This is the author's version of an article that is to appear in the Proceedings of the Sixteenth International Conference on the Integration of Constraint Programming, Artficial Intelligence, and Operations Research (CPAIOR2019).

A Constraint Programming Approach to Electric Vehicle Routing with Time Windows

Kyle E. C. Booth and J. Christopher Beck

Department of Mechanical & Industrial Engineering
University of Toronto, Toronto, Ontario M5S 3G8, Canada
{kbooth, jcb}@mie.utoronto.ca

Abstract. The Electric Vehicle Routing Problem with Time Windows (EVRPTW) extends traditional vehicle routing to address the recent development of electric vehicles (EVs). In addition to traditional VRP problem components, the problem includes consideration of vehicle battery levels, limited vehicle range due to battery capacity, and the presence of vehicle recharging stations. The problem is related to others in emissions-conscious routing such as the Green Vehicle Routing Problem (GVRP). We propose the first constraint programming (CP) approaches for modeling and solving the EVRPTW and compare them to an existing mixed-integer linear program (MILP). Our initial CP model follows the alternative resource approach previously applied to routing problems, while our second CP model utilizes a single resource transformation. Experimental results on various objectives demonstrate the superiority of the single resource transformation over the alternative resource model, for all problem classes, and over MILP, for the majority of medium-to-large problem classes. We also present a hybrid MILP-CP approach that outperforms the other techniques for distance minimization problems over long scheduling horizons, a class that CP struggles with on its own.

Keywords: Electric Vehicle Routing · Green Vehicle Routing · Constraint Programming · Mixed-Integer Linear Programming · Optimization

1 Introduction

Fueled by emission regulations, government subsidies, and the benefits of a more eco-friendly image, electric vehicle (EV) utilization in logistics has seen significant growth in recent years [9]. Outside of logistics, EVs have experienced a growing adoption within the consumer automotive industry [28] and have shown promise in car sharing pilot projects [26]. While not currently cost-competitive with internal combustion engines due to high acquisition costs and limited operational range [14], the benefits of EVs coupled with an increasing number of socially and environmentally-aware consumers are driving the adoption of the technology. The industry has also seen significant investment in the development of required recharging infrastructure. As with traditional fleets, the case for EVs can be significantly bolstered via effective route planning.

This is the author's version of an article that is to appear in the Proceedings of the Sixteenth International Conference on the Integration of Constraint Programming, Artficial Intelligence, and Operations Research (CPAIOR2019).

The vehicle routing literature has recently addressed this emerging technology through the introduction of the Electric Vehicle Routing Problem with Time Windows (EVRPTW) [29], building on previous work conducted on green logistics, including the Green Vehicle Routing Problem (GVRP) [12]. The problem involves routing a fleet of vehicles to satisfy customer demands while adhering to the battery capacity and range of the fleet EVs. The EVRPTW literature has seen considerable research activity, including the development of sophisticated exact approaches [10, 7], metaheuristics [29, 13], and the introduction of increasingly rich problem definitions driven by real-world logistics use cases [27, 17]. There have been, however, no efforts thus far to explore the use of constraint programming (CP) to model and solve the problem.

Recognizing EV routing as a strategic area for methodological development, we investigate the use of monolithic (i.e., non-decomposed) MILP and CP models to solve the problem. The contributions of this paper are as follows:

i. We propose the first CP approaches for the EVRPTW.
ii. We introduce a single resource transformation for CP formulations that use optional interval, sequence, and cumulative function expression variables. The transformation significantly extends the size of problems that can be solved with CP, and can be applied to other homogeneous VRP and multi-machine scheduling problems.
iii. We demonstrate, through empirical evaluation, that our single resource CP approach significantly outperforms the alternative resource CP model, and outperforms MILP for nearly all medium-to-large problem classes.
iv. Following the observation that MILP excels at quickly finding high quality solutions to distance minimization problems with large scheduling horizons, we propose a hybrid MILP-CP technique that outperforms the individual approaches on this problem class.

This paper is organized as follows. Section 2 defines the EVRPTW problem and presents an existing MILP model. Section 3 details related work for the problem studied. Section 4 presents two CP models, alternate modeling strategies, and an initial empirical evaluation with accompanying analysis. Section 5 illustrates a hybrid CP-MILP approach, motivated by the strength of MILP for larger, long horizon problems, and presents hybrid experimental results with detailed analysis. Finally, Section 6 provides concluding remarks.

