum between the lower time window of the node, i.e.,  $e_{\delta_k^j}$ , and the starting time of the immediately preceding node together with the travel time between them and service time, i.e.,  $t(\delta_k^{j-1}) + p_{\delta_k^{j-1}} + d_{\delta_k^{j-1}, \delta_k^j}$ .

$$\text{Thus, } t(\delta_k^j) = \max \left\{ e_{\delta_k^j}, t(\delta_k^{j-1}) + p_{\delta_k^{j-1}} + d_{\delta_k^{j-1}, \delta_k^j} \right\}$$

**Definition 2.** The completion time for node  $\delta_k^j$ , namely  $c(\delta_k^j)$ , is equal to the combination of the starting time and the service time of the node, i.e.,  $c(\delta_k^j) = s(\delta_k^j) + p_{\delta_k^j}$ .

**Definition 3.** The finishing time for node  $\delta_k^j$ , namely  $f(\delta_k^j)$ , is equal to the maximum of completion time between all nodes visited by caretaker  $k$ , i.e.,  $f(\delta_k^j) = \max_{j \in \{1, 2, \dots, |R_k|\}} \{c(\delta_k^j)\}$ .

**Lemma 1.** Let  $R_k = [0, \delta_k^1, \delta_k^2, \dots, \delta_k^{|R_k|}, 0]$  be a routing sequence for vehicle  $k$ . The node  $\delta$  can be feasibly inserted into the routing sequence  $R_k$  next to the  $i$  position, i.e., between  $\delta_k^i$  and  $\delta_k^{i+1}$  if the following conditions hold:

$$t(\delta_k^{i+1}) - d_{\delta, \delta_k^{i+1}} - p_\delta \geq e_\delta$$



$$c(\delta_k^i) + d_{\delta_k^i, \delta} + p_\delta \leq l_\delta$$

$$t(\delta_k^{i+1}) - c(\delta_k^i) \geq d_{\delta_k^i, \delta} + p_\delta + d_{\delta, \delta_k^{i+1}}$$

**Proof.** Suppose the node  $\delta$  is inserted between  $\delta_k^i$  and  $\delta_k^{i+1}$ . The completion time for node  $\delta$  can be calculated as

$$t(\delta) = \max\{e_\delta, c(\delta_k^i) + d_{\delta_k^i, \delta}\}.$$

Consequently, the starting time for the node  $\delta_k^{i+1}$  can be recalculated as:

$$t(\delta_k^{i+1}) = \max\{e_{\delta_k^{i+1}}, c(\delta) + d_{\delta, \delta_k^{i+1}}\}.$$

To check for feasibility, we must ensure that the starting times of  $\delta$  and  $\delta_k^{i+1}$  lie within their time windows. Specifically, the starting time for  $\delta_k^{i+1}$  needs to be recalculated; we set it as  $t^*(\delta_k^{i+1})$ .

If  $e_\delta \geq c(\delta_k^i) + d_{\delta_k^i, \delta}$ , the starting time for  $\delta$  is set to be equal to its lower time window, i.e.,  $t(\delta) = e_\delta \leq l_\delta$ . Consequently, from condition (1), we obtain

$$t^*(\delta_k^{i+1}) \geq t(\delta_k^{i+1}) \geq e_\delta + d_{\delta, \delta_k^{i+1}} + p_\delta.$$

The starting time for  $\delta_k^{i+1}$  can thus be set to remain the same. Therefore, the starting times for  $\delta$  and  $\delta_k^{i+1}$  lie within their time windows.

Similarly, in the case of  $e_\delta \leq c(\delta_k^i) + d_{\delta_k^i, \delta}$ , by using condition (3), the new starting time for  $\delta_k^{i+1}$  also satisfies its time windows.

For  $\delta$ , from the assumption,  $t(\delta) = \max\{e_\delta, c(\delta_k^i) + d_{\delta_k^i, \delta}\}$  and condition (2), the starting time is calculated to be  $t(\delta) = c(\delta_k^i) + d_{\delta_k^i, \delta} \leq l_\delta - p_\delta \leq l_\delta$ . The starting times for  $\delta$  and  $\delta_k^{i+1}$  therefore lie within their time windows, and hence the Lemma is proved.  $\square$

Lemma 1 can serve as a basis for pruning inefficient explorations within the algorithm. These conditions may be integrated into the insertion process and beyond. Furthermore, the acts of swapping and moving, which entail additional steps, are analogous to the processes of removal and insertion.

#### Simulated Annealing (SA)

Our proposed algorithm is based on SA, as we include both our balancing and insertion algorithms. The algorithm also employs general neighborhood operators. At each iteration, one of the neighborhood operators will be chosen based on their performance records (probability). At the start, the probability for each operator is identical. The variables  $pc_1, pc_2$  and  $pc_3$  represent the number of improvements that arise from applying the moving, swapping and path exchanging operators, respectively. Using the roulette wheel selection method frequently used in Genetic algorithm [30], the probability of choosing the operator  $i$  can be calculated by  $pc_i / (pc_1 + pc_2 + pc_3)$ . At each iteration, if applying the operator  $i$  can improve the solution, the number  $pc_i$  will be increased by 1; otherwise, the number will be decreased by 1. The pseudocode is presented in Algorithm 1 below.

**Algorithm 1.** The framework of SA

1. **Input:** Initial temperature as  $T$ , terminating temperature as  $T_m$ , decreasing rate as  $\alpha$ , and maximum iteration as  $I_m$ .
2. **Initial step:** Generate an initial solution using Algorithm 2, set as  $Sol$ . Set the counter iteration variable as  $i = 1$ .  
The number of improved  $pc_1 = pc_2 = pc_3 = 1$ .
3. **While** the termination criteria do not meet ( $T > T_m$  and  $i < I_m$ ) **do**
4.     Choose the Local search procedure based on the probability.
5.     Apply Local search procedures to the solution, set as  $Sol_{new}$ .
6.     **If** the objective value of  $Sol_{new} <$  objective value of  $Sol$  **do**
7.         Update new solution,  $Sol = Sol_{new}$ .
8.         **If** the objective value of  $Sol <$  objective value of  $Sol_{best}$  **do**
9.              $Sol_{best} = Sol$  and increase the number of improvements of the operator by 1.
10.         **End If**
11.     **Else**
12.         Randomize a value uniformly, set to  $s$ .
13.         **If**  $s < \exp(-t)$  **do**
14.             Accept new solution,  $Sol = Sol_{new}$ .
15.         **End If**
16.     **End If**
17.     Apply ruin-and-recreate procedure.
18.     Increase the number of iterations,  $i = i + 1$ .
29. **End While**
20. **Output :**  $Sol_{best}$

**4.1. Initial Solution**

To obtain qualified solutions, it is preferable to start with a good one. In this section, the insertion algorithm is proposed. An initial solution inspired by the insertion operator is generated. The fundamental concept is to assign a patient to the routing sequence (caretaker) with the fewest patients, so that the number of patients each caregiver must visit will be balanced. In addition, the new patient will be added to the position in the sequence with the earliest completion time. When a patient is inserted into a sequence of other patients, it is common for subsequent patients to have to delay their procedures. To fix this, tardy patients will be removed from the route and reassigned to other positions. This procedure will continue until no patients remain. Moreover, the procedure will try to minimize both objectives simultaneously by choosing the minimum finishing time as the primary and the minimum total distance as the secondary objective. The pseudocode is given in Algorithm 2.

