



Cumulative VRP with Time Windows: A Trade-Off Analysis

Alejandro Fernández Gil¹(✉), Mariam Gómez Sánchez¹, Eduardo Lalla-Ruiz²,
and Carlos Castro¹

¹ Departamento de Informática, Universidad Técnica Federico Santa María,
Valparaíso, Chile

{affernan,mggomez}@jp.inf.utfsm.cl, Carlos.Castro@inf.utfsm.cl

² Department of Business Information Systems and Industrial Engineering,
University of Twente, Enschede, The Netherlands
e.a.lalla@utwente.nl

Abstract. In this work, the Cumulative Vehicle Routing Problem (CumVRP) is studied. It is a routing optimization problem, in which the objective is to construct a set of vehicle routes with the minimum cumulative cost in terms of distance and weight over a traveled arc. The CumVRP can be defined with hard and soft time windows constraints for incorporating customer service. To tackle this problem, a matheuristic approach based on combining mathematical programming and an iterative metaheuristic algorithm Greedy Randomized Adaptive Search Procedure (GRASP) is proposed. In each step of our approach, a feasible solution (set of routes) is built using GRASP, and, afterward, the solution is optimized using a MILP optimizer. The main objective of this research is to analyze the trade-off between the environmental cost produced by the delivery of goods complying with the limits of time windows and the customer's dissatisfaction when these limits are violated at a certain time limit previously defined. The results show that the environmental cost is reduced if the violation of the upper limits of the customers' time windows is allowed. These violations generate a cost associated with penalties that are well balanced with respect to the reduction of emissions.

Keywords: Cumulative Vehicle Routing Problem · Green VRP · Time windows · Matheuristic · GRASP · MILP

1 Introduction

Nowadays, there is increasing concern at enterprise and social levels concerning greenhouse gas emissions. The transportation companies seek to offer a good service to the customers while taking into account solutions that tackle climate change. This circumstance has been widely studied in the area of Green Logistics (GL), which considers a set of activities that propose measuring the environmental impact of different product distribution strategies, minimizing the

energy usage in logistics activities, reducing waste and managing its treatment [1]. Currently, burning fossil fuels causes 80% of environmental pollution in the world, and around 60% of this fuel is generated by the freight transport sector [2]. Optimizing fuel consumption helps reduce the rate of environmental pollution and health impacts. Moreover, fuel consumption by a vehicle is affected by several factors, e.g., distance traveled, weight and speed of the vehicle and traffic, among others [3]. The inclusion of these factors in optimization models increases the realism of the solutions but, on the other hand, increases their complexity.

One of the main optimization problems in logistics to model freight distribution is the well-known Vehicle Routing Problem (VRP, [4]). It is commonly used for modeling transportation and logistics scenarios while optimizing a given objective function, e.g., distance, traveling time, etc. A recent VRP variation considering the reduction of the *cumulative cost*, calculated based on the distance traveled by a vehicle carrying a certain load, is the Cumulative VRP (CumVRP, [5,6]), which is an \mathcal{NP} -Hard problem. CumVRP belongs to the problems within Green VRP area as it considers the reduction of fuel consumption [3,7]. Another relevant aspect studied in the VRP related literature is the inclusion of time window constraints such as in the VRP with time windows (VRP-TW), where customers must be served within a predefined time interval [8,9]. When considering time windows, it can be distinguished between hard and soft time windows. In the hard time windows case, a vehicle must serve the customers right at a specified time interval, if the vehicle arrives earlier than the time window, it has to wait, late arrival is not allowed. The soft time window case permits the violation of the time window constraints subject to some penalty [10].

Considering previous discussion, in this work, we focus on analysing the inclusion of time windows on the CumVRP. Firstly, we consider the case where the freights distribution is constrained by hard time windows, and secondly, the case where the service time is constrained by soft time windows so that time window violations are permitted subject to penalization, i.e., the service is allowed to begin outside the time windows at the cost of a given penalty. Depending on the application, the use of soft time windows can reflect practical situations much better than hard time windows [11].

The goal of including time windows into the CumVRP aims at analyzing the trade-off between the environmental cost produced by the transportation of goods to customers, complying with the intervals of the time windows and infringement of quality service to the customers. Investigating on soft time windows permits analyzing the contribution of soft time windows penalty and environmental related costs. Furthermore, to provide feasible solutions for these problem variants, this work proposes a hybrid approach, specifically a matheuristic (named *MathGRASP*). *MathGRASP* is based on combining a mathematical programming approach (MILP optimizer) within the Greedy Randomized Adaptive Search Procedure (GRASP) metaheuristic.

