A branch and price algorithm for the resource constrained home health care vehicle routing problem ☆

Neda Tanoumand and Tonguç Ünlüyurt[1]

Sabanci University, Orhanli, Tuzla, 34956 Istanbul

Abstract

We consider the vehicle routing problem with resource constraints motivated by a home health care application. We propose a branch and price algorithm to solve the problem. In our problem, we consider different types of patients that require a nurse or a health aid or both. The patients can be serviced by the appropriate vehicles that may carry a nurse, or a health aid or both. The number of nurses and health aids are limited. We also need to satisfy time window constraints for each patient. We try to find feasible routes to minimize the total distance travelled. Our proposed branch and price scheme utilizes a label correcting algorithm with ng-relaxation and a heuristic pricing method. We demonstrate the effectiveness of our algorithm on random problem instances that we have generated based on Solomon instances upto 100 patients.

Keywords: Health care, Vehicle routing, Resource constraints, Branch and Price

1. Introduction and Literature Review

Home Health Care (HHC) services have become widespread in recent years due to rapid demand growth. The services include basic medical needs and care for sick and/or elderly people in their homes. Aging of society, increasing number of people suffering from chronic diseases, and increasing health care costs are some factors that cause this ever increasing demand for such services. Efficient management of providing such services has been a critical issue in terms of minimizing costs of companies, increasing the number of people served, and the quality of services with limited resources. Consequently, different versions of the scheduling and routing problems of HHC staff have been studied in the literature from different perspectives in recent years. Many researchers have considered the problem as a variation of Vehicle Routing Problem (VRP) and Vehicle Routing Problem with Time Windows (VRPTW). They try to solve it by using exact or heuristic solution methodologies that are appropriate for VRP problems [1] [2] [3].

Motivated by the interest in scheduling and routing problem of HHC staff, we investigate a variant of VRPTW in which patients may require two types of services and each type of service can be performed

Email address: ntanoumand,tonguc@sabanciuniv.edu (Neda Tanoumand and Tonguç Ünlüyurt)

by a specific set of personnel. In this model, the patients have time windows and the personnel that can perform each type of service is limited. This model was proposed in [4] motivated by a real life case.

We consider a deterministic and single period setting for the problem where the patients and the corresponding time windows are known beforehand. This is compatible with the real life case described in [4] where the patients for the next day are determined until 5:00 pm and during the night the company plans the routes for the following day. The problem is to specify the routes and schedule of the HHC staff for a single day. Therefore, each day the routing problem of next day can be solved and staff can start their itinerary based on the determined schedule. We consider three types of vehicles carrying at most two personnel, an aid, a nurse or both. Each patient must be visited by a vehicle that is compatible with the requested services within the patient's time windows. The goal is to minimize the total distance travelled.

HHC scheduling and routing problems are often motivated by real world cases that can be distinguished based on objective function and constraints. In terms of objective functions, the most frequent ones are minimization of travelling time and cost which can be seen in [1], [2], [5], [6], and [7]. Additionally, there are some studies which consider minimization of the waiting time of the patients and staff in their models [8], [9], [10]. Moreover, minimization of the violations of the time windows and dissatisfaction of patients and staff are also focus point of some studies as an objective function [2], [6], [11], [12]. Beside common objective functions, there are also less frequent ones such as the minimization of the number of uncovered patients [1],the minimization of the over time of the staff [6], and the maximization of the work load balance of the staff [12].

HHC routing and scheduling problem is considered as a variant of VRP and VRPTW in the literature. Therefore, most of the proposed models for the problem contain basic constraints of VRP and VRPTW such as sub-tour elimination, set covering, and time window constraints. Although some authors regard time windows as hard constraints [7], [10], [11], there are some studies which mention them as soft constraints [13], [6].

Motivated by the real life cases, researchers have considered specific settings and incorporated various additional constraints in their models. In some studies, researchers define constraints to build a standard working atmosphere for HHC staff. The working time regulations and limitations of HHC staff are such constraints that can be seen in most of the models [2], [7], [9], [10], [11], [13]. Although in most of the studies these constraints are regarded as hard ones, in [2] the authors consider them as soft constraints and any violations are penalized in the objective function. Mandatory breaks for HHC staff are considered in [5], [9], and [11] where HHC staff must have break for lunch between their visits. Another constraint that can be observed in a few studies is over time of HHC staff. In [6], the authors consider it as a soft constraint, while in [7] it is regarded as hard constraint.

Additionally, in order to increase the service level, qualification level and preference constraints are incorporated in some of the models. In these works, certain patients may require staff of a certain qualification level. In other models, the patients are allowed to choose the staff that will visit them. Examples of such studies are [6], [7], [8], [11], [13]. In most of the studies, less qualified staff can be substituted by high qualified ones which might not be preferable for the high qualified staff. However, such substitutions are penalized in the objective functions [6], [7].

Besides qualification constraints, considering preference of the patients to be visited by the same care-givers can be seen frequently in models [1], [3], [13]. Moreover, there are some studies in the literature which incorporate preferences of HHC staff as soft [11] or hard constraints [2], [6] in the models. Another assumption might be the existence of different modes of transportation in the models. It can be effective for decreasing the waiting time of the patients and minimizing the overall costs. In [9], authors incorporate the use of public transportation in their model which cause a reduction in the overall costs. In [5], authors consider walking routes in addition to the routes traversed by car. Their outcomes illustrated that the utilization of appropriate walking routes decreases the number of required vehicles tremendously.

For solving HHC routing and scheduling problem, some studies developed a decision support system [3], [12]. In [3] they implemented repeated matching algorithm which is more flexible and fast. In [12] the authors introduced a multi-objective HHC problem and developed a decision support tool based on fuzzy evolution technique to tackle the problem. Variable Neighbour Search (VNS) and Adapted Variable Neighbour Search (AVNS) based meta-heuristics are among the common methodologies for solving the problem [2], [7], [8], [11]. To solve the problem in [7], a meta-heuristic is proposed which incorporate a relatively large neighbourhood search into a multi-directional local search framework. There are studies in the literature which solve the problem using Simulated annealing [2], [13] and Tabu Search [9], [13] based heuristics.

Studies using matheuristics and exact methods for solving HHC scheduling and routing problems are rare in the literature. In [5] where the authors attempted to utilize the walking routes a two-stage procedure was developed for solving the problem. In the first stage, they attempted to find possible walking routes, and beneficial ones were chosen by set partitioning. In the second stage, possible transportation routes were generated using extended biased-randomised savings heuristic. Finally, for the optimization purpose and making use of suitable walking routes unified Tabu Search was developed.

In [1], they implemented an efficient branch-and-price algorithm for solving their problem. They incorporated temporal dependencies and carer-visit preferences into the HHC routing problem. Authors also defined novel visit clustering approach based on the preferences of the patients and staff. In the algorithm, temporal precedence constraints were enforced in the branching procedure and visit clusters generated based on the preferences parameters were utilized to solve the pricing sub-problems effectively.

In the literature, there are a few papers which discuss the HHC routing problem considering the different types of services that can be performed by staff with specific skills and scarcity of the staff, although these are important considerations in HHC problems. Furthermore, based on the recent comprehensive literature reviews [14], [15], it can be observed that the HHC literature does not consider scarcity of the staff and different levels of required services simultaneously. Additionally, there are limited number of studies which implement an exact solution methodology to tackle the problem and find the optimal solution. In this study we attempt to fill this gap in the literature; where we develop an effective branch-and-price method for solving a variant of HHC routing problem optimally. We utilize the model proposed by [4] in which there are different types of services that are provided by staff with different skill levels that cannot be substituted. The model is more realistic, because in real world cases there are limited number of care-giver staff. Moreover, in HHC services there are two main service category (medical and assistive) each need special skills, thus, substitution is not preferable. Our extensive computations illustrate the efficiency of the algorithm which is able to solve Solomon based instances up to 100 patients.

The contributions of this work can be summarized as follows.