**Algorithm 2.** Inserting procedure

1. **Input:** A set of patients (waiting to be assigned), set to  $J_c$  and a partial solution,  $J = \{J_1, J_2, \dots, J_K\}$  where  $J_i$  is the routing sequence of route  $i$ , and  $K$  is the number of caretakers.
2. **Set**  $J_w = \emptyset$  and arrange the patients in  $J_c$  according to their upper time windows in ascending order.
3. **While**  $J_c \neq \emptyset$  **do**
4.     Choose the first patient in  $J_c$ , say  $j$ . Remove  $j$  from  $J_c$
5.     Choose a caretaker  $k$  that has the minimum number of patients. Consider all inserting assignments of  $j$  into  $J_k$ .
6.     **If** there is more than one inserting assignment such that all patients are not late **do**
7.         Choose the case that has the minimum finishing time and if there are the same value of finishing time, choose the minimum distance, replace the route into  $J$ .
8.     **Else If** there are tardy patients in every inserting assignment of  $j$  **do**
9.         Choose the assignment that has the lowest number of tardy patients and whose inserting assignment's completion time after removing tardy patients is the smallest, replace the route in  $J$ . Add the tardy patients to  $J_w$ .
10.     **End If**
11. **End While**
12. **While**  $J_w \neq \emptyset$  **do**
13.     Add a new caretaker. Repeat all steps in the While loop with  $J_w$ , instead of  $J_c$ , until there is no change in  $J$ .
14. **End While**

### 4.2. Neighborhood Operators

In this study, the solution space is explored using three neighborhood operators, namely swap, move and path exchange. After obtaining an initial solution, neighborhood operators will iteratively enhance the solution. Each iteration will select one of these operators at random. The procedures are detailed below.

**Moving operator:** The objective of the operator is to move patients from their current location in the routing sequence to a new location where they will not be tardy. This operator will begin moving patients from the route with the highest number of patients to the route with the lowest number of patients in order to balance out completion times. Even if the completion time increases, it is still possible for the solution to be accepted due to its probability. Figure 1 below shows the example of such moving operation.

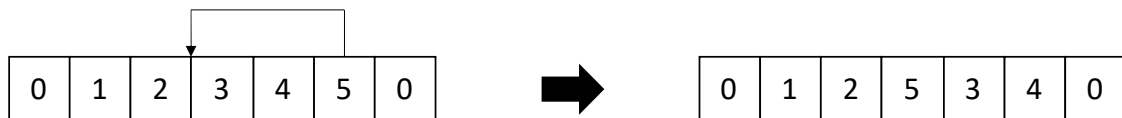


Figure 1. Example of moving operation.

**Swapping operator:** Shown in Figure 2 is the example of swapping operation. The objective of this operator is to swap the locations of two patients. Similarly to the moving operator, the switch operator will be applied to the pair of patients that will not cause other patients to be late and will accept the worst possible solution.

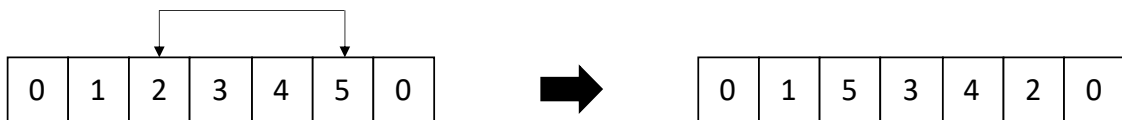


Figure 2. Example of swapping operation.

**Path exchange operator:** The purpose of the path exchange operator is to choose an edge and exchange the path either at the start or end of that edge in a pair of routes. It is analogous to the swap operator, except that it swaps a pair of edges. The operator will be applied if the result is feasible. For example, as shown in Figure 3, path 4-5-0 is selected from the first route and path 0-6-7 is selected from the second route. The operator will then concatenate the two selected paths and two unselected paths together.

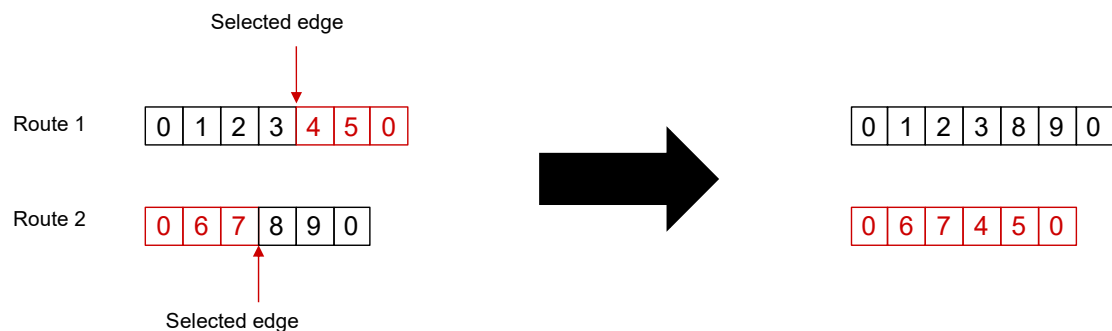


Figure 3. Example of path exchange operation.

### 4.3. Ruin and Recreate Procedure

The procedure aims to help escape from the local optimum by deleting one of the caretakers (routes) and trying to feasibly insert the deleted patients into the remaining routes using the Algorithm 2. If there is no place to insert them, a new caretaker will be spawned. The main idea is to jump out of being stuck in a local optimum and reduce the total number of caretakers at the same time [31]. The pseudocode is presented in Algorithm 3.

---

**Algorithm 3.** Ruin and Recreate procedure

---

1. **Input:** The routing sequence
  2. Randomly select one of the caretakers (route) in the solution and delete it, set the removed patients to *Removed\_patients*  
Apply Algorithm 2 by inputting the set of patients with Removed patients to the solution, if
  3. Algorithm 2 cannot find a feasible solution or cannot insert all remaining patients, a new caretaker will be spawned.
- 

**5. Results and Discussions**

To evaluate the efficacy of the proposed algorithm, we applied it to the well-known benchmark instances, Solomon’s instances. The initial objective behind the creation of Solomon’s instances was to minimize the total number of customers and total travel distance [32,33]. In this work, we modify the benchmark instances to better suit our problem. The vehicles can represent caretakers, while the customer nodes can represent patients. The objective function is modified into a bi-objective optimization problem to minimize the total finishing time differences between each pair of caretakers and minimize total distance. The benchmark includes six varieties based on the node distribution pattern and the range of time windows: random with narrow and wide time windows (R1, R2), clustered with narrow and wide time windows (C1, C2), and a mixture of random and clustered with narrow and wide time windows (RC1, RC2). All of the algorithms in this study were coded using the Julia programming language [34,35] and executed on AMD Ryzen 9 12-core processors with 3.8 GHz CPU and 64 GB of RAM.

To study the sensitivity of the two objectives, each of them will first be used as a single objective to the problem and then the bi-objectives will be considered and compared using the proposed method in Section 3.