The remainder of the paper is organized as follows. Section 2 reviews related work. Section 3 describes the mathematical formulation of the CumVRP and

CumVRP with hard and soft time windows. The solution approach based on metaheuristic algorithm is given in Sect. 4. Computational experiments and results are given in Sect. 5 and finally, we present the conclusions and future work in Sect. 6.

2 Related Works

In the freight transportation industry, the importance of achieving optimal vehicle routing considering sustainability factors is growing [12]. The area of Green Vehicle Routing Problems (Green VRPs) is characterized by incorporating environmental features (e.g., emissions per type of vehicle, energy minimization, etc.). Solution approaches in this area aim at proposing effective routes to satisfy the environmental concerns and commercial indicators of enterprises and societies. An extensive survey on this can be found in the work of Lin et al. [13].

In recent years, the Cumulative Vehicle Routing Problem (CumVRP) has received considerable attention from researchers. CumVRP has been introduced by Kara et al. [5] and referred to as a linear model of fuel consumption [7], and also as an energy minimization model [6]. The objective of the CumVRP is to find a set of routes that minimize the total cumulative cost, calculated based on the distance traveled by a vehicle transporting a specific weight load. In [3], it is stated that CumVRP is a type of problem that belongs to the area of Green VRPs, and in the study proposed in [7], some variations are suggested where the models consider factors aimed at reducing fuel consumption.

Several algorithms and approaches have been designed and developed for solving the CumVRP. Regarding exact algorithms, in [14] a mathematical model for CumVRP is presented as a Set Covering Problem and solved by a Column Generation Algorithm (CGA). However, the exact proposed algorithms demonstrate solving a considerable number of instances optimally, but always require a high amount of computational time and excessive memory consumption to solve more complex ones. For large instances, it is known that this type of approaches are not efficient enough to obtain the global optimum in reasonable times [15].

Heuristic and metaheuristic algorithms are able to find good quality solutions in relatively small times [16]. These types of algorithms have shown promise in solving VRPs: Clarke and Wright Savings algorithm (C&W) has been used in [17] to address CumVRP with Limited Duration; a Simulated Annealing (SA) is proposed in [18] for solving the previous problem considering the multi-trip factor and minimize fuel consumption; in [19], a memetic heuristic is reported that is aimed at solving CCVRP; an Iterated Greedy procedure is presented in [20] to achieve larger instances of CCVRP, and the proposed metaheuristic proves to be competitive.

3 Cumulative VRP

In this work, we address the Cumulative Vehicle Routing Problem (CumVRP) proposed in [5]. The goal of the CumVRP is to reduce *cumulative costs*, calcu-

lated based on the distance traveled by a vehicle carrying a certain load. The following parameters are defined in the problem:

- V , set of customer nodes.
- A , set of edges or $arc (i, j) \in A$ between each pairs of customers $i, j \in V$.
- d_{ij} , representing the travel distance between each $arc (i, j) \in A$.
- k , number of vehicles.
- ω , curb weight without associated weight load for each vehicle.
- Q , the maximum weight load for each vehicle.
- R , the flow capacity of the arcs of the network.

Let $G = (V, A)$ be a directed graph where each node $i \in V$ represents a customer and each $arc (i, j) \in A$ the distance between two customers $i, j \in V$. The node 0 represents the depot vertex from where a homogeneous fleet of k vehicles depart to satisfy the demand of the $|V| - 1$ customers. For each vehicle, its curb weight ω is known and the maximum weight load that it can carry is Q . The flow capacity of each arc of the network R is known (maximal value of the flow in any arc of the network, for example, curb weight plus the capacity of the vehicles). For each customer $i > 0$, there is a demand $Q \geq Q_i > 0$. The cumulative cost of moving a vehicle from a customer i to a customer j is defined as $w_{ij} \cdot d_{ij}$. The decision variables are as follows:

- $x_{ij} \in \{0, 1\}$, $\forall i, j \in V$, set to 1 if the $arc (i, j)$ is in the tour of a vehicle, and 0 otherwise.
- w_{ij} , $\forall i, j \in V$, the flow on the $arc (i, j)$ if the vehicle goes from i to j , and 0 otherwise.
- c_{ij} , $\forall i, j \in V$, the cost of traversing an $arc (i, j)$, is defined as $w_{ij} \cdot d_{ij}$.

The objective function of CumVRP (1) is to find a set of routes of minimum total cost where the cost is defined as the product of the distance of the $arc (i, j)$ and flow on this arc.

$$Min \sum_{i=0}^V \sum_{j=0}^V d_{ij} w_{ij} \quad (1)$$

In the CumVRP the following constraints are considered:

- (a) Each customer has to be served exactly by one vehicle.
- (b) Each route starts and ends at the depot.
- (c) For each tour, the flow on the arcs accumulate as much as preceding node's supply in the case of collection or diminish as much as preceding node's demand in the case of delivery.
- (d) The flow on any arc of each tour does not exceed the flow capacity of the arcs.
- (e) For each customer, the demand required must be satisfied.