- We propose a set partitioning formulation for HHC routing problem.
- We develop and implement a branch and price algorithm for the resource constrained vehicle routing problem.
- We boost our algorithm by utilizing effective optimization techniques.
- We show the efficiency of our algorithm by testing it on problem instances (upto 100 patients) that are generated by adapting Solomon instances to our problem.

The remainder of this paper is organized as follows. In Section 2, we define the setting of our problem, and then present a mathematical formulation in Section 3. Section 4 describes different parts of our methodology; it starts with proposing a set partitioning formulation and continues with implementation issues. The analysis of outcomes of the algorithm is provided in Section 5. Finally, concluding remarks and some notes on future researches are mentioned in Section 6.

2. Problem Description

We consider a set of patients each of whom requires a certain type of service. There are two types of personnel (i.e. two different types of resources), to provide the required services and the available number of each type of personnel is limited. We will refer to these different resources as health aids and nurses. Each patient has a predetermined hard time window for their required service. Vehicles, in this setting, carry at most two personnel. As mentioned before, the problem is motivated by a real life case. A private company is contracted by a municipality in İstanbul to provide such services. Since there are two types of personnel and the driver cannot be one of personnel that would provide service, it is possible to carry at most 2 personnel in each vehicle. A feasible tour for a vehicle starts at the health center, visits some patients by considering their time windows, and return to the health center before completion time. All the patients should be served within their time window. There are three types of patients depending on their required services. Type 1 are the patients who require just nursing services, type 2 are the patients need just health aid services, and type 3 are the patients that ask both nursing and health aid services. Additionally, we assume that a health aid provider cannot be substituted by a nurse and vice versa.

Depending on the services that the patient requires, the patient should be visited by a health aid, or a nurse or both. The current practice of the company was to one nurse and one aid in each vehicle. This strategy may cause an inefficient utilization of the resources. By allowing some vehicles to carry only a nurse or only an aid may lead to better solutions in different perspectives. For instance, with this policy an infeasible problem instance may become feasible. In order to use the resources efficiently and overcome some infeasible cases, it has been proposed in [4] to define three types of vehicles. Type 1 vehicles carry just one nurse, type 2 vehicles carry one aid provider, and type 3 vehicles carry a nurse and an aid provider. Note that in this strategy, vehicles type 1 and 2 can provide relevant services for patients type 1 and type 2, respectively. On the other hand, vehicles type 3 can serve all types of the patients

The objective of the model is to minimize the total distance traveled by the vehicles. Practically, this is a relevant objective if the fleet of vehicles is leased and paid by the total distance travelled. There is no explicit limit on the number of vehicles used. On the other hand, the limited number of staff implicitly limits the number of vehicles. We refer to this problem as (long version) Health Care Routing Problem with Time Windows and Resource Constraints (HCRPTWRC).

Essentially, the problem is to find feasible routes with limited number of personnel to visit each patient with the total minimum cost. A feasible route visits a patient in the appropriate time window and is compatible with the type of the patients. In other words, a type 1 route only visits type 1 patients, type 2 route visits type 2 patients and type 3 route can visit any type of patient.

We provide a small example to demonstrate how this new practice may help. Suppose that the company has 2 nurses and 2 health aid provider staff and assume that the patients are as illustrated in Figure 2.1. In the case of current strategy that each vehicle caries one nurse and one aid provider, the company needs two vehicles and tries to solve the routing problem of two vehicles. From Figure 2.2, it can be seen that due to the time windows and the limitation in the number of HHCS staff, all patients cannot be visited; therefore, this problem is infeasible. On the other hand, using the proposed strategy, as it can be observed from Figure 2.3, all patients can be visited in their predetermined time window. Note that if the company can visit all customers with its current practice, this will also be one of the feasible solutions that can be obtained with our strategy. In this case, the proposed strategy may also decrease the total traveled distance of feasible problems.

Figure 2.1: Illustration of a small example with different types of patients and a health center.

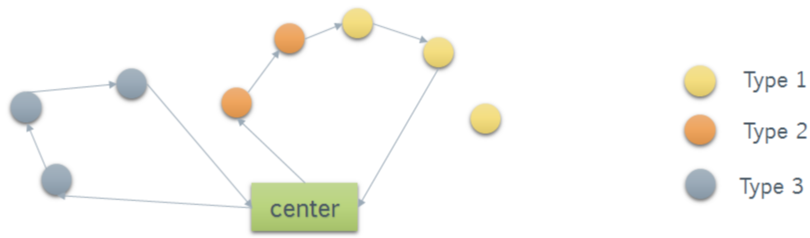

Figure 2.2: Best solution for the small example using current strategy of the company

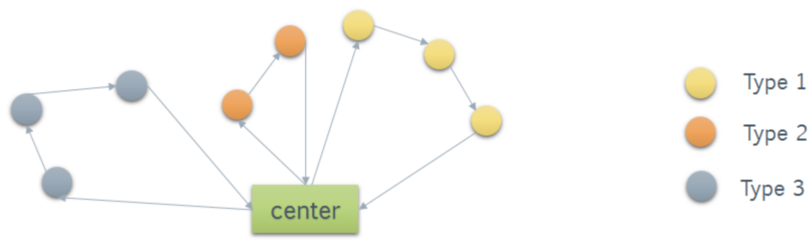

Figure 2.3: A feasible solution for the problem with proposed strategy.

3. Mathematical Formulations

HCRPTWRC can be mathematically formulated as follows. Let $V = \{1, 2, \ldots, N\}$ be the set of patients. Then, we define $V_0 = V \cup \{0\}$, $V_{N+1} = V \cup \{N + 1\}$, and $V_{0,N+1} = V \cup \{0, N + 1\}$ where 0 and $N+1$ are the health centers that each route should starts and ends, respectively. $G = (V_{0,N+1}, A)$ is a complete directed graph with $A = \{(i, j)|i, j \in V_{0,N+1}, i \neq j\}$, set of arcs. Each arc (i, j) has associated distance and travel time that we represent them by d_{ij} and t_{ij}, respectively. $S = \{1, 2, 3\}$ is the set of types of vehicles and $R = \{1, 2\}$ represents the types for the resources (1 represents nurse, and 2 represents health aid). Also, let V^i be the set of patients of type i for $i = 1, 2, 3$ and h_r be the number of available staff type $r \in R$. Furthermore each patient i has a predetermined tight time window $[e_i, l_i]$ and a service time of st_i. Also, note that the time window of the health centers are $[0, T]$ which means that all tours should be completed before T.

Using the parameters that are defined in the previous paragraph, the mathematical formulation of HCRPTWRC can be written as following:

$$minimize \sum_{i\in V_{0,N+1}} \sum_{j\in V_{0,N+1}, j\neq i} \sum_{s\in S} d_{ij} x_{ij}^s \tag{1}$$

$$\text{subject to: } \sum_{i\in V_0, i\neq j} x_{i,j}^1 + \sum_{i\in V_0, i\neq j} x_{i,j}^3 = 1 \qquad \forall j \in V^1 \tag{2}$$

$$\sum_{i\in V_0, i\neq j} x_{i,j}^2 + \sum_{i\in V_0, i\neq j} x_{i,j}^3 = 1 \qquad \forall j \in V^2 \tag{3}$$

$$\sum_{i\in V_0, i\neq j} x_{i,j}^3 = 1 \qquad \forall j \in V^3 \tag{4}$$

$$\sum_{j\in V_{N+1}} x_{0j}^1 + \sum_{j\in V_{N+1}} x_{0j}^3 \leq h_1 \tag{5}$$

$$\sum_{j\in V_{N+1}} x_{0j}^2 + \sum_{j\in V_{N+1}} x_{0j}^3 \leq h_2 \tag{6}$$

$$\sum_{i\in V_0, i\neq j} x_{ij}^s = \sum_{i\in V_{N+1}, i\neq j} x_{ji}^s \qquad \forall j \in V, \forall s \in S \tag{7}$$

$$q_i + x_{ij}^s(t_{ij} + st_i) - T(1 - x_{ij}^s) \leq q_j \qquad \forall i \in V_0, \forall j \in V_{N+1}, j \neq i, \forall s \in S \tag{8}$$