Table 2 presents the computational results of a comparison between our algorithm and the exact algorithm for the problem with an objective function to minimize the total finishing time differences between each pair of caretakers. To clarify the column headings in Table 2, Name represents the name of the instance starting with the distribution of the problem followed by the width of time windows and the numbering of problem in the last two digits. Time represents execution time in seconds. The rest of the columns show a comparison between our results and the optimal solutions obtained by the exact algorithm [36]. The second and third columns show the number of vehicles obtained by our algorithm and by the exact algorithm, respectively. The fourth and fifth columns show the calculated total distance from our solution and the optimal solution, respectively. The objective values (minimizing the total finishing time differences in each pair of caretakers) from our algorithm and from the exact algorithm are reported in the sixth and seventh columns. Columns eight and nine show, in brackets, the minimum and maximum number of patients per caretaker, respectively. Note that cases where the optimal solution cannot be found within the defined execution time (1 h) are indicated by “-”. Additionally, Table 2 reports only the lowest number of caretakers. We can see that the algorithm can find the optimal solution in most cases for 25 customers. In some instances, the algorithm can find another optimal solution (same value of total difference) that is more balanced (the minimum number of patients per caretaker is greater while the maximum number is lower) than the optimal solution obtained from the Gurobi solver.

**Table 2.** Computational results for VRPTWs ability to minimize the total difference between each pair of caretakers (w10\_w0).

Name	Number of Vehicles		Total Distance		Total Difference		(Minimum, Maximum) Patients per Caretaker	
	Our	Opt	Our	Opt	Our	Opt	Our	Opt
C101	3	3	267.8	277.2	68.2	68.2	(6, 10)	(7, 10)
C102	4	3	421.5	405.5	2.8	0.2	(3, 9)	(7, 10)

Table 2. Cont.

Name	Number of Vehicles		Total Distance		Total Difference		(Minimum, Maximum) Patients per Caretaker	
	Our	Opt	Our	Opt	Our	Opt	Our	Opt
C103	4	3	520.6	481.5	0.0	0.2	(2, 8)	(7, 9)
C104	4	3	614.3	523.8	0.3	0.0	(4, 8)	(7, 9)
C105	3	3	520.7	532.5	0.8	0.2	(7, 9)	(7, 9)
C106	3	3	354.4	446.4	59.2	59.2	(8, 9)	(7, 9)
C107	3	3	426.3	497.9	0.0	0.0	(8, 9)	(8, 9)
C108	3	3	425.1	518.7	0.0	0.2	(7, 9)	(7, 9)
C109	4	3	492.2	588.7	0.3	0.2	(2, 8)	(7, 9)
C201	2	2	476.4	385.2	95.5	95.5	(8, 17)	(9, 16)
C202	2	2	578.9	589.0	0.0	0.0	(8, 17)	(10, 15)
C203	3	2	619.7	549.6	0.2	0.0	(3, 18)	(11, 14)
C204	3	2	725.5	674.2	0.0	0.0	(7, 10)	(12, 13)
C205	2	2	426.6	403.7	0.0	0.0	(9, 16)	(12, 13)
C206	2	2	460.8	401.4	0.0	0.0	(8, 17)	(7, 18)
C207	3	2	464.0	442.9	0.2	0.0	(3, 15)	(9, 16)
C208	2	2	537.0	567.4	0.0	0.0	(11, 14)	(9, 16)
R101	8	8	830.5	859.5	431.8	431.8	(2, 4)	(2, 4)
R102	7	7	858.3	878.3	50.4	36.8	(3, 5)	(3, 4)
R103	7	5	802.1	759.5	17.6	154.8	(2, 5)	(4, 6)
R104	6	-	820.8	-	8.4	-	(3, 5)	-
R105	7	6	797.6	780.8	72.2	6.3	(3, 4)	(4, 5)
R106	6	-	735.5	-	9.4	-	(3, 5)	-
R107	6	-	847.4	-	6.3	-	(3, 5)	-
R108	6	4	791.0	653.8	5.4	96.4	(2, 6)	(5, 7)
R109	7	5	898.1	792	16.2	2.4	(3, 4)	(5, 5)
R110	8	-	1009.3	-	18.5	-	(2, 4)	-
R111	6	5	830.1	764.1	5.9	1.8	(3, 5)	(4, 6)
R112	8	-	852.7	-	17.8	-	(2, 4)	-
R201	2	2	797.3	817.0	1.3	1.1	(4, 21)	(6, 19)
R202	2	4	732.4	971.1	0.0	23.5	(7, 18)	(6, 7)
R203	3	3	804.0	912.7	0.0	0.2	(5, 13)	(5, 13)
R204	2	2	754.1	908.7	0.0	0.0	(12, 13)	(12, 13)
R205	2	3	831.9	890.2	0.0	1.2	(8, 17)	(4, 11)
R206	3	3	800.3	942.8	0.0	0.4	(3, 17)	(7, 10)
R207	3	3	1017.3	811.5	0.2	0.6	(3, 15)	(7, 11)
R208	3	2	897.0	922.4	0.0	0.0	(4, 12)	(9, 16)
R209	2	2	696.2	811.1	0.0	0.0	(7, 18)	(11, 14)
R210	2	3	789.3	920.2	0.0	0.2	(9, 16)	(3, 16)
R211	4	2	862.2	811.1	0.9	0.0	(4, 12)	(12, 13)
RC101	6	4	743.9	535.2	13.9	43.5	(2, 7)	(4, 8)
RC102	6	3	833.6	411.3	9.4	54.4	(3, 6)	(8, 9)
RC103	6	-	846.2	-	2.3	-	(3, 6)	-
RC104	6	-	978	-	1.9	-	(3, 5)	-
RC105	5	4	673.2	519.1	39.4	3.9	(3, 8)	(5, 8)
RC106	7	3	752.2	395.4	22.6	0.6	(2, 6)	(8, 9)
RC107	6	-	822.4	-	2.6	-	(2, 6)	-
RC108	3	-	382.9	-	2.4	-	(8, 9)	-
RC201	2	3	778.5	1108.4	0.0	2.6	(5, 20)	(7, 11)
RC202	3	3	825.9	1336.4	0.0	0.4	(2, 16)	(7, 11)
RC203	2	3	896.7	953.2	0.0	0.4	(9, 16)	(5, 12)
RC204	3	3	1033.9	930	0.0	0.4	(2, 17)	(5, 14)
RC205	2	3	1031.8	1103.9	0.0	13.2	(9, 16)	(5, 13)
RC206	3	3	778.3	1318.4	0.2	0.0	(2, 18)	(4, 11)
RC207	2	3	807.3	1095.9	0.0	0.2	(7, 18)	(4, 14)
RC208	1	2	596.3	1074.8	0.0	0.0	(25, 25)	(10, 15)

To demonstrate the performance of our algorithm, the algorithm has been used to solve the Solomon benchmark instances with an objective function to minimize total distance. In this case, the algorithm uses only total distance as its objective. Table 3 shows the differences in the results for 25 customers between the best-known solutions and the solutions obtained by our algorithm. Columns 2, 4, 6, 8 show the solution obtained using our algorithm with number of vehicles, total distance, total difference and minimum and maximum number of patients per caretaker, respectively; columns 3, 5, 7, 9 show the optimal solution. It is

evident that our algorithm obtains the optimal solutions in almost all cases. Moreover, we observe that, in instance R104, our solution has a distance that is slightly higher than the optimal solution, but the maximum number of jobs is lower. Note that the solution that cannot be found within the defined time frame (1 h) is indicated as “-”. In this case, the solution obtained via our algorithm is a more balanced solution compared to the optimal solution using the same objective function, making it more reasonable to use in the real world. In contrast, we notice that, when the total distance is minimized, the minimum and maximum numbers of patients per caretaker become more imbalanced than when only total difference between finishing times is minimized, e.g., C101, R105, R101.