In this research, we investigate the additions of time windows constraints in the CumVRP. The following two variants of CumVRP are evaluated: (i)

CumVRP-hTW, where the time windows cannot be violated, and (ii) CumVRP-sTW, where the upper bound of the time windows can be violated at a certain delay threshold.

The objective when solving CumVRP-hTW is based on obtaining a set of subtours, in such a way that the cumulative cost added to the number of vehicles used to supply the total demand is minimized. In the case of CumVRP-sTW, we will minimize the cumulative cost added to the penalties for service delay and use of vehicles. Both variants will be studied as delivery cases. Two variants are proposed in the following subsections.

3.1 CumVRP with Hard Time Windows

The addition of the hard time windows constraints to the CumVRP allows establishing strict frames of time to indicate when the customer has to be served. To properly include such features, the following parameters have to be incorporated:

Parameters:

- $L_i, \forall i \in V$, lower bound of the time window from which the customer i must be served.
- $U_i, \forall i \in V$, upper bound of the time window until which the customer i must be served.
- $S_i, \forall i \in V$, service time to serve customer i .
- M , big enough value.

Using the above parameters, the time interval $[L_i, U_i]$ represents the time windows during which the customer i must be served.

Decision variable:

t_i : time at which customer i starts to be serviced.

Objective function:

$$f_1 = \sum_{i=0}^V \sum_{j=0}^V d_{ij} w_{ij} \tag{2}$$

$$f_2 = Mk \tag{3}$$

$$\text{Min } f_1 + f_2 \tag{4}$$

In the CumVRP-hTW, besides the constraints presented for CumVRP from (a) to (d), the following ones have to be considered for the correct consideration of the time windows:

- (f) The starting time for servicing a customer $i \in V$ has to be equal to or greater than L_i time windows lower bound.
- (g) The end service time of a customer i cannot exceed the U_i time window bound.

3.2 CumVRP with Soft Time Windows

Adding soft time windows constraints to the CumVRP allows establishing customers' time windows where its upper limit U_i does not have to be necessarily satisfied, in such a case a penalization is applied. Thus, it is necessary to add the parameters μ and P representing the allowed delay threshold and the penalty applied to each time unit of delay, respectively. In addition, it is necessary to include the following decision variables:

- $y_i \in \{0, 1\}, \forall i \in V$, set to 1 if not complied with the upper limit for the time window in the customer i , and 0 otherwise.

The objective function, besides considering f_1 and f_2 as defined in the model for CumVRP-hTW, must incorporate the cost associated with penalties for violations (5), so it must be redefined as:

$$f_3 = \sum_{i=1}^V (P(t_i + S_i - U_i))y_i \tag{5}$$

$$\text{Min } \lambda_1 f_1 + f_2 + \lambda_3 f_3 \tag{6}$$

The previous constraints from (a) to (f) remain in this formulation. The constraint (g) is redefined as:

- (h) The end service time of a customer $i \in V$, U_i , cannot be exceeded by more than μ , i.e., $U_i + \mu$.

Finally, with regards to the depot, there is a time windows upper limit, U_0 , associated with it, therefore, the following constraint has to be incorporated:

- (i) The depot upper time windows U_i cannot be violated.

4 A *MathGRASP* Approach for the CumVRP

Matheuristics are algorithms that have been used successfully in routing problems [21]. These algorithms are based on combining heuristic or metaheuristic schemes and mathematical programming models to obtain high quality solutions. There are various applications of these techniques to solve variants of VRPs, such as time windows [22], multi-depot [23], electric vehicle [24], and cross-docking [25]. A matheuristic for the Pollution Routing Problem (PRP) is presented in [26], where a metaheuristic based on local search is combined with a MILP model. In [23] Multi-Depot CCVRP is solved using the POPMUSIC matheuristic algorithm (see [27]). This matheuristic consists in dividing a large problem into parts and then several of those parts are seen as subproblems to be solved through an exact approach.

For solving the CumVRP with time windows, we present a matheuristic based approach termed as *MathGRASP*. It is based on the hybridization of the Greedy Randomized Adaptive Search Procedure (GRASP) with an solving to

optimality the optimization model. Our metaheuristic is described in the context of a GRASP metaheuristic (see Algorithm 1). The GRASP algorithm [28] is an iterative method where each iteration has two phases: (i) construction and (ii) local search (lines 5–6).