$$e_j \leq q_j \leq l_j \qquad \forall j \in V_{0,N+1} \tag{9}$$

$$x_{ij}^s \in \{0,1\} \qquad \forall i \in V_0, \forall j \in V_{N+1}, j \neq i, \forall s \in S \tag{10}$$

$$q_j \geq 0 \qquad \forall j \in V_{0,N+1} \tag{11}$$

where x_{ij}^s is a binary decision variable indicating whether arc (i, j) is used by vehicle type s or not, also, the other decision variable is q_j which is the arrival time at patient j. The objective function (1) is to minimize the total distance traveled by the vehicles. Constraints (2) – (4) ensure that all patients should be visited only once. Since patients type 1 can be served by vehicles type 1 and type 3, in writing constraint (2) we consider the vehicles of appropriate types. Analogous to type 1 patients, type 2 patients can be served by vehicles type 2 and type 3; therefore, constraint (3) takes into account the vehicles of both types. Constraint (4) just considers vehicles type 3, since patients type 3 can only be served by type 3 vehicles. Constraints (5) and (6) assure that the usage of the nurses and health aid providers is no more than the available staff. Since the problem is a variant of VRPTW, to assure the feasibility and elimination of the sub-tours we need constraints (7) – (9) which are fundamental routing restrictions. Constraint (7) is a balance constraint and constraint (8) is a sub-tour elimination restriction in our problem. Moreover, constraint (9) makes all patients to be visited at their predetermined time windows. At the end, we have binary and non-negativity constraints.

Aforementioned formulation is a mixed integer programming (MIP) formulation of HCRPTWRC. This formulation is originally introduced by [4]. To the best of our knowledge, exact solution methods for solving this problem have not been studied in the literature.

4. Branch-and-price algorithm

In this study, we introduce an exact solution method for solving HCRPTWRC. We adapt the well known branch-and-price algorithm to solve the problem by using the set-partitioning formulation. This formulation can be obtained by applying Dantzig-Wolfe decomposition to the MIP formulation as follows.

4.1. Set Partitioning formulation

Let Ω^s be the set of routes of type $s \in S$ and c_p^s be the cost of path $p \in \Omega^s$ that is traversed by vehicle type $s \in S$, the set partitioning formulation of HCRPTWRC can be stated as following:

$$minimize \sum_{s \in S} \sum_{p \in \Omega^s} c_p^s \theta_p^s \tag{12}$$

$$\text{subject to: } \sum_{p \in \Omega^s} \alpha_{jp}^s \theta_p^s = 1 \qquad \forall s \in S, \forall j \in V^s \tag{13}$$

$$\sum_{s \in S} \sum_{p \in \Omega^s} \beta_{rp}^s \theta_p^s \leq h_r \qquad \forall r \in R \tag{14}$$

$$\theta_p^s \in \mathbb{N} \qquad \forall s \in S, \forall p \in \Omega^s \tag{15}$$

where θ_p^s is the number of times that route p of type s is used. The objective function (12) is the minimization of the total travelled distance. Constraints (13) ensure that each patient must be visited by an appropriate vehicle and constraints (14) enforce the solution to be resource feasible.

To enhance the performance of the column generation, constraints (13) can be replaced by the following constraints:

$$\sum_{p \in \Omega^s} \alpha_{jp}^s \theta_p^s \geq 1 \qquad \forall s \in S, \forall j \in V^s \tag{16}$$

since the problem is a minimization problem and the cost matrix satisfies triangular inequality, the optimal solution still visits all the patients exactly once. This modification makes the dual variables corresponding to this set of constraints become non-negative. This restriction causes the column generation algorithm to converge faster [16]. The modified mathematical formulation along with relaxation of integrality constraints provide *restricted master problem* (RMP).

4.2. Pricing Sub-Problem

The pricing sub-problem of HCRPTWRC should be able to produce a non-basic resource feasible route with the smallest reduced cost. Let γ_j^s and ω_r be the dual variables corresponding to the constraints (13) and (14), respectively; then, the objective function of the sub-problem can be stated as following:

$$\overline{c}_p^s = \sum_{(i,j)\in A^s} (d_{ij} - \gamma_j^s)x_{ij}^s - \sum_{r\in R} \omega_r \beta_{rp}^s \qquad \forall s \in S, \forall p \in \Omega^s \tag{17}$$

The objective function of the sub-problem is the minimization of the expressions in the right hand side of the equation (17). Since, the decision variable, x_{ij}^s, only exists in the first expression, the second expression is a parameter that we call it *dual constant*. For the sake of simplicity, we drop the dual constant from the objective function of the sub-problem, since, it does not affect the optimal solution of the sub-problem. When the optimal solution of the sub-problem is obtained, we use the dual constant to check the optimality condition (whether reduced cost is negative or not). Consequently, the sub-problem of set partitioning formulation of HCRPTWRC can be written as follows:

$$\textit{minimize} \sum_{s\in S}\sum_{(i,j)\in A^s} \overline{c}_{ij}^s x_{ij}^s$$
$$\text{subject to: (7)-(11)}$$

Observing the structure of the sub-problem, it turns out that the sub-problem is decomposable into $|S|$ independent sub-problems as all the constrains are written for each type of vehicles $s \in S$ and the objective is minimizing the summation over all vehicle types. In fact, decomposition helps us to partition the feasible region of the sub-problem into smaller regions which accelerates the solution algorithms. From an algorithmic point of view, it enables us to solve the sub-problems in parallel and each time instead of sending one column with negative reduced cost to the RMP, we can send one column per sub-problem with negative reduced cost (if there exists any). This parallelization provides more convenience and less computational time. Therefore, the mathematical formulation for sub-problem(s) can be written as following:

$$\textit{minimize} \sum_{(i,j)\in A^s} \overline{c}_{ij}^s x_{ij}^s \tag{18}$$

$$\text{subject to:} \sum_{i\in Q_0^s, i\neq j} x_{ij}^s = \sum_{i\in Q_{N+1}^s, i\neq j} x_{ji}^s \qquad \forall j \in Q^s \tag{19}$$

$$q_i + x_{ij}^s(t_{ij} + st_i) - T(1 - x_{ij}^s) \leq q_j \qquad \forall i \in Q_0^s, \forall j \in Q_{N+1}^s, j \neq i \tag{20}$$

$$e_j \leq q_j \leq l_j \qquad \forall j \in Q_{0,N+1}^s \tag{21}$$

$$x_{ij}^s \in \{0,1\} \qquad \forall i \in Q_0^s, \forall j \in Q_{N+1}^s, j \neq i \tag{22}$$

$$q_j \geq 0 \qquad \forall j \in Q_{0,N+1}^s \tag{23}$$

where Q^s is the set of patients that can be served by vehicle type $s \in S$. In our problem, since we have three different types of vehicles, our sub-problem is decomposed into three sub-problems. From the mathematical formulation (18) − (23), it can be observed that every pricing sub-problem is an *elementary shortest path* problem. Note that vehicle type 1 and type 2 can serve the patients type 1 and type 2, respectively; therefore, we can state that $Q^s = V^s$ for $s = 1, 2$. However, this is not true for vehicles type 3. Since these vehicles can serve all types of the patients, the corresponding set Q contains all the patients ($Q^3 = V$).

In order to solve the proposed formulations in a systematic fashion, we proceed the steps of the Branch-and-Price algorithm. The algorithm is a Branch-and-Bound algorithm which embeds column generation into every node of the search tree. In this framework, the linear relaxation of the set partitioning formulation ($\theta_p^s \geq 0$) is solved in each node. Since the enumeration of all routes in $\Omega = \bigcup_{s \in S} \Omega^s$ is typically too time consuming, column generation algorithm dynamically generates high quality routes through sub-problems. In the following section, we will discuss the implementation issues and the techniques that we utilized to improve the performance of the branch-and-price algorithm.