**Table 3.** Computational results for VRPTW to minimize total distance only (w0\_w10).

Instance	Number of Vehicles		Total Distance		Total Difference		(Minimum, Maximum) Patients per Caretaker	
	Our	Opt	Our	Opt	Our	Opt	Our	Opt
C101	3	3	191.3	191.3	505.6	505.6	(6, 11)	(6, 11)
C102	3	3	190.3	190.3	819.8	819.8	(6, 11)	(6, 11)
C103	3	3	190.3	190.3	819.8	819.8	(6, 11)	(6, 11)
C104	3	3	186.9	186.9	596.0	596.0	(6, 10)	(6, 10)
C105	3	3	191.3	191.3	505.6	505.6	(6, 11)	(6, 11)
C106	3	3	191.3	191.3	505.6	505.6	(6, 11)	(6, 11)
C107	3	3	191.3	191.3	505.6	505.6	(6, 11)	(6, 11)
C108	3	3	191.3	191.3	505.6	505.6	(6, 11)	(6, 11)
C109	3	3	191.3	191.3	505.6	505.6	(6, 11)	(6, 11)
C201	2	2	214.7	214.7	1722.1	1722.1	(6, 19)	(6, 19)
C202	2	2	214.7	214.7	1242.2	1242.2	(6, 19)	(6, 19)
C203	2	2	214.7	214.7	1722.1	1722.1	(6, 19)	(6, 19)
C204	2	2	214.5	214.5	1721.9	1242.0	(6, 19)	(6, 19)
C205	2	2	214.7	214.7	1722.1	1722.1	(6, 19)	(6, 19)
C206	2	2	214.7	214.7	1733.8	1733.8	(6, 19)	(6, 19)
C207	2	2	214.5	214.5	1888.1	1888.1	(6, 19)	(6, 19)
C208	2	2	214.5	214.5	2197.9	2197.9	(6, 19)	(6, 19)
R101	8	8	617.1	617.1	1040.0	1040.0	(1, 5)	(1, 5)
R102	7	7	547.1	547.1	797.2	797.2	(1, 5)	(1, 5)
R103	5	5	454.6	454.6	286.8	286.8	(2, 7)	(2, 7)
R104	4	4	430.1	416.8	94.2	39.8	(5, 8)	(5, 8)
R105	6	6	530.5	530.5	365.5	365.5	(3, 5)	(3, 5)
R106	5	-	465.4	-	413.6	-	(4, 7)	(-, -)
R107	4	4	424.3	424.3	145.2	145.2	(5, 8)	(5, 8)
R108	4	4	397.3	397.3	199.0	199.0	(5, 8)	(5, 8)
R109	5	5	441.3	441.3	311.6	311.6	(4, 6)	(4, 6)
R110	5	4	444.7	444.7	366.6	90.5	(3, 6)	(5, 7)
R111	5	5	430.1	430.1	523.4	523.4	(1, 7)	(1, 7)
R112	4	4	394.0	393.0	146.5	75.0	(5, 8)	(5, 8)
R201	4	4	463.3	463.3	1250.9	1250.9	(6, 7)	(6, 7)
R202	4	4	410.5	410.5	1411.0	1411.0	(5, 7)	(5, 7)
R203	3	3	393.9	391.4	529	569.4	(5, 11)	(7, 10)
R204	2	2	357.5	355.0	158.2	459.4	(12, 13)	(8, 17)
R205	3	3	399.0	393.0	156.0	166.0	(3, 13)	(3, 15)
R206	3	3	374.4	374.4	453.0	453.0	(6, 12)	(6, 12)
R207	3	3	361.6	361.6	453.0	453.0	(6, 10)	(6, 10)
R208	2	2	334.6	334.6	244.3	244.3	(12, 13)	(12, 13)
R209	2	2	370.7	370.7	3.0	3.0	(10, 15)	(10, 15)
R210	3	3	404.6	404.6	617.0	617.0	(6, 11)	(6, 11)
R211	2	2	350.9	350.9	96.1	96.1	(12, 13)	(12, 13)



Table 3. Cont.

Instance	Number of Vehicles		Total Distance		Total Difference		(Minimum, Maximum) Patients per Caretaker	
	Our	Opt	Our	Opt	Our	Opt	Our	Opt
RC101	4	4	461.1	461.1	157.8	157.8	(3, 8)	(3, 8)
RC102	4	3	401.6	351.8	196.0	43.6	(2, 9)	(8, 9)
RC103	3	3	333	332.8	65.4	29.2	(8, 9)	(8, 9)
RC104	3	3	307.2	299.7	51.6	54.8	(8, 9)	(8, 9)
RC105	4	4	412.5	411.3	212.3	227.3	(3, 8)	(3, 8)
RC106	3	3	345.5	345.5	28.2	28.2	(8, 9)	(8, 9)
RC107	3	3	298.3	298.3	38.2	38.2	(8, 9)	(8, 9)
RC108	3	3	294.5	294.5	81.4	81.4	(8, 9)	(8, 9)
RC201	3	3	360.2	360.2	188.0	188.0	(8, 9)	(8, 9)
RC202	3	3	338.0	338.0	200.0	200.0	(8, 9)	(8, 9)
RC203	3	3	326.9	326.9	240.8	240.8	(8, 9)	(8, 9)
RC204	3	3	299.7	299.7	292.8	292.8	(8, 9)	(8, 9)
RC205	3	3	338.0	338.0	334.8	334.8	(8, 9)	(8, 9)
RC206	3	3	324	324	170.8	170.8	(8, 9)	(8, 9)
RC207	3	3	298.3	298.3	276.0	276.0	(8, 9)	(8, 9)
RC208	2	2	270.9	269.1	140.8	132.4	(8, 17)	(8, 17)

To illustrate and analyze the solution obtained from the problem converted from bi-objective to single objective using weighted parameters, boxplot graphs for the calculated total difference and total distance are presented in Figures 4 and 5. The objective function  $w_i w_j$  where  $i, j \in \{0, 1, \dots, 10\}$  represent the weighted sum of total finishing time differences and total distance, i.e.,  $w_1 \times \sum_{i,j \in N, i \neq j} CD_{ij} + w_2 \times \sum_{i,j \in N, i \neq j, k \in K} x_{ij}^k d_{ij}$  where  $w_1 + w_2 = 1$ . As shown by the boxplot of the objective function and total distance, the total distance does not have any cases of outliers, and the minimum distances in each case are comparable to each other. In contrast, we can see that there are outliers in each case of the objective function for the total finishing time difference because the total difference is hard to optimize. Moreover, for the objective function  $w_0 w_{10}$ , which minimizes only the distance, the balance between finishing times and number of jobs per caretaker is extremely wide, making it unsuitable for balancing the workload.