Algorithm 1. *MathGRASP* Pseudocode

```

Input:  $C$  vehicles,  $V$  customers,  $rcl$ ,  $Max\_Iter$ 
1:  $\mathcal{T}_F = \emptyset$ ;
2: Preprocessing();
3: while ( $Max\_Iter > 0$ ) do
4:    $\mathcal{T} = \emptyset$ ;
5:    $\mathcal{T} = \text{Construction\_Algorithm}(rcl, V, C)$ ;
6:    $\mathcal{T}^* = \text{Local\_Search}(\mathcal{T})$ ;
7:   if ( $f(\mathcal{T}^*) < f(\mathcal{T}_F)$ ) then
8:      $\mathcal{T}_F = \mathcal{T}^*$ ;
9:   end if
10:   $Max\_Iter --1$ ;
11: end while
12: return  $\mathcal{T}_F$ ;

```

Initially a preprocessing of data (line 2) is performed. The constructive algorithm is provided with the set of vehicles and customers, in addition to the parameter rcl (size of restricted candidate list) and provides feasible solutions consisting of a set of subtours ($\tau_i \in \mathcal{T}$) (line 5). After construction procedure, the best solution is selected and a local search procedure is applied to it (line 6). Finally, the algorithm updates the best tour found (lines 7-9) and returns the best solution obtained in all iterations \mathcal{T}_F (line 12). In the following subsections we describe the components of the algorithm.

Solution Representation. The solution of the problem is stored in \mathcal{T}_F , represented by a matrix, where each row represents a subtour τ_i , and in each subtour there is a sequence of ($Customers \in \tau_i$) to visit (see Fig. 1).

$$\mathcal{T}_F = \begin{matrix} \tau_1 \\ \tau_2 \\ \vdots \\ \tau_n \end{matrix} \begin{bmatrix} Depot & Customer_9 & Customer_8 & \dots & Depot \\ Depot & Customer_3 & Customer_7 & \dots & Depot \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ Depot & Customer_6 & Customer_5 & \dots & Depot \end{bmatrix}$$

Fig. 1. Solution structure composed of subtours.

Construction Algorithm. In each iteration of Algorithm 2, a feasible solution to the problem is constructed. It aims at making as few subtours as possible.

Each subtour considers the depot as the starting node. To create a subtour at each iteration, the array of customers is sorted in ascending order with respect to the distance between each customer and the last customer added to the subtour (initially the depot) (line 4). A restricted candidate list of customers RCL is constructed with rcl size, and on it the first RCL customer from V is added (line 5–6). After that, a customer is randomly selected from the restricted candidate list (line 7) and takes into account the available capacity of the vehicle, the candidate is added to the corresponding subtour (lines 8–12). Once the maximum capacity of the vehicle is completed, the variables are updated to form a new subtour (line 13). The procedure is performed while there are customers without visiting (line 3). Once the subtours have been established, the route to be carried out complying either with the hard or soft time windows is locally optimized (procedure described in the following subsection) and each subtour is updated (line 16). Finally, a feasible solution to the problem is returned (line 17).

Algorithm 2. Construction_Algorithm(rcl, V, C) Pseudocode

```

1:  $\mathcal{T} = \emptyset$ ;
2:  $c =$  Choose the initial Vehicle of  $C$ ;
3: while (! $V.empty()$ ) do
4:   Sort array of Customer  $V$ ;
5:    $Ub = \min(rcl, |V|)$ ;
6:    $RCL = V[0 : Ub]$ ;
7:    $cand = RCL[random(1, |RCL|)]$ ;
8:   if ( $c.Current\_Load + cand.Demand \leq c.Maximum\_Payload$ ) then
9:     Add  $cand$  to  $\mathcal{T}$  at  $c.index$ ;
10:    Update  $Current\_Load$ ;
11:    Delete the  $cand$  from  $V$ ;
12:   else
13:      $c =$  Choose the next Vehicle of  $C$ ;
14:   end if
15: end while
16:  $\mathcal{T} = \text{Solve\_MILP}(\mathcal{T})$ ; ▷ see Exact solution subsection
17: return  $\mathcal{T}$ ;

```

- **Exact solution:** The model proposed in [5] incorporating the time windows constraints as discussed in Subsect. 3.1 are reformulated to optimize each subtour obtained by (Algorithm 2, lines 1–16). In both cases, to cover the total demand of each subtour, only one vehicle will be necessary, since the sum of the customers' demand considered each time is less than the maximum load supported by the vehicle. After completing the previous step in both formulations, the decision variable k is eliminated and the new assumption is added:

(j) For each subtour, only one loop can be performed.