4.3. Implementation

We develop a branch-and-price algorithm to solve HCRPTWRC. There are some factors which affect the efficiency of the algorithm. In this section, we will touch upon these factors and how we implement different components of the algorithm for the problem.

4.3.1. Initial Solution

Initial solution is one of the crucial factors which affects the convergence of the column generation algorithm in each node of the search tree. In the column generation algorithm, at the first step, we need to solve RMP which contains only a subset of decision variables. These variables should be able to create an initial basic feasible solution. Therefore, it is important to choose or create the initial variables such that the initial RMP becomes feasible. There are different ways to generate initial variables and their corresponding coefficient columns.

In this study, we use a heuristic algorithm which is proposed by [17] for constructing initial solution. This approach is based on greedy algorithm and it is quite fast in populating the set of initial columns and generating high quality feasible routes. Since the combination of generated routes might not be feasible in terms of resource levels, we add as many artificial columns as needed with high costs to the set. Each artificial column corresponds to a route starts from a health center, visits a particular patient and finally returns to the health centre without any resource consumption. The combination of the routes generated by heuristic and artificial routes produces a feasible solution for the problem. Therefore, final set of routes can be used as an initial feasible solution for our Branch-and-price algorithm.

4.3.2. Solving the sub-problem

Solving the sub-problem is a very crucial part in the implementation of branch-and-price algorithm. Since, it affects the performance of the algorithm directly, using efficient techniques for solving the sub-problems becomes more important. To this end, there are many methods proposed in the literature for solving the sub-problems. We adapt the label correcting algorithm for our problem as described next.

Label Correcting Algorithm

In our problem which is a variant of VRPTW, the pricing sub-problem is an *Elementary Shortest Path* problem [16]. One of the algorithms proposed in the literature is *Label Correcting Algorithm* which is proposed by [18]. This algorithm is appropriate for solving *elementary shortest path problem with resource constraints.* Label correcting algorithm is a dynamic programming approach that tries to find an optimal path from source node s to sink node t by considering the resource limitations. In this algorithm, every label represents a partial path and contains elements which correspond to cost and resource consumption of the partial path. Additionally, for adapting label correcting algorithm for elementary shortest path problems, binary node-visit resources must be considered by the labels. This algorithm starts with a trivial label (with all elements equal to 0) from the source node and tries to extend this label along an arc if the extension is feasible. The extension is infeasible, if the resulting partial path is not resource-feasible or if it cannot reach to the sink node because of the resource limitations.

The performance of the label correcting algorithm depends mostly on the dominance rule that eliminates partial paths. Let P'_i and P^*_i be two distinct partial paths from source to node i. Also, let (C', R') and (C^*, R^*) be the corresponding labels, where, C' and C^* are costs, and R' and R^* are resource consumption of P'_i and P^*_i, respectively. Path P'_i dominates path P^*_i if and only if $(C', R') \neq (C^*, R^*)$, $C' \leq C^*$, and $R'_k \leq R^*_k$ for $\forall k \in K$ where $|K|$ is the number of resources.

To strengthen the dominance relation and consequently eliminate more partial paths, Feillet et al. [18] introduced the concept of *unreachable nodes* and proposed a new definition for the labels. For a path P_i from the source node to a node i, a node v is said to be *unreachable* if it has already been visited by P_i, or it cannot be visited because of the resource limitation. Using this notation, the description of the labels can be modified. In the new definition, node-visit resource of a particular node in a label is consumed when the node is *unreachable* for the partial path corresponding to the label.

In our implementation, we define a label, extension of a label, and the dominance relation as follows.

Definition of a label. A label corresponding to a partial path from source node 0 to node i can be defined as $\Lambda_i = (\theta, \tau, \Gamma, P, \rho)$, where, θ represents the reduced cost of the partial path, τ shows the beginning time of the service at node i, $\Gamma = \{\gamma_1, \gamma_2, \ldots, \gamma_N\}$ is a visit-vector representing availability of node-visit resource of each node, P tracks the sequence of the nodes in the partial path, and ρ is the cost of the route.

Label extension. Extension of a label $\Lambda_i = (\theta, \tau, \Gamma, P, \rho)$ from node i to node j creates label $\Lambda_j = (\theta', \tau', \Gamma', P', \rho')$, where:

- $\theta' = \theta + \bar{c}_{ij}$
- $\tau' = max\{e_j, \tau + st_i + t_{ij}\}$
- $\Gamma' = \{\gamma_k = 1$ if k is an *unreachable* node for label Λ_j; 0 otherwise$\}$
- $P' = P \cup \{j\}$
- $\rho' = \rho + c_{ij}$

Note that τ' which represents the starting time of the service at node j, cannot occur before the specified time window e_j. Therefore, if a service provider reaches the patient earlier that its time window, s/he should wait to start service at e_j. Furthermore, the definition of an *unreachable* node is the same as proposed description. Specifically, node k is an *unreachable* node for label Λ_j if:

- node k has been already visited via partial path corresponding to label Λ_j,
- Λ_j cannot extend to node k because of the time restriction.
- extension of the resulting label, Λ_k, to the health center exceeds the completion time, T.

Dominance relation. Let $\Lambda_i = (\theta, \tau, \Gamma, P, \rho)$ and $\Lambda'_i = (\theta', \tau', \Gamma', P', \rho')$ be two label on node i. Label Λ_i dominates Λ'_i if:

- $\theta \leq \theta'$
- $\tau \leq \tau'$
- $\gamma_r \leq \gamma'_r$, $\forall r \in V$

This algorithm performs quite well in the small size instances; however, in medium and large instances the performance of the algorithm deteriorates. To overcome this, a relaxed version of the sub-problem can be solved which is much easier to handle.

ng-route relaxation

As mentioned before, in this work, the sub-problem of Branch-and-price algorithm is a *resource constrained elementary shortest path problem*(RCESPP). Due to difficulty and complexity of *RCESPP*, it has been proposed in the literature to solve the relaxed version of sub-problems [19]. Although these sub-problems might generate paths containing cycles, they are easier to solve. Such a sub-problem can be incorporated in the structure of the Branch-and-price algorithm; then, through the steps of the algorithm the columns containing cycles can be eliminated.

In this study, we implement *ng-route* relaxation which has been proposed by [20]. In this relaxation, for every node i, we associate a neighbourhood $V_i \subset V$ which contains i and its "closest" neighbours. In this case, the sub-problems are allowed to generate the routes containing cycles of the form $i - \cdots - j - \cdots - i$ where $i \notin V_j$. Although the reduced cost of the column containing such a cycle might be less than others, its distance will be much more than others. Since i is not in the neighbourhood of j, having such a cycle leads to a costly tour which cannot be a part of optimal solution of Branch-and-price algorithm; so, the validity of the optimal solution remains.

Let us consider the extension of a label $\Lambda_i = (\theta, \tau, \Gamma, P, \rho)$ from node i to node j that creates label $\Lambda_j = (\theta', \tau', \Gamma', P', \rho')$. To incorporate *ng-route* relaxation into *label correcting* algorithm, we make slight modifications in the definition of unreachable node. In this method, node k is unreachable for label Λ_j in the following situations:

- if node k is visited by label Λ_j and $k \in V_j$;
- if label Λ_j cannot visit node k before its latest time, and
- if reaching to the health center after node k exceeds the completion time T.

As it can be inferred, the only modification is in the first part where label Λ_j can visit node k again, if node k is not in the neighbourhood of node j. Regarding the modifications of the other components, they are updated as mentioned in the previous section.

Note that in our experiments, we set the size of the neighbourhoods to be 5. That is $|V_j| = 5$ for all $j \in V$. This modification helps the dominance relations to eliminate more labels. Consequently, the number of labels associated with one node decreases and the algorithm is computationally improved. Along with performance enhancement, using *ng-route* relaxation confirms the validity of the optimal solution.

Heuristic Pricing

Instead of returning the column with the most negative reduced cost in each iteration of the column generation, we can return any column with negative reduced cost (if one exists). This strategy may cause an increase in the number of iterations of the column generation and decrease in the improvement of the objective function in each iteration. However, the reduction in the computation time caused by this strategy is promising. The computational time of each pricing problem decreases drastically, and this causes a reduction in the overall computational time.