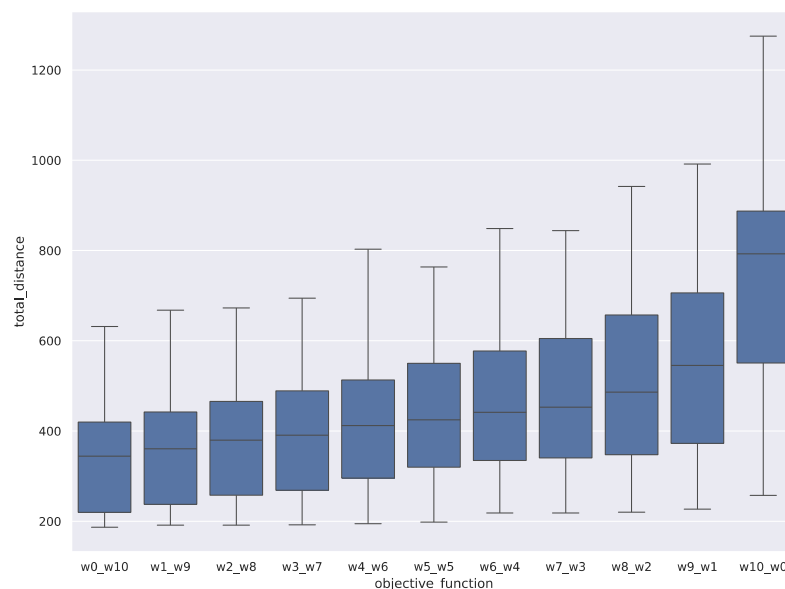
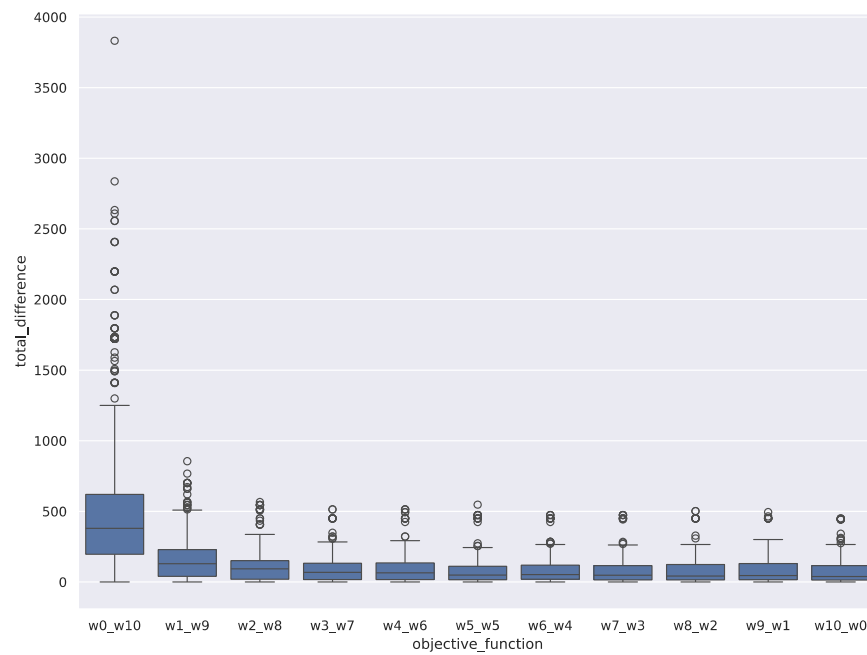


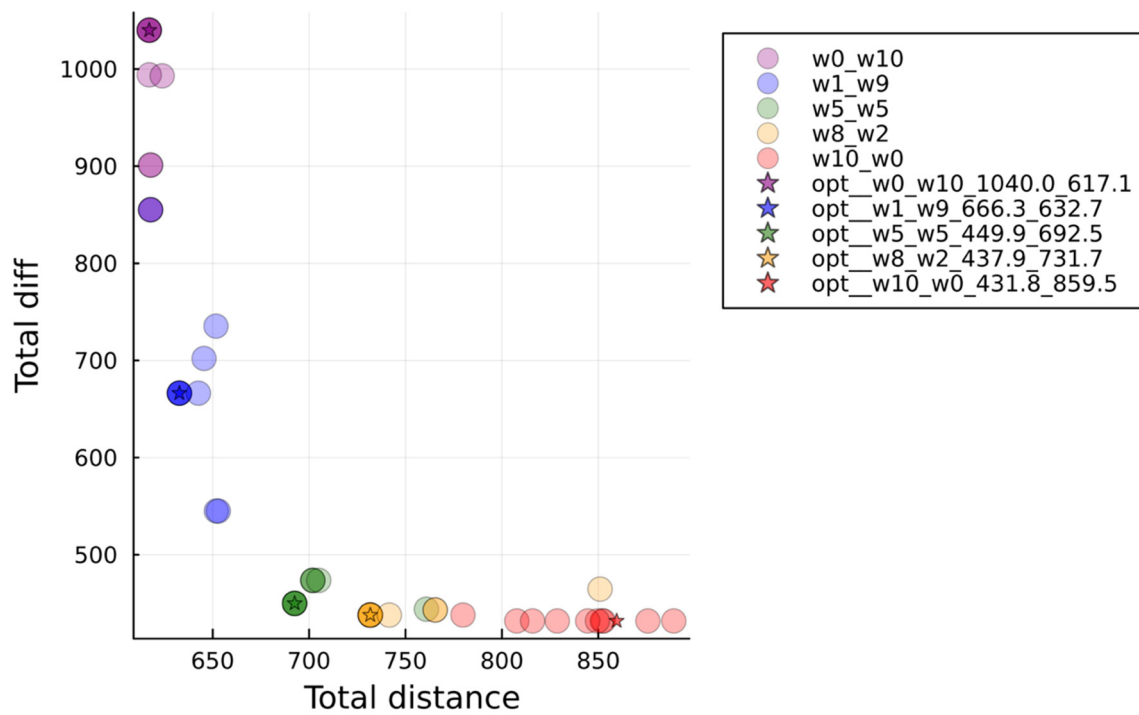
Figure 4. Boxplot between objective functions and calculated total distance.



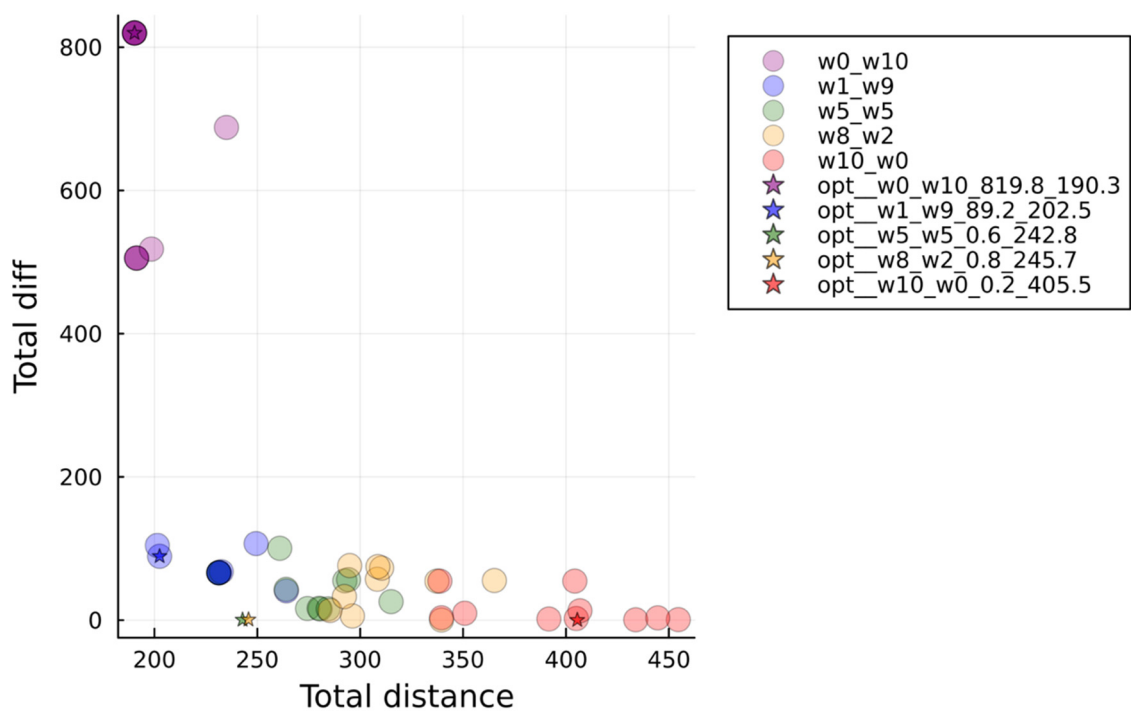
**Figure 5.** Boxplot between objective functions and calculated total finishing time differences.