In both cases, it is not necessary to assess the penalty for using vehicles in the objective function, therefore the Eqs. (4) and (6) are reformulated as the Eqs. (7) and (8) for CumVRP-hTW and CumVRP-sTW respectively.

$$\text{Min } f_1 \tag{7}$$

$$\text{Min } \lambda_1 f_1 + \lambda_3 f_3 \tag{8}$$

Local Search Algorithm. The Local Search (LS) algorithm is based on Hill-Climbing (HC) algorithm with the first improvement strategy. The pseudocode of the algorithm is shown in Algorithm 3. The used Hill-Climbing algorithm works with the best feasible solutions constructed by the Algorithm 2. The defined movement is inspired by [29] where two subtours are randomly selected (lines 3–4). After that, two customers are selected so that the distance between them is minimal (line 5) and a swap is performed (lines 6–7). Subsequently, the vehicle capacity constraints are checked in each subtour and if these were satisfied, the modified subtours are again optimized (line 9), and \mathcal{T} is updated with the best solution found (lines 10–12). Finally, after LS_Iter iterations, the algorithm returns the best solution found so far.

Algorithm 3. Local_Search(\mathcal{T}) Pseudocode

```

1: while ( $LS\_Iter > 0$ ) do
2:    $\mathcal{T}' = \mathcal{T}$ ;
3:    $\tau_a = \mathcal{T}'[random(0, |\mathcal{T}'|)]$ ;
4:    $\tau_b = \mathcal{T}'[random(0, |\mathcal{T}'|)]$  with  $\tau_b \neq \tau_a$ ;
5:    $C_a, C_b =$  Choose Two Customer of Minimum Distance from  $\tau_a$  and  $\tau_b$ ;
6:   Add  $c_a$  into  $\tau_b$ , Delete  $c_a$  from  $\tau_a$ ;
7:   Add  $c_b$  into  $\tau_a$ , Delete  $c_b$  from  $\tau_b$ ;
8:   if (Capacity Constraints for  $\tau_a$  and  $\tau_b$ ) then
9:      $\mathcal{T}' = \text{Solve\_MILP}(\mathcal{T}')$ ;
10:    if ( $f(\mathcal{T}') < f(\mathcal{T})$ ) then
11:       $\mathcal{T} = \mathcal{T}'$ ;
12:    end if
13:  end if
14: end while
15: return  $\mathcal{T}$ ;

```

5 Computational Study

In this section, we analyze the performance of *MathGRASP* for solving the previously discussed CumVRP variants. All implementations were done in C++11 using Visual Studio v15.9.2. The associated MILP formulations are solved using IBM ILOG CPLEX v12.9.0 API. The equipment used for the tests was a processor Intel(R) Xeon(R) Platinum 81171M CPU 2.60 GHz with 16 GB RAM memory on Windows 10 OS. The matheuristic was run in single thread mode.

5.1 Problem Instances

We perform tests on a selected set of 9 instances of PRPLIB proposed in [30], this dataset consists of 9 sets of 20 instances each one, where the customers or nodes quantities are in the range of 10 to 200 customer, and the values $\omega = 6350$ kg and $Q = 3650$ kg. In Table 1 shows the features for the set of selected instances.

Table 1. Features of PRPLib instances.

Instance	Customers
UK10_01	10
UK15_01	15
UK20_20	20
UK25_01	25
UK50_01	50
UK75_01	75
UK100_05	100
UK150_12	150
UK200_01	200

5.2 Parameter Setting

A parameter tuning process was performed using ParamILS [31], an iterated local search algorithm that works searching for better-quality parameter settings in the neighborhoods of the current one. We provide to the tuner the following set of parameters: $rcl \in \{3, 5, 7\}$, $Max_Iter \in \{25, 50\}$, and $LS_Iter \in \{25, 50\}$. The results of this parameter setting indicate that values of $Max_Iter=50$, $LS_Iter=50$ and $rcl=5$ are the most suitable parameters.

5.3 Results

Our matheuristic approach executed on the selected instances shows that it keeps a stable performance mainly in small instances. This performance is due to the wide scalability of the problem that we solve and the construction of the initial subtours, trying to incorporate in each subtour the largest (see box-plots in Fig. 2).

Table 2 shows the results of the matheuristic approach proposed in this work (see Sect. 4) for CumVRP-hTW and CumVRP-sTW. In this table, column 1 reports the instance studied, columns 2 and 4 and columns 3 and 6 provide the cumulative cost values and the number of vehicles with which the tour can be traveled in CumVRP-hTW and CumVRP-sTW, respectively. Column 5 represents the cost of penalties due to delays of end of service, only valued for

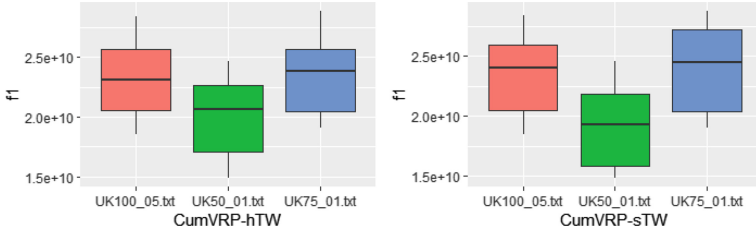


Fig. 2. Box-plot showing the f_1 values (*cumulative cost*) for CumVRP-hTW and CumVRP-sTW on PRPLib selected instances.