To incorporate the above idea into our algorithm, we define a heuristic pricing in which the labels associated with each node are limited to 10. In each iteration of the *label correcting* algorithm, the labels on each node is sorted based on their reduced cost, then 10 labels which have the smaller reduced cost for each node are picked and the algorithm proceeds with the limited labels. The heuristic pricing algorithm is quick so that in a few seconds it can return columns with negative reduced cost. When the heuristic pricing cannot produce any column with negative reduced cost, then we run the exact *label correcting* algorithm.

Another strategy to merge the idea with the column generation steps is to import all columns with negative reduced cost that are found at every iteration to RMP. Our observations show that the strategy populates the column pool fast and decreases the number of iterations of the column generation; consequently, it enhances the performance of the Branch-and-price algorithm by reducing the overall computational time.

4.3.3. Branching Scheme

Branching strategy is another crucial point of the branch-and-price algorithm. When the optimal solution of the RMP is fractional, we need to perform branching to achieve an integer solution. There are several strategies for branching i.e. branching on fractional variable of master problem (θ_p^s), variable of the original formulation (x_{ij}^s), and number of vehicles. Among those techniques, branching on the variables of the original formulation is more common and accepted to be more convenient [16].

In branching on the variables of the original formulation, we need to select an arc (i, j) with a fractional flow, $0 < f_{ij} < 1$. Then, we obtain two branches: in one branch we forbid arc (i, j) to be a part of the solution ($f_{ij} = 0$), in another branch we enforce arc (i, j) to be a part of the solution ($f_{ij} = 1$). The flow of an arc can be calculated using solution values of the master problem. It is computed by summing values of those decision variables which correspond a path containing arc (i, j). Our strategy for selecting an arc is to choose arc (i, j) which its flow is closest to 0.5.

Once branching is performed, two children nodes are generated and in each child node the column generation algorithm should be performed compatible with the branching decision. This requires modifications in the master problem and the sub-problems of the child nodes. To modify master problem, in each child node, we simply discard all the columns from master problem and rebuild it with the columns of the parent node which are compatible with the branching decision and artificial identity columns (columns that are associated with visiting just one patient without any resource consumption).

The next step is to make the sub-problems compatible with the branching decision which is much complicated in comparison to the master problem adaptation. We modify sub-problems of the child nodes as follows. To enforce an arc (i, j) to be a part of solution, we discard all the arcs (i, l) with $l \neq j$ and (l, j) with $i \neq l$ from the pricing sub-problems. This modification makes the sub-problems generate paths which visit node j only after node i. Also, consider that in the master problem we have the restriction of visiting all nodes at least one time. The modification along with the restriction enforce arc (i, j) to be a part of the optimal solution. On the other hand, to forbid an arc (i, j) to be a part of solution, we simple discard this arc from the sub-problems. Note that since we have three sub-problems we need to modify all of them in every node of the search tree. For more detail on branching scheme see [16].

5. Computational Study

In this section, we briefly introduce our instance pool, then, we will evaluate the performance of the algorithm as well as the effects of different strategies that are applied.

All computations were performed on a Windows 7 professional 64-bit operating system with an Intel Core i7-4770 CPU with 3.40 GHz and 16.0 GB RAM. We compare the outputs of the proposed branch-and-price algorithm against solving original MIP using CPLEX 12.6.

5.1. Instance Generation

In our computational experiment, we utilized modified version of the instances generated by [4]. The instances are adapted version of the benchmark Solomon VRPTW instances which have 25, 50, and 100 nodes.

To be more compatible with the real world applications, we do not prefer to use Solomon instances with clustered properties and/or long time windows. Therefore, we select our base instances from Solomon R1-type and RC1-type categories. Two data sets from each of the categories are chosen randomly; thus, for all our instances with 25, 50, and 100 patients, we utilize Solomon's R103, R108, RC101, and RC105 as our base cases. From these data the information regarding coordinates and the time windows are obtained. Then, to adapt them to HHC routing problem settings, it is required to assign a service type and service time to each patient.

In our model, there are three possible services that we can randomly assign to each patient and generate various versions of base cases. For populating our instance pool systematically, we define three different groups, G1, G2 and G3 which are generated based on the properties presented in Table 5.1. For example, 60%, 48%, and 48% of the patients in G1, G2, and G3, respectively, require type 1 services. For each group, we generate a specific version of each base case such that percentage of the patients in the new generated instances are compatible with the group properties.

Table 5.1: Percentage of each service type in different groups

	Service type		
Groups	1	2	3
G1	60%	32%	8%
G2	48%	36%	16%
G3	48%	28%	24%

The service time of each patient is related to the type of the service that is requested. In all our instances, the service times of type 1, type 2, and type 3 patients are 10, 40, and 45 minutes, respectively.

Other important data that should be determined are the number of available nurses and health aid providers. This part is an arduous task, since, the problem can become infeasible if the number of resources are too tight. On the other hand, if the resources are too loose, then, resource constraints will become redundant.

For instances with 25 patients, based on the tightness and looseness of the resource parameters, 4 different variants of each instance can be produced. where the resource level for the nurse is loose and for the aid provider is tight, the instance is called loose-tight; in a similarly way, we can name other variants tight-loose, tight-tight, and loose-loose.

For obtaining loose and tight levels of the resources, our proposed model for HHC routing problem $((1) - (11))$ is solved optimally by CPLEX under different objective functions. The optimal solution obtained by solving the problem using minimization of the resources as the objective function is considered as tight resource levels. On the other hand, loose resources are achieved by solving the problem with the objective of distance minimization when resource constraints are relaxed.

Defining tight and loose resource levels using discussed strategy is valid just for small instances. Using this strategy, we are not able to define valid resource levels for instances with 50 and 100 patients. This is because CPLEX cannot solve the model optimally when the number of patients is high. Therefore, for each instances with 50 and 100 patients we solve the LP relaxation of the model by column generation algorithm where the objective function is distance minimization. Then, we use the outcome to define loose and tight resource levels. It turns out that in most of the instances the outcomes of the algorithm is too tight that the instances become infeasible. In this cases, we increase the resource levels one by one to find out the tightest resource level. For the instances with 50 and 100 patients we just define tight and loose variants of each instance. Ultimately, we have 48, 24, and 24 different variations of Solomon based instances with 25, 50, and 100 patients, respectively.

For more information on the instances the interested reader can be referred to Appendix A. In the following section, we represent and discuss the results obtained from implementation of the branch-and-price algorithm on the generated instances.

5.2. Results and Discussion

In this section, we will evaluate the performance of the branch-and-price algorithm that is boosted by the several techniques that were discussed in section 4. The algorithm is implemented in Java using the branch-and-price framework of Java OR Library (Jorlib). For solving RMP, CPLEX 12.6 is utilized. The time limit for instances including 25 patients is 2 hours and for instances with 50 and 100 patients is 6 hours. For calculating optimality gap we use the following formula:

$$Gap(\%) = \frac{upperbound - lowerbound}{lowerbound} \times 100$$

The results of our numerical experiments are summarized in tables 5.2-5.4. In these tables we briefly report the average optimality gap and percentage of optimally solved instances of enhanced branch-and-price algorithm and CPLEX for Solomon based instances. Each row of the tables correspond to the instances with 20, 50, and 100 patients that are labeled in similar way.

As mentioned earlier, we obtained our instance pool from Solomon's R-type and RC-type categories. In the tables, we use "R-type" and "RC-type" labels for former and later ones respectively, in order to distinguish the data. Also, for conciseness we refer enhanced branch-and-price as "BAP".

For Solomon based instances with 25 patients the average optimality gap of enhanced branch-and-price is presented in table 5.2. Based on the outputs, all 48 instances with 25 patients are solved optimally within the time limit. The average running time of our algorithm is 6.55 seconds, where the average running time of the instances is 1.76 seconds if we exclude an instance with 231.63 seconds running time.