Shown in Figures 6–8 are the selected plots to illustrate the differences in the 10 solutions between the total finishing time difference and total distance for the selected objective functions, for instance, R101, C102, and RC201. The circles represent solutions obtained by our algorithm, and they are darker when there is an overlap between the same solutions. The star represents the optimal solution obtained from the Gurobi solver; in this case, the calculated values for the total finishing time differences and total distance for the solution are suffixed and separated by “\_” from the name of objective function, respectively. For example, “opt\_w1\_w\_9\_666.3\_632.7” represents the optimal solution for an objective function with a weight of 0.1 for the total distance, a weight of 0.9 for the finishing time difference, a calculated total finishing time difference for the solution of 666.3 and a calculated total distance of 632.7. We can see in the figure that the algorithm can find the Pareto front. Moreover, some objective functions are always above the Pareto front, and in some cases, the algorithm obtains the optimal solution for the objective function. Note that the total difference and total distance for objective functions between w1\_w9 and w8\_w2 are almost identical in most cases of objective function.

We analyzed a total of 56 instances. As shown in Figure 7, the values for each objective function are close to each other. For this reason, we have selected instance C102 to show the relationship between total finishing time difference and total distance for each objective function in Tables 4 and 5, respectively. Tables 4 and 5 show the calculated total finishing time difference and total distance for each objective function solved by our algorithm, with each row in the tables showing the trend when increasing the weight for total finishing time difference. Table 4 shows that, in almost every instance, the highest value is obtained for the total finishing time difference when the weight for the total distance is zero (w0\_w10). Moreover, increasing the weight of total finishing time difference does not result in a decreased value. Conversely, when aiming to minimize the total distance, the distance decreases inversely in proportion to an increase in weight. In general, the solution for a multi-objective problem is not unique; in the end, it needs to be left to the administrator to decide the weights required to convert to a single objective or choose between the Pareto solutions. We can specifically consider the solutions on the Pareto front line to be the best solutions.



**Figure 6.** Plot between total distance and total finishing time differences for 10 solutions and optimal solutions for R101.



**Figure 7.** Plot between total distance and total finishing time differences for 10 solutions and optimal solutions for C102.

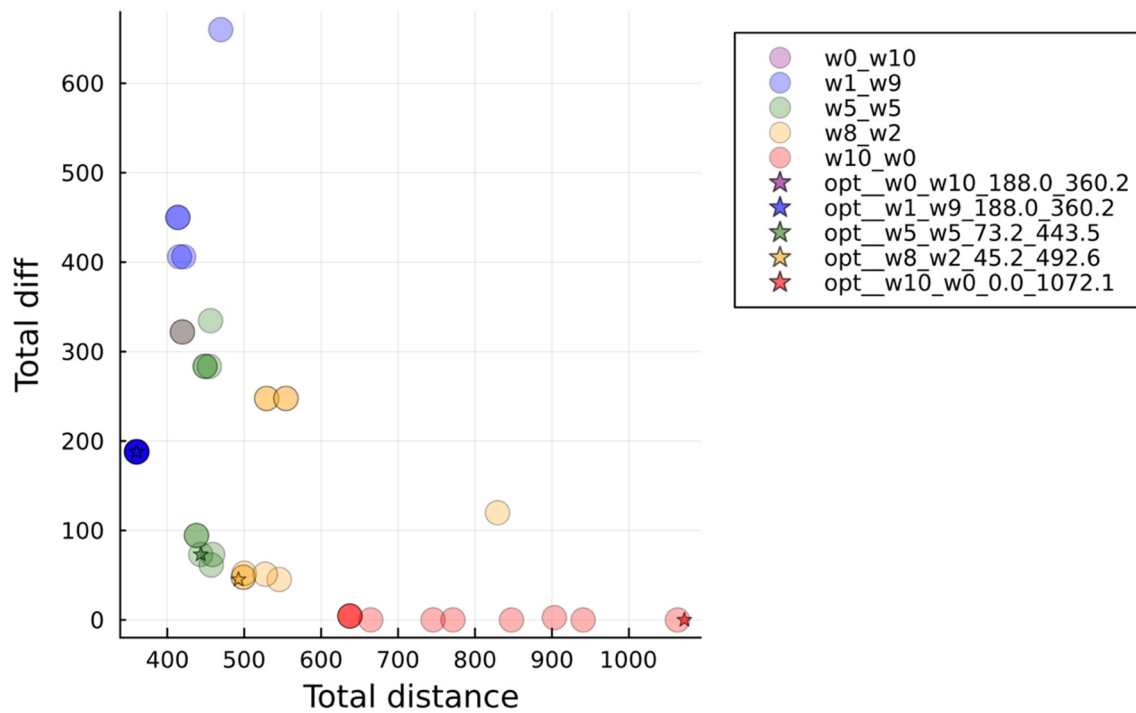


Figure 8. Plot between total distance and total finishing time differences for 10 solutions and optimal solutions for RC201.

Table 4. Total finishing time differences for each objective for instance C102.

w0_w10	w1_w9	w2_w8	w3_w7	w4_w6	w5_w5	w6_w4	w7_w3	w8_w2	w9_w1	w10_w0
819.8	108.4	108.4	72.6	65.0	29.6	121.8	36.5	78.7	63.6	30.9
819.8	74.0	75.0	72.6	64.5	30.7	64.0	26.8	79.8	89.2	89.0
505.6	11.4	42.6	53.6	64.5	130.8	87.7	130.8	29.4	89.2	129.2
819.8	108.4	205.2	65.0	100.8	33.3	29.4	86.5	89.2	87.7	130.4
505.6	108.4	108.4	39.4	202.2	21.8	24.3	64.8	59.2	130.8	81.9
819.8	108.4	108.4	64.5	46.6	42.3	37.2	23.4	23.4	130.8	26.7
517.6	42.3	74.0	64.5	46.6	22.7	90.7	28.9	131.6	129.2	56.8
819.8	74.0	46.6	65.0	64.5	130.8	29.4	258.4	69.2	64.8	20.3
272.2	49.6	75.0	67.7	128.8	87.7	46.6	129.2	130.8	35.9	91.3
819.8	89.2	47.4	46.6	68.2	13.2	87.7	87.7	130.8	88.2	129.2

Table 5. Total distance for each objective function for instance C102.