CumVRP-sTW. In the column (Gap (%)) the gap between both solutions (see column 2 and 4) is calculated according to: $100 \times (f_{1Col4} - f_{1Col2}) / f_{1Col4}$, f_{1Col4} represents the best solution obtained in each instance. In this sense, it shows that the best results are obtained by solving CumVRP with soft time windows. A negative value in this column indicates that an improvement in terms of the objective function value is obtained.

Table 2. Computational results of the proposed matheuristic on PRPLib selected instances. We consider $\mu = 1800$ s and $P = 1$ cost/s.

Instance	<i>MathGRASP</i>					
	CumVRP-hTW		CumVRP-sTW			Gap (%)
	f_1	k	f_1	f_3	k	
UK10_01	3153462870	2	3153462870	0.000	2	0.000
UK15_01	6304836673	2	6189389516	0.000	2	-1.865
UK20_20	8625693080	3	8510155558	299.000	3	-2.507
UK25_01	6403951960	3	6401431234	0.000	3	-0.039
UK50_01	14863720450	7	14822031464	0.000	8	-0.281
UK75_01	19127115942	15	19001902452	846.002	14	-0.659
UK100_05	18568502490	23	18504497844	1269.003	21	-0.346
UK150_12	39788688093	21	39094971926	792.000	21	-1.774
UK200_01	36133585632	25	34569374470	144.030	25	-4.525

The results on the instances UK10_01, UK15_01, UK25_01, and UK50_01 does not show a reduction in the cumulative cost between CumVRP-hTW and CumVRP-sTW, because the time windows interval in these instances is very wide, and it is not necessary to violate the upper limit U_i . The results for UK20_20, UK75_01, UK100_05, UK150_12, and UK200_01 shows a minimization of cumulative cost incurring in several penalties values by allowing violations in the end times of services in some customers.

Regarding the number of vehicles, this can vary between the soft and hard variant of the problem, mainly in the instances with the largest number of customers, this is due to the fact that in the construction procedure in Algorithm 2 used for constructing subtours, it is taken into account the capacity of the vehicle. However, in the constructive phase of this algorithm, the use of randomness causes the construction of different subtours in each iteration of the algorithm. Thus, for the same instance in an execution a different number of subtours can be generated when compared to another execution of the algorithm, that implies the variation on the number of vehicles.

The results obtained show that when time window violations are permitted, the routes obtained have lower cumulative costs than problem with hard time windows is solved. Due to the fact that the emissions of greenhouse gases can be proportional to the cumulative cost, when this cost is reduced it also causes some dissatisfaction of the customers (in terms of satisfying the time window constraints), so it is possible to reduce the emissions.

The level of emissions reduction depends considerably on the attitude that a company shows in compromising its service process. In this sense, if the allowed delay time is very low or the penalties are very high, then it is not feasible to violate the time window. In addition, a company could estimate different penalties values P depending on how much each customer represents. Figure 3 shows the calculation of CO₂ emissions for each of the variants studied.

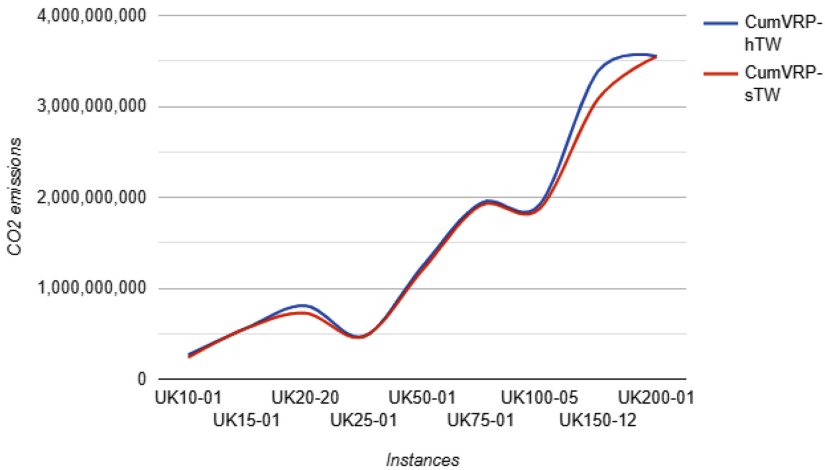


Fig. 3. Comparison of total CO₂ emissions values for CumVRP-hTW and CumVRP-sTW on the selected instances.