Whereas, CPLEX is not able to solve optimally two R-type instances with medium and tight resource levels within the time limit based on the result observations. Also, table 5.2 illustrates that the average optimality gap of CPLEX for these two instances is 1.43%. Excluding two feasibly solved instances that their running times hit the time limit, CPLEX solves other 46 instances optimally on average within 239.38 seconds.

It turns out that the enhanced branch-and-price performs quite well in the Solomon based instances with 50 patients. Based on the results presented in table 5.3 our algorithm solved 87.50% of the instances optimally and the total optimality gap is 0.60%. The running time for the optimally solved instances with 50 patients is on average 1266.74 seconds, where the average running time of 58% of the instances is 9.86 seconds.

Based on the tables 5.2 and 5.3, CPLEX solves 50% of RC-type instances optimally within on average 7.43 seconds, and other RC-type instances feasibly with 1.32% average optimality gap. Whereas, 25% of the R-type instances are solved feasibly with the average gap of 44.52%. More than 58% of the instances with 50 patients could not be solved by CPLEX due to either memory or time restriction. It can be observed that CPLEX fails to solve instances with 50 patients efficiently, especially for R-type instances the performance is fairly weak.

Enhanced branch-and-price algorithm optimally solves 58.33% of the RC-type and 16.67% of the R-type instances with 100 patients. Although in total only 37.50% of all 100-instances are solved optimally, the average optimality gap is almost 2.11%. Additionally, the average running time of the optimally solved instances is 5198.95 seconds which seems quite promising.

For instances with 100 patients, CPLEX could not perform efficiently. The results are presented in the tables 5.2 and 5.3. Based on the tables, CPLEX solves non of R-type instances optimally, and the average optimality gap of the ones which are solved feasibly is 88.89%. The results for RC-type instances are better than R-type ones. Where 33.33% of RC-type instances containing 100 patients are solved optimally within 1310.30 seconds. The average optimality gap of CPLEX for RC-type instances

Table 5.2: Average gap for Solomon based instances

Inst.	BAP Gap (%)			CPLEX Gap (%)		
	R-type	RC-type	Total	R-type	RC-type	Total
25	0.00	0.00	0.00	1.43	0.00	0.72
50	0.72	0.47	0.60	44.52	1.32	14.28
100	3.07	1.15	2.11	88.89	10.84	28.19

Table 5.3: Percentage of the optimally solved Solomon based instances

Inst.	BAP (%)			CPLEX (%)		
	R-type	RC-type	Total	R-type	RC-type	Total
25	100.00	100.00	100.00	91.67	100.00	95.83
50	83.33	91.67	87.50	0.00	50.00	25.00
100	16.67	58.33	37.50	0.00	33.33	16.67

which are solved either feasibly or optimally is 10.84%. As considering all instances with 100 patients, CPLEX is able to optimally solve 16.67% of them; furthermore, the optimality gap as reported in the table 5.2 is 28.19%.

It is worthy to mention that, in our computational experiments, there are instances that both enhanced branch-and-price algorithm and CPLEX could not find an incumbent solution within our time limits. When calculating the average optimality gap of groups of instances that are reported in table 5.2, these instances are not taken into the account.

Our numerical experiment reveals that both enhanced branch-and-price algorithm and CPLEX spend much time and effort for solving R-type Solomon based instances than RC-type ones. Additionally, it is observed that among R-type instances the ones obtained from modification of R108 are more laborious, since, both solvers faced with CPU and memory problems when tackling them.

Additionally, we examined the effects of resource levels on the performance of the enhanced branch-and-price algorithm and CPLEX and it is observed that there is no explicit relationship. In some cases, instances with tight resource levels are more time consuming, whereas, in other cases, instances with loose or medium resource levels are hard to solve.

The detailed results for each problem instance are presented in Appendix B for interested readers..

6. Concluding Remarks

Home health care services have been very popular in recent years. The special requirements of problems give rise to different problem settings. In this work, we consider one such problem where

Table 5.4: Status of BAP and CPLEX for Solomon based instances

Inst.	Total	BAP			CPLEX		
		Opt.	Feas.	n. incom.	Opt.	Feas.	n. incom.
25	48	48	0	0	46	2	0
50	24	21	3	0	6	4	14
100	24	9	15	0	4	5	15

the patients may require different types of services. These services are provided with limited amount of resources. In our setting, the resources are the nurses and health aids and they are transported to patients in a certain way. In this work, we develop an effective branch-and-price algorithm to minimize total distance travelled while providing the required service to each patient in the specified time window.

From a modeling perspective, deterministic travel times and demand and tight time windows are the main assumptions of the model. There is work in the literature regarding stochastic travel times for other types of vehicle routing problems. One may try to adapt these to our setting. In terms of demand, it may be worthwhile to consider a version of the problem where the new demand may arrive during the day. Depending on the condition and status of the patients, some of the time windows may be considered as soft time windows. From an algorithmic perspective, it is worthwhile to develop effective heuristic algorithms for large problems.

7. Acknowledgements

We would like to show our gratitude to Dr. Bülent Çatay and Dr. Gizem Özbaygın for comments that greatly improved the manuscript. We would also like to thank Nozir Shokirov for sharing his test instances and corresponding heuristic solutions.

Appendix A.

The characteristics of the instances with 25 patients are demonstrated in Table A.1. First column shows the groups that each instance belongs to them. They are determined by using the information of Table 5.1. Second and third columns are names and IDs of the instances. Following column is the best found objective function value of the instances (OFV). Next two columns refer to the available number of nurses and health aid providers for each instances. Based on the available resources, the service levels are determined in the next column; L_L, T_T, and medium mention to the loose, tight, and medium level resources. Finally, the last column is the completion time (CT) of the problem in which all vehicles visiting the patients should return to the health center.

In Table A.2, the first column illustrates the groups that each instance belongs to them. These groups are determined based on the percentage of the Type 3 patients. Next two columns are the names and the IDs of the instances. The number of available nurses and health aid providers are mentioned in the next two columns. Finally, the completion time of each instance is demonstrated in the last column.

Table A.1: Characteristics of the instances with 25 patients

				Resources			
Group	Instance	ID	OFV	#N	# aid	Level	CT(s)
G1	R103-25-8-1	1	51480	6	6	L_L	230
	R103-25-8-2	2	56781	5	4	T_T	230
	R103-25-8-3	3	55113	5	5	medium	230
	R103-25-8-4	4	53946	6	4	medium	230
G2	R103-25-16-1	5	56022	7	7	L_L	230
	R103-25-16-2	6	61753	5	6	T_T	230
	R103-25-16-3	7	61710	5	7	medium	230
	R103-25-16-4	8	59195	6	6	medium	230
G3	R103-25-24-1	9	57707	7	7	L_L	230
	R103-25-24-2	10	57707	6	5	T_T	230
	R103-25-24-3	11	59391	6	6	medium	230
	R103-25-24-4	12	61248	7	5	medium	230
G1	R108-25-8-1	13	44775	5	5	L_L	230
	R108-25-8-2	14	48450	4	4	T_T	230
	R108-25-8-3	15	45671	4	5	medium	230
	R108-25-8-4	16	48450	5	4	medium	230
G2	R108-25-16-1	17	48710	6	6	L_L	230
	R108-25-16-2	18	48909	5	6	T_T	230
	R108-25-16-3	19	48909	5	7	medium	230
	R108-25-16-4	20	48710	7	6	medium	230
G3	R108-25-24-1	21	56022	7	7	L_L	230
	R108-25-24-2	22	49709	5	6	T_T	230
	R108-25-24-3	23	49709	5	7	medium	230
	R108-25-24-4	24	47704	7	6	medium	230
G1	RC101-25-8-1	25	62139	6	6	L_L	240
	RC101-25-8-2	26	69228	4	5	T_T	240
	RC101-25-8-3	27	63861	4	6	medium	240
	RC101-25-8-4	28	67703	5	5	medium	240
G2	RC101-25-16-1	29	68237	6	8	L_L	240
	RC101-25-16-2	30	80357	4	7	T_T	240
	RC101-25-16-3	31	70340	4	8	medium	240
	RC101-25-16-4	32	70220	5	7	medium	240
G3	RC101-25-24-1	33	67028	7	7	L_L	240
	RC101-25-24-2	34	70611	5	7	T_T	240
	RC101-25-24-3	35	82493	6	6	medium	240
	RC101-25-24-4	36	67790	6	7	medium	240
G1	RC105-25-8-1	37	61270	6	7	L_L	240
	RC105-25-8-2	38	65560	4	6	T_T	240
	RC105-25-8-3	39	64073	4	7	medium	240
	RC105-25-8-4	40	63747	5	6	medium	240
G2	RC105-25-16-1	41	64301	6	7	L_L	240
	RC105-25-16-2	42	69024	5	5	T_T	240
	RC105-25-16-3	43	67062	5	6	medium	240
	RC105-25-16-4	44	67263	6	5	medium	240
G3	RC105-25-24-1	45	67572	7	7	L_L	240
	RC105-25-24-2	46	73099	5	6	T_T	240
	RC105-25-24-3	47	71198	5	7	medium	240
	RC105-25-24-4	48	69346	6	6	medium	240