w0_w10	w1_w9	w2_w8	w3_w7	w4_w6	w5_w5	w6_w4	w7_w3	w8_w2	w9_w1	w10_w0
190.3	249.8	249.8	235.8	237.0	317.0	337.6	284.5	315.6	307.5	303.6
190.3	231.7	232.7	235.8	235.6	293.6	334.0	291.7	393.8	326.4	430.7
191.3	247.6	260.4	251.2	235.6	323.0	304.6	323.0	298.5	326.4	453.5
190.3	249.8	299.3	237.0	253.9	271.1	282.7	316.0	326.4	306.7	395.7
191.3	250.1	259.4	268.1	283.9	271.7	273.8	312.2	310.3	333.4	469
190.3	249.8	249.8	235.6	271.2	271.6	288.0	283.2	283.2	323.0	423.5
194.7	265.0	231.7	235.6	261.6	275.2	300.4	313.1	345.3	338.2	478.6
190.3	231.7	261.6	237.0	235.6	334.0	282.7	263.4	334.9	338.1	417.5
222.2	264.0	232.7	235.1	334.8	308.8	263.6	343.3	323	286.1	395.5
190.3	202.5	262.7	261.6	246.6	264.3	308.8	308.8	323	414.9	445.2

From Figure 5, it is evident that the total difference and total distance may not follow a normal distribution. Moreover, it is hard to guarantee a normal distribution for other benchmark instances when solving with the algorithm. To assess the similarity between

objective functions, the most appropriate test to use is the Kruskal–Wallis test [37]. Under this test, data cannot be assumed to come from any specific distribution. The primary purpose of the test is to determine whether there exists a statistical difference between the medians of distinct groups. In this case, we aim to determine whether the median values of the total difference for each objective function and the total distance between each objective function are statistically significant. Each Solomon’s instance is independently tested with Kruskal test.

The null and alternative hypotheses are determined as follows:

*H0*: There is no significant difference in the total distance or total difference in the objective functions with a 95% confidence interval.

*Ha*: There is a significant difference in the total distance or total difference in the objective functions with a 95% confidence interval.

In this case, the *p*-value for the Kruskal test will accept the hypothesis when the *p*-value is  $< 0.05$ . We separated the test according to the values of the calculated total distance and total finishing time difference and tested the hypothesis grouping by the objective functions. We tested two settings: testing between all objective values, i.e., *w*<sub>0\_w10</sub>, *w*<sub>1\_w9</sub>, . . . , *w*<sub>9\_w1</sub> and *w*<sub>10\_w0</sub>, named Setting1, and testing for partial group of objectives functions, named Setting2. After trying several groups, we chose as the best setting {*w*<sub>0\_w10</sub>, *w*<sub>1\_w9</sub>, *w*<sub>2\_w8</sub>, *w*<sub>8\_w2</sub>, *w*<sub>9\_w1</sub>, *w*<sub>10\_w0</sub>}. The first setting aims to test the similarity between all objective functions while the second setting aims to cut off the objective functions that are similar. The results shown in Table 6 are the *p*-values for each instance and each setting, ignoring the row that has all zero values. We can see that there is no significant difference in total distance and total finishing time difference in Setting1 for each instance (*p*-value  $< 0.05$ ). Moreover, most of the cases in Setting2 are significant except for two, RC103 and RC106 (*p*-value  $> 0.05$ ). The results suggest that solving only the objective functions in Setting2 is enough for use as the Pareto set.

**Table 6.** The *p*-value for each instance of testing for total distance and total finishing time difference.

Instance	Total Distance		Total Finishing Time Difference	
	Setting1	Setting2	Setting1	Setting2
R110	0.00	0.00	0.03	0.01
R112	0.00	0.00	0.01	0.00
R209	0.00	0.00	0.00	0.01
RC103	0.00	0.00	0.02	0.12
RC106	0.00	0.00	0.00	0.06

## 6. Conclusions

This paper investigates multi-objective vehicle routing and scheduling in home health-care systems. Modeling for this problem is based on the vehicle routing problem with time windows. The objectives are to minimize total finishing time difference and minimize the total travel distance, combining typical objective functions for routing and services scheduling problems. Home healthcare has been a crucial support system for the overall health system as it reduces the burden on individual patients who would otherwise have to travel to the hospital regularly to receive care. In real-life situations, minimizing costs alone may not adequately answer the needs of a business, especially those in the healthcare sector. The best way to allocate resources (costs and caretakers) is to consider both aspects simultaneously. For this reason, we have tried to fill the gap in the literature by proposing new objective functions.

We chose the weighted-sum approach to solve multiple-objective optimization problems; and given the NP-hard nature of these problems, an algorithm based on simulated annealing (SA) has been constructed. The problem was converted into a single-objective problem with different choices of weights (a total of 11 choices), then the SA was deployed to solve the problem according to those weights and the results are combined together to obtain the Pareto set. The initial solution was generated using the inserting procedure. Then,

the improvement operators, swapping, moving, path-exchange, and ruin and recreate, are applied to find the improved solution.

To assess the performance of our algorithm, we used it to solve the well-known benchmark instances called Solomon's instances. The results, when compared to the optimal solutions, show that in most cases our algorithm is able to find the optimal solution with the objective function to minimize total finishing time difference. Moreover, the results of testing the median between the objective functions converted from the weighted sum approach suggest that the Pareto set can be obtained by solving only a partial set of objective functions.

In terms of future work, methods should be investigated to solve multiple objective functions based on the idea of transforming the objectives into a single objective by adding weights. Choosing the weight is an important decision to make as the objective values may not be on the same scale. Moreover, to make the problem more realistic, additional constraints may have to be considered, for instance, multiple depots (multiple-health centers), simultaneous visits, and preferences for patients or caretakers. The model can be extended by considering the combination of new objective functions such as finishing time, total traveling time and total number of workloads, etc.

**Author Contributions:** Conceptualization, P.S. and C.L.; Data curation, P.S.; Formal analysis, C.L.; Investigation, P.S.; Methodology, P.S. and C.L.; Software, P.S.; Supervision, C.L.; Validation, P.S. and C.L.; Visualization, C.L.; Writing—original draft, P.S.; Writing—review and editing, C.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by Fundamental Fund 2023 [Grant No. FF66/024], Chiang Mai University.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The original data presented in the study are openly available at <http://web.cba.neu.edu/~msolomon/problems.htm> (accessed on 1 July 2024).

**Acknowledgments:** This research was supported by Chiang Mai University. The authors appreciate Wiriya Sungkhaniyom's proof-reading help.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. United Nations Department of Economic and Social Affairs. *World Population Prospects 2022 of Results*; United Nations Department of Economic and Social Affairs: New York, NY, USA, 2022.
2. Fernandez, A.; Gregory, G.; Hindle, A.; Lee, A.C. A Model for Community Nursing in a Rural County. *J. Oper. Res. Soc.* **1974**, *25*, 231–239. [\[CrossRef\]](#)
3. Hindle, T.; Hindle, A.; Spollen, M. Resource Allocation Modelling for Home-Based Health and Social Care Services in Areas Having Differential Population Density Levels: A Case Study in Northern Ireland. *Health Serv. Manag. Res.* **2000**, *13*, 164–169. [\[CrossRef\]](#)
4. Eveborn, P.; Flisberg, P.; Rönnqvist, M. Laps Care—An Operational System for Staff Planning of Home Care. *Eur. J. Oper. Res.* **2006**, *171*, 962–976. [\[CrossRef\]](#)
5. Bredström, D.; Rönnqvist, M. Combined Vehicle Routing and Scheduling with Temporal Precedence and Synchronization Constraints. *Eur. J. Oper. Res.* **2008**, *191*, 19–31. [\[CrossRef\]](#)
6. Rasmussen, M.S.; Justesen, T.; Dohn, A.; Larsen, J. The Home Care Crew Scheduling Problem: Preference-Based Visit Clustering and Temporal Dependencies. *Eur. J. Oper. Res.* **2012**, *219*, 598–610. [\[CrossRef\]](#)
7. Ait Haddadene, S.R.; Labadie, N.; Prodhon, C. A GRASP × ILS for the Vehicle Routing Problem with Time Windows, Synchronization and Precedence Constraints. *Expert Syst. Appl.* **2016**, *66*, 274–294. [\[CrossRef\]](#)
8. Mankowska, D.S.; Meisel, F.; Bierwirth, C. The Home Health Care Routing and Scheduling Problem with Interdependent Services. *Health Care Manag. Sci.* **2014**, *17*, 15–30. [\[CrossRef\]](#)
9. Fikar, C.; Hirsch, P. Home Health Care Routing and Scheduling: A Review. *Comput. Oper. Res.* **2017**, *77*, 86–95. [\[CrossRef\]](#)
10. Mascolo, M.D.; Martinez, C.; Espinouse, M.-L. Routing and Scheduling in Home Health Care: A Literature Survey and Bibliometric Analysis. *Comput. Ind. Eng.* **2021**, *158*, 107255. [\[CrossRef\]](#)