The calculation of emissions is done by using the following equation:

$$\begin{aligned} CO_2 \text{ emissions (kg)} &= \text{distance travelled(km)} \\ &\times \text{payload(t)} \\ &\times \text{emission factor(kg } CO_2e/t \text{ km)} \end{aligned} \quad (9)$$

where emission cost is obtained using the emission factor value of 0.41693 $kg CO_2e/t km$, averagely value for diesel-powered rigid vehicles weighted between 7.5 t and 17 t , suggested by DEFRA (2010) [32].

6 Conclusion and Future Research

In this work, we have investigated the Cumulative VRP with hard and soft time windows. For solving these problem variants, we have developed a matheuristic optimization approach that combines a Greedy Randomized Adaptive Search Procedure with the exact solution of the optimization model.

The results show that including soft time windows constraints, lead to significant reductions in environmental costs. In this sense, this consideration of the time windows might reflect better situations than hard time windows, especially considering dynamic changes in problem features. However, when considering the violation of a time window, a customer's dissatisfaction occurs, but despite this, there is a correct trade-off between time windows penalties and environmental related costs.

Our matheuristic approach shows that the quality of the solutions increase when each subtour is seen as a low complexity subproblem that can be solved by a mathematical programming approach.

Finally, as future work, we will propose a benchmark that encompasses novel features and new challenges for CumVRP with hard and soft time windows constraints. In addition, we propose to extend CumVRP with soft and hard constraints by considering fleets of heterogeneous vehicles.

Acknowledgments. This work has been partially supported by ANID-PFCHA/ Doctorado Nacional /2020-21200871 and CONICYT-PFCHA (Doctorado Nacional /2017-21171857), and in part by Proyectos de Línea de Investigación Regular (PLLIR_2020_67, UTFSM) and Programa de Incentivo a la Iniciación Científica (PIIC, UTFSM). We are grateful to anonymous referees, whose valuable comments helped us to strongly improve the paper.

References

1. Sbihi, A., Eglese, R.W.: Combinatorial optimization and green logistics. *Ann. Oper. Res.* **175**(1), 159–175 (2010). <https://doi.org/10.1007/s10479-009-0651-z>
2. Sahin, B., Yilmaz, H., Ust, Y., Guneri, A.F., Gulsun, B.: An approach for analysing transportation costs and a case study. *Eur. J. Oper. Res.* **193**(1), 1–11 (2009)
3. Demir, E., Bektaş, T., Laporte, G.: A review of recent research on green road freight transportation. *Eur. J. Oper. Res.* **237**(3), 775–793 (2014)

4. Dantzig, G.B., Ramser, J.H.: The truck dispatching problem. *Manag. Sci.* **6**(1), 80–91 (1959)
5. Kara, İ., Kara, B.Y., Yetiş, M.K.: Cumulative vehicle routing problems. In: *Vehicle Routing Problem*, pp. 85–98. In-Teh (2008)
6. Kara, İ., Kara, B.Y., Yetiş, M.K.: Energy minimizing vehicle routing problem. In: Dress, A., Xu, Y., Zhu, B. (eds.) *COCOA 2007*. LNCS, vol. 4616, pp. 62–71. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-73556-4_9
7. Singh, R.R., Gaur, D.R.: Cumulative VRP: a simplified model of green vehicle routing. In: Cinar, D., Gakis, K., Pardalos, P.M. (eds.) *Sustainable Logistics and Transportation*. SOIA, vol. 129, pp. 39–55. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-69215-9_3
8. Solomon, M.M.: Algorithms for the vehicle routing and scheduling problems with time window constraints. *Oper. Res.* **35**(2), 254–265 (1987)
9. Desaulniers, G., Madsen, O.B.G., Ropke, S.: The vehicle routing problem with time windows (chap. 5). In: *Vehicle Routing: Problems, Methods, and Applications*, 2nd edn., pp. 119–159. SIAM (2014)
10. Kallehauge, B.: Formulations and exact algorithms for the vehicle routing problem with time windows. *Comput. Oper. Res.* **35**(7), 2307–2330 (2008). Part Special Issue: Includes selected papers presented at the ECCO 2004 European Conference on combinatorial Optimization
11. Fagerholt, K.: Ship scheduling with soft time windows: an optimisation based approach. *Eur. J. Oper. Res.* **131**(3), 559–571 (2001)
12. Bektaş, T., Ehmke, J.F., Psaraftis, H.N., Puchinger, J.: The role of operational research in green freight transportation. *Eur. J. Oper. Res.* **274**(3), 807–823 (2019)
13. Lin, C., Choy, K.L., Ho, G.T.S., Chung, S.H., Lam, H.Y.: Survey of green vehicle routing problem: past and future trends. *Expert Syst. Appl.* **41**(4, Part 1), 1118–1138 (2014)
14. Gaur, D.R., Singh, R.R.: Cumulative vehicle routing problem: a column generation approach. In: Ganguly, S., Krishnamurti, R. (eds.) *CALDAM 2015*. LNCS, vol. 8959, pp. 262–274. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-14974-5_25
15. Laporte, G.: The vehicle routing problem: an overview of exact and approximate algorithms. *Eur. J. Oper. Res.* **59**(3), 345–358 (1992)
16. Gendreau, M., Potvin, J.-Y., et al.: *Handbook of Metaheuristics*, vol. 2. Springer, Cham (2010). <https://doi.org/10.1007/978-3-319-91086-4>
17. Cinar, D., Cayir Ervural, B., Gakis, K., Pardalos, P.M.: Constructive algorithms for the cumulative vehicle routing problem with limited duration. In: Cinar, D., Gakis, K., Pardalos, P.M. (eds.) *Sustainable Logistics and Transportation*. SOIA, vol. 129, pp. 57–86. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-69215-9_4
18. Cinar, D., Gakis, K., Pardalos, P.M.: Reduction of co2 emissions in cumulative multi-trip vehicle routing problems with limited duration. *Environ. Model. Assess.* **20**(4), 273–284 (2015). <https://doi.org/10.1007/s10666-014-9434-2>
19. Ngueveu, S.U., Prins, C., Calvo, R.W.: An effective memetic algorithm for the cumulative capacitated vehicle routing problem. *Comput. Oper. Res.* **37**(11), 1877–1885 (2010). *Metaheuristics for Logistics and Vehicle Routing*
20. Nucamendi-Guillén, S., Angel-Bello, F., Martínez-Salazar, I., Cordero-Franco, A.E.: The cumulative capacitated vehicle routing problem: new formulations and iterated greedy algorithms. *Expert Syst. Appl.* **113**, 315–327 (2018)