Table A.2: Characteristics of the instances with 50 patients

Group	Instance	ID	OFV	Resources #N	# aid	Level	CT(s)
G1	R103-50-8-1	1	95437	9	9	L_L	230
	R103-50-8-2	2	105036	8	7	T_T	230
G2	R103-50-16-1	3	109884	8	11	L_L	230
	R103-50-16-2	4	118443	7	10	T_T	230
G3	R103-50-24-1	5	109891	10	12	L_L	230
	R103-50-24-2	6	117006	9	11	T_T	230
G1	R108-50-8-1	7	77169	12	14	L_L	230
	R108-50-8-2	8	90006	6	7	T_T	230
G2	R108-50-16-1	9	83301	10	12	L_L	230
	R108-50-16-2	10	88999	7	10	T_T	230
G3	R108-50-24-1	11	83849	9	11	L_L	230
	R108-50-24-2	12	85257	8	10	T_T	230
G1	RC101-50-8-1	13	122772	10	14	L_L	240
	RC101-50-8-2	14	141659	8	11	T_T	240
G2	RC101-50-16-1	15	143670	13	16	L_L	240
	RC101-50-16-2	16	178719	10	13	T_T	240
G3	RC101-50-24-1	17	128398	13	15	L_L	240
	RC101-50-24-2	18	145113	10	12	T_T	240
G1	RC105-50-8-1	19	119737	10	14	L_L	240
	RC105-50-8-2	20	137389	7	11	T_T	240
G2	RC105-50-16-1	21	123889	12	13	L_L	240
	RC105-50-16-2	22	141547	9	10	T_T	240
G3	RC105-50-24-1	23	130882	11	13	L_L	240
	RC105-50-24-2	24	155584	10	11	T_T	240

Table A.3: Characteristics of the instances with 100 patients

Group	Instance	ID	OFV	Resources #N	# aid	Level	CT(s)
G1	R103-100-8-1	1	152623	18	20	L_L	230
	R103-100-8-2	2	175560	13	13	T_T	230
G2	R103-100-16-1	3	168935	15	21	L_L	230
	R103-100-16-2	4	195374	13	19	T_T	230
G3	R103-100-24-1	5	169204	35	45	L_L	230
	R103-100-24-2	6	169161	25	28	T_T	230
G1	R108-100-8-1	7	131747	20	24	L_L	230
	R108-100-8-2	8	131243	16	18	T_T	230
G2	R108-100-16-1	9	139626	18	23	L_L	230
	R108-100-16-2	10	149524	13	18	T_T	230
G3	R108-100-24-1	11	151089	15	20	L_L	230
	R108-100-24-2	12	155079	12	17	T_T	230
G1	RC101-100-8-1	13	212550	25	30	L_L	240
	RC101-100-8-2	14	227444	18	20	T_T	240
G2	RC101-100-16-1	15	227897	25	32	L_L	240
	RC101-100-16-2	16	232243	20	27	T_T	240
G3	RC101-100-24-1	17	231311	20	26	L_L	240
	RC101-100-24-2	18	270470	17	23	T_T	240
G1	RC105-100-8-1	19	213820	15	22	L_L	240
	RC105-100-8-2	20	245907	12	19	T_T	240
G2	RC105-100-16-1	21	218237	27	30	L_L	240
	RC105-100-16-2	22	218237	25	28	T_T	240
G3	RC105-100-24-1	23	227086	20	22	L_L	240
	RC105-100-24-2	24	242795	18	20	T_T	240

Appendix B.

The detailed results of both enhanced-branch-and-price algorithm and CPLEX are presented in tables B.1-B.3 for the instances containing 25, 50, and 100 patients.

The labels and results in the tables are as follows. The instances in these tables are labeled by their ID numbers. In the first row of the tables instance ID's and enhanced branch-and-price algorithm are abbreviated as Inst. ID and BAP, respectively. In the tables, columns 2-4 report the outcomes of our enhanced branch-and-price algorithm; where columns 5-7 illustrate the results of CPLEX.

The numbers in the columns labeled as 'Status' depict the final condition of the solvers within time limits; 0, 1, and 2 are representative of the conditions that a solver solves a specific instance optimally, feasibly, or the solver cannot find an incumbent solution, respectively.

The columns labeled as 'GAP(%)' report the percent optimality gap of the solvers for each instance. For the instances that the solvers could not find an incumbent solution the optimality gap is depicted as a dash.

The columns with 'CPU(s)' labels indicate the running time of the solvers in seconds for each instances. The characters 'M' and 'C' in front of some instances under the mentioned columns respectively state that the solvers face with memory error and/or CPU error during the run. Additionally, character 'T' is written in front of the instances that the solvers hit time limit when tackling them.

Table B.1: Solvers' detailed results for Solomon based instances with 25 patients

Inst. ID	BAP			CPLEX		
	Status	Gap(%)	CPU(s)	Status	Gap(%)	CPU(s)
1	0	0	1.97	0	0	9.11
2	0	0	1.26	0	0	134.79
3	0	0	3.11	0	0	99.90
4	0	0	0.68	0	0	35.92
5	0	0	0.26	0	0	7.35
6	0	0	0.77	0	0	1194.61
7	0	0	3.34	0	0	143.28
8	0	0	1.32	0	0	41.97
9	0	0	0.23	0	0	16.11
10	0	0	0.23	0	0	5860.24
11	0	0	0.63	0	0	35.96
12	0	0	0.78	0	0	1711.50
13	0	0	0.93	0	0	72.76
14	0	0	2.38	1	25.57	T
15	0	0	2.03	0	0	235.79
16	0	0	10.53	1	8.80	T
17	0	0	0.61	0	0	147.65
18	0	0	0.64	0	0	165.98
19	0	0	0.60	0	0	146.44
20	0	0	0.60	0	0	128.78
21	0	0	0.28	0	0	7.41
22	0	0	8.40	0	0	420.67
23	0	0	9.07	0	0	272.00
24	0	0	0.80	0	0	36.97
25	0	0	0.35	0	0	0.27
26	0	0	0.10	0	0	0.05
27	0	0	0.10	0	0	0.04
28	0	0	0.10	0	0	0.05
29	0	0	0.34	0	0	0.05
30	0	0	0.12	0	0	0.03
31	0	0	0.10	0	0	0.03
32	0	0	0.24	0	0	0.14
33	0	0	0.32	0	0	0.11
34	0	0	0.21	0	0	0.06
35	0	0	0.10	0	0	0.13
36	0	0	0.17	0	0	0.09
37	0	0	1.78	0	0	4.39
38	0	0	0.16	0	0	5.61
39	0	0	0.27	0	0	7.01
40	0	0	0.16	0	0	6.22
41	0	0	12.64	0	0	14.38
42	0	0	0.24	0	0	6.5
43	0	0	5.58	0	0	15.18
44	0	0	1.86	0	0	4.62
45	0	0	231.63	0	0	3.81
46	0	0	1.87	0	0	5.92
47	0	0	1.80	0	0	7.76
48	0	0	2.89	0	0	3.75