11. Paolo, T.; Daniele, V. *The Vehicle Routing Problem*; Toth, P., Vigo, D., Eds.; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2002; ISBN 978-0-89871-498-2.
12. Akjiratikarl, C.; Yenradee, P.; Drake, P.R. PSO-Based Algorithm for Home Care Worker Scheduling in the UK. *Comput. Ind. Eng.* **2007**, *53*, 559–583. [[CrossRef](#)]
13. Dohn, A.; Kolind, E.; Clausen, J. The Manpower Allocation Problem with Time Windows and Job-Teaming Constraints: A Branch-and-Price Approach. *Comput. Oper. Res.* **2009**, *36*, 1145–1157. [[CrossRef](#)]
14. Dantzig, G.B.; Wolfe, P. Decomposition Principle for Linear Programs. *Oper. Res.* **1960**, *8*, 101–111. [[CrossRef](#)]
15. Allaoua, H.; Borne, S.; Létocart, L.; Wolfler Calvo, R. A Matheuristic Approach for Solving a Home Health Care Problem. *Electron. Notes Discret. Math.* **2013**, *41*, 471–478. [[CrossRef](#)]
16. Saksuriya, P.; Likasiri, C. Vehicle Routing Problem with Time Windows to Minimize Total Completion Time in Home Healthcare Systems. *Mathematics* **2023**, *11*, 4846. [[CrossRef](#)]
17. Trautsamwieser, A.; Gronalt, M.; Hirsch, P. Securing Home Health Care in Times of Natural Disasters. *OR Spectr.* **2011**, *33*, 787–813. [[CrossRef](#)]
18. Yuan, B.; Liu, R.; Jiang, Z. A Branch-and-Price Algorithm for the Home Health Care Scheduling and Routing Problem with Stochastic Service Times and Skill Requirements. *Int. J. Prod. Res.* **2015**, *53*, 7450–7464. [[CrossRef](#)]
19. de Aguiar, A.R.P.; Ramos, T.R.P.; Gomes, M.I. Home Care Routing and Scheduling Problem with Teams' Synchronization. *Socioecon. Plann. Sci.* **2023**, *86*, 101503. [[CrossRef](#)]
20. Braekers, K.; Hartl, R.F.; Parragh, S.N.; Tricoire, F. A Bi-Objective Home Care Scheduling Problem: Analyzing the Trade-off between Costs and Client Inconvenience. *Eur. J. Oper. Res.* **2016**, *248*, 428–443. [[CrossRef](#)]
21. Niakan, F.; Rahimi, M. A Multi-Objective Healthcare Inventory Routing Problem; a Fuzzy Possibilistic Approach. *Transp. Res. E Logist. Transp. Rev.* **2015**, *80*, 74–94. [[CrossRef](#)]
22. Fathollahi-Fard, A.M.; Hajiaghayi-Keshteli, M.; Tavakkoli-Moghaddam, R. A Bi-Objective Green Home Health Care Routing Problem. *J. Clean. Prod.* **2018**, *200*, 423–443. [[CrossRef](#)]
23. Bertels, S.; Fahle, T. A Hybrid Setup for a Hybrid Scenario: Combining Heuristics for the Home Health Care Problem. *Comput. Oper. Res.* **2006**, *33*, 2866–2890. [[CrossRef](#)]
24. Maya Duque, P.A.; Castro, M.; Sörensen, K.; Goos, P. Home Care Service Planning. The Case of Landelijke Thuiszorg. *Eur. J. Oper. Res.* **2015**, *243*, 292–301. [[CrossRef](#)]
25. Çakırgil, S.; Yücel, E.; Kuyzu, G. An Integrated Solution Approach for Multi-Objective, Multi-Skill Workforce Scheduling and Routing Problems. *Comput. Oper. Res.* **2020**, *118*, 104908. [[CrossRef](#)]
26. Kordi, G.; Divsalar, A.; Emami, S. Multi-Objective Home Health Care Routing: A Variable Neighborhood Search Method. *Optim. Lett.* **2023**, *17*, 2257–2298. [[CrossRef](#)]
27. Cui, H.; Qiu, J.; Cao, J.; Guo, M.; Chen, X.; Gorbachev, S. Route Optimization in Township Logistics Distribution Considering Customer Satisfaction Based on Adaptive Genetic Algorithm. *Math. Comput. Simul.* **2023**, *204*, 28–42. [[CrossRef](#)]
28. Cissé, M.; Yalçındağ, S.; Kergosien, Y.; Şahin, E.; Lenté, C.; Matta, A. OR Problems Related to Home Health Care: A Review of Relevant Routing and Scheduling Problems. *Oper. Res. Health Care* **2017**, *13–14*, 1–12. [[CrossRef](#)]
29. Kirkpatrick, S.; Gelatt, C.D.; Vecchi, M.P. Optimization by Simulated Annealing. *Science* **1983**, *220*, 671–680. [[CrossRef](#)]
30. Holland John Henry. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*; MIT Press: Cambridge, MA, USA; University of Michigan Press: Ann Arbor, MI, USA, 1992.
31. Schrimpf, G.; Schneider, J.; Stamm-Wilbrandt, H.; Dueck, G. Record Breaking Optimization Results Using the Ruin and Recreate Principle. *J. Comput. Phys.* **2000**, *159*, 139–171. [[CrossRef](#)]
32. Solomon, M.M.; Desrosiers, J. Survey Paper—Time Window Constrained Routing and Scheduling Problems. *Transp. Sci.* **1988**, *22*, 1–13. [[CrossRef](#)]
33. Solomon, M.M. Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints. *Oper. Res.* **1987**, *35*, 254–265. [[CrossRef](#)]
34. Bezanson, J.; Karpinski, S.; Shah, V.B.; Edelman, A. Julia: A Fast Dynamic Language for Technical Computing. *arXiv* **2012**, arXiv:1209.5145. [[CrossRef](#)]
35. Bezanson, J.; Edelman, A.; Karpinski, S.; Shah, V.B. Julia: A Fresh Approach to Numerical Computing. *SIAM Rev.* **2017**, *59*, 65–98. [[CrossRef](#)]
36. Gurobi Optimization, LLC. Available online: <http://www.gurobi.com/> (accessed on 13 June 2024).
37. Kruskal, W.H.; Wallis, W.A. Use of Ranks in One-Criterion Variance Analysis. *J. Am. Stat. Assoc.* **1952**, *47*, 583–621. [[CrossRef](#)]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.