21. Archetti, C., Speranza, M.G.: A survey on matheuristics for routing problems. *EURO J. Comput. Optimiz.* **2**(4), 223–246 (2014). <https://doi.org/10.1007/s13675-014-0030-7>
22. Sartori, C.S., Buriol, L.S.: A matheuristic approach to the pickup and delivery problem with time windows. In: Cerulli, R., Raiconi, A., Voß, S. (eds.) *ICCL 2018*. LNCS, vol. 11184, pp. 253–267. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00898-7_16
23. Lalla-Ruiz, E., Voß, S.: A POPMUSIC approach for the multi-depot cumulative capacitated vehicle routing problem. *Optimiz. Lett.* **14**, 671–691 (2019). <https://doi.org/10.1007/s11590-018-1376-1>
24. Keskin, M., Çatay, B.: A matheuristic method for the electric vehicle routing problem with time windows and fast chargers. *Comput. Oper. Res.* **100**, 172–188 (2018)
25. Grangier, P., Gendreau, M., Lehuédé, F., Rousseau, L.-M.: A matheuristic based on large neighborhood search for the vehicle routing problem with cross-docking. *Comput. Oper. Res.* **84**, 116–126 (2017)
26. Kramer, R., Subramanian, A., Vidal, T., Lucídio dos Anjos, F.C.: A matheuristic approach for the pollution-routing problem. *Eur. J. Oper. Res.* **243**(2), 523–539 (2015)
27. Lalla-Ruiz, E., Voß, S.: POPMUSIC as a matheuristic for the berth allocation problem. *Ann. Math. Artif. Intell.* **76**(1–2), 173–189 (2016). <https://doi.org/10.1007/s10472-014-9444-4>
28. Feo, T.A., Resende, M.G.C.: Greedy randomized adaptive search procedures. *J. Global Optimiz.* **6**(2), 109–133 (1995)
29. Sánchez, M.G., Gil, A.F., Castro, C.: Integrating a SMT solver based local search in ant colony optimization for solving RCMPSP. In: *2019 IEEE Latin American Conference on Computational Intelligence, LA-CCI*, pp. 1–6 (2019)
30. Demir, E., Bektaş, T., Laporte, G.: An adaptive large neighborhood search heuristic for the pollution-routing problem. *Eur. J. Oper. Res.* **223**(2), 346–359 (2012)
31. Hutter, F., Hoos, H.H., Stützle, T.: Automatic algorithm configuration based on local search. In: *Proceedings of the 22nd National Conference on Artificial Intelligence - Volume 2, AAAI 2007*, pp. 1152–1157. AAAI Press (2007)
32. Food Statistics Pocketbook Defra and Ecosystems Evidence Plan. Department for environment. Food and Rural Affairs (2010)