Table B.2: Solvers' detailed results for Solomon based instances with 50 patients

Inst. ID	BAP			CPLEX		
	Status	Gap(%)	CPU(s)	Status	Gap(%)	CPU(s)
1	0	0	15.91	1	31.04	9,11
2	0	0	13.13	2	-	T
3	0	0	312.40	2	-	M
4	0	0	10.97	2	-	M
5	0	0	11.86	2	-	M
6	0	0	5.19	2	-	M
7	1	3.85	T	1	43.37	M
8	0	0	9359.13	2	-	M
9	0	0	14777.61	1	59.16	T
10	1	4.77	T	2	-	M
11	0	0	1113.69	2	-	M
12	0	0	69.51	2	-	M
13	0	0	26.70	0	0	10.46
14	0	0	5.56	0	0	18.46
15	0	0	3.56	0	0	2.51
16	0	0	2.39	0	0	4.85
17	0	0	2.84	0	0	1.25
18	0	0	5.77	0	0	7.06
19	0	0	156.67	1	9.25	M
20	0	0	674.55	2	-	M
21	1	5.69	T	2	-	M
22	0	0	18.38	2	-	M
23	0	0	9.61	2	-	M
24	0	0	6.16	2	-	M

Table B.3: Solvers' detailed results for Solomon based instances with 100 patients

Inst. ID	BAP			CPLEX		
	Status	Gap(%)	CPU(s)	Status	Gap(%)	CPU(s)
1	1	1.71	T	2	-	T
2	0	0	3611.30	2	-	T
3	1	7.93	T	2	-	T
4	0	0	1000.13	2	-	C
5	1	0.82	T	2	-	C
6	1	0.79	T	1	58.55	T
7	1	3.86	T	1	119.23	T
8	1	4.31	T	2	-	T
9	1	6.25	T	2	-	M
10	1	1.89	T	2	-	T
11	1	4.89	T	2	-	C
12	1	4.41	T	2	-	C
13	0	0	4948.89	0	0	344.76
14	0	0	8965.50	1	2.99	T
15	1	2.21	T	0	0	275.36
16	0	0	4806.74	0	0	3925.54
17	0	0	1690.67	0	0	695.53
18	0	0	4348.00	2	-	C
19	1	3.52	T	2	-	C
20	0	0	466.01	2	-	C
21	0	0	16953.24	1	36.97	T
22	1	2.01	T	1	35.95	T
23	1	3.47	T	2	-	T
24	1	2.56	T	2	-	T

References

[1] M. S. Rasmussen, T. Justesen, A. Dohn, J. Larsen, The home care crew scheduling problem: Preference-based visit clustering and temporal dependencies, European Journal of Operational Research 219 (3) (2012) 598–610.

[2] G. Hiermann, M. Prandtstetter, A. Rendl, J. Puchinger, G. R. Raidl, Metaheuristics for solving a multimodal home-healthcare scheduling problem, Central European Journal of Operations Research 23 (1) (2015) 89–113.

[3] P. Eveborn, P. Flisberg, M. Rönnqvist, Laps care—an operational system for staff planning of home care, European Journal of Operational Research 171 (3) (2006) 962 – 976, feature Cluster: Heuristic and Stochastic Methods in Optimization Feature Cluster: New Opportunities for Operations Research. doi:https://doi.org/10.1016/j.ejor.2005.01.011.
URL `http://www.sciencedirect.com/science/article/pii/S0377221705001207`

[4] B. Tozlu, R. Daldal, T. Ünlüyurt, B. Çatay, Crew constrained home care routing problem with time windows, in: Computational Intelligence, 2015 IEEE Symposium Series on, IEEE, 2015, pp. 1751–1757.

[5] C. Fikar, P. Hirsch, A matheuristic for routing real-world home service transport systems facilitating walking, Journal of Cleaner Production 105 (Supplement C) (2015) 300 – 310, decision-support models and tools for helping to make real progress to more sustainable societies. doi:https://doi.org/10.1016/j.jclepro.2014.07.013.
URL `http://www.sciencedirect.com/science/article/pii/S0959652614007045`

[6] M. Mısır, P. Smet, G. Vanden Berghe, An analysis of generalised heuristics for vehicle routing and personnel rostering problems, Journal of the Operational Research Society 66 (5) (2015) 858–870. doi:10.1057/jors.2014.11.
URL `https://doi.org/10.1057/jors.2014.11`

[7] K. Braekers, R. F. Hartl, S. N. Parragh, F. Tricoire, A bi-objective home care scheduling problem: Analyzing the trade-off between costs and client inconvenience, European Journal of Operational Research 248 (2) (2016) 428 – 443. doi:https://doi.org/10.1016/j.ejor.2015.07.028.
URL `http://www.sciencedirect.com/science/article/pii/S037722171500661X`

[8] D. S. Mankowska, F. Meisel, C. Bierwirth, The home health care routing and scheduling problem with interdependent services, Health Care Management Science 17 (1) (2014) 15–30. doi:10.1007/s10729-013-9243-1.
URL `https://doi.org/10.1007/s10729-013-9243-1`

[9] K.-D. Rest, P. Hirsch, Daily scheduling of home health care services using time-dependent public transport, Flexible Services and Manufacturing Journal 28 (3) (2016) 495–525. doi:10.1007/s10696-015-9227-1.
URL https://doi.org/10.1007/s10696-015-9227-1

[10] M. Di Mascolo, M.-L. Espinouse, C. E. Ozkan, Synchronization between human resources in home health care context, in: Proceedings of the International Conference on Health Care Systems Engineering, Springer, 2014, pp. 73–86.

[11] A. Trautsamwieser, P. Hirsch, Optimization of daily scheduling for home health care services, Journal of Applied Operational Research 3 (3) (2011) 124–136.

[12] M. Mutingi, C. Mbohwa, Multi-objective homecare worker scheduling: A fuzzy simulated evolution algorithm approach, IIE Transactions on Healthcare Systems Engineering 4 (4) (2014) 209–216. arXiv:http://dx.doi.org/10.1080/19488300.2014.966213, doi:10.1080/19488300.2014.966213.
URL http://dx.doi.org/10.1080/19488300.2014.966213

[13] S. Bertels, T. Fahle, A hybrid setup for a hybrid scenario: combining heuristics for the home health care problem, Computers and Operations Research 33 (10) (2006) 2866 – 2890, part Special Issue: Constraint Programming. doi:https://doi.org/10.1016/j.cor.2005.01.015.
URL http://www.sciencedirect.com/science/article/pii/S030505480500016X

[14] C. Fikar, P. Hirsch, Home health care routing and scheduling: A review, Computers and Operations Research 77 (2017) 86–95.

[15] D. C. Paraskevopoulos, G. Laporte, P. P. Repoussis, C. D. Tarantilis, Resource constrained routing and scheduling: Review and research prospects, European Journal of Operational Research.

[16] D. Feillet, A tutorial on column generation and branch-and-price for vehicle routing problems, 4OR 8 (4) (2010) 407–424. doi:10.1007/s10288-010-0130-z.
URL https://doi.org/10.1007/s10288-010-0130-z

[17] N. Shokirov, A variable neighborhood search approach for solving the crew constrained home care routing problem with time windows, Master's thesis, Sabanci University, Turkey (2003).

[18] D. Feillet, P. Dejax, M. Gendreau, C. Gueguen, An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems, Networks 44 (3) (2004) 216–229.

[19] G. Desaulniers, O. B. Madsen, S. Ropke, Chapter 5: The vehicle routing problem with time windows, in: Vehicle Routing: Problems, Methods, and Applications, Second Edition, SIAM, 2014, pp. 119–159.

[20] R. Baldacci, A. Mingozzi, R. Roberti, New route relaxation and pricing strategies for the vehicle routing problem, Operations research 59 (5) (2011) 1269–1283.