2 Problem Definition

The Electric Vehicle Routing Problem with Time Windows (EVRPTW) is a static optimization problem that aims to route a fleet of electric vehicles to satisfy customer requests [29]. Following existing notation, we let $V' = V \cup F'$ be the set of N vertices where $V = \{v_1, \dots, v_n\}$ is the set of customer requests, F is the set of recharging stations, and $F' = \{v_{n+1}, \dots, v_N\}$ is the set of augmented recharging stations that includes dummy vertices to allow multiple visits to each of the stations in F. We let vertices v_0 and v_{N+1} correspond to start and end

This is the author's version of an article that is to appear in the Proceedings of the Sixteenth International Conference on the Integration of Constraint Programming, Artficial Intelligence, and Operations Research (CPAIOR2019).

instances of the vehicle depot, where each vehicle starts and ends. Sets with depot subscripts include the indicated instances of the depot (i.e., $V'_{N+1} = V' \cup \{v_{N+1}\}$ and $V'_{0,N+1} = V' \cup \{v_0, v_{N+1}\}$). The problem is then defined on a graph with vertices $V'_{0,N+1}$ and undirected arcs $A = \{(i,j)|i,j \in V'_{0,N+1}, i \neq j\}$. Each arc is assigned a distance, travel time, and energy consumption, d_{ij}, t_{ij}, and $h \cdot d_{ij}$, respectively, where h is a constant energy consumption rate. Vehicles are initially positioned at the depot and start with maximum capacity C, while customer vertices, $i \in V$, are assigned a positive demand, $q_i \leq C$, and a time window, $[e_i, l_i]$.[1] The time window of the start depot is $[0, 0]$, and the end depot is $[H, H]$, where H is the problem horizon. Each recharging station has a time window of the entire horizon, namely $[e_i = 0, l_i = H], \forall i \in F'$. Customer vertices, $i \in V$, have a service time s_i. The depot instances each have a null service time and the service time at recharging stations is a variable. Vehicles have maximum battery capacity Q and recharge linearly at rate g. The problem then minimizes an objective function, often a combination of fleet size and travel distance.

An existing two-index MILP model from the literature [29] is detailed by Eqns. (1) through (12). Binary variable x_{ij} is 1 if arc $(i,j) \in A$ is traveled and 0 otherwise. Continuous variables τ_i, u_i, and y_i represent the arrival time, remaining cargo, and remaining energy, respectively, at vertex $i \in V'_{0,N+1}$. This formulation assumes an unlimited number of homogeneous vehicles are available and only permits full vehicle recharges (i.e., if a vehicle visits a recharge station vertex, the service time is the difference between its maximum energy capacity and current energy level, divided by the recharge rate). The augmented recharge station set, F', is constructed such that the number of dummy vertices associated with each recharge station, n_f, represents the number of times the associated recharge station can be visited across all vehicles (with $|F'| = n_f \cdot |F|$). Following the literature, n_f is set to be relatively small, to reduce the network size, but large enough to not restrict multiple beneficial visits [12]. We note that heuristically choosing a value for n_f, as in the literature, can potentially remove optimal solutions.

Objective (1) details the weighted objective function, where $\alpha \in [0, 1]$ identifies the emphasis on fleet size minimization and $\beta \in [0, 1]$ on travel distance minimization. Constraint (2) ensures each customer request is satisfied, while Constraint (3) restricts each recharge station in the augmented set to be visited at most once (due to the augmented vertices, each recharge station can be visited at most n_f times). Constraint (4) enforces the flow for non-depot nodes. Constraints (5)-(6) prevent the formation of subtours, with disjunctive constant $M = (l_0 + g \cdot Q)$. Constraint (7) ensures demand fulfillment at customer vertices and Constraints (8)-(9) constrain energy levels to be feasible. Constraint (10) requires customer visits to satisfy the time windows and Constraints (11)-(12) identify binary and continuous variable domains.

[1] Service must start within the time window.

This is the author's version of an article that is to appear in the Proceedings of the Sixteenth International Conference on the Integration of Constraint Programming, Artficial Intelligence, and Operations Research (CPAIOR2019).

$$\min \quad \alpha \sum_{j \in V'} x_{0j} + \beta \sum_{i \in V'_0} \sum_{j \in V'_{N+1}, i \neq j} d_{ij} x_{ij} \tag{1}$$

$$\text{s.t.} \quad \sum\nolimits_{j \in V'_{N+1}, i \neq j} x_{ij} = 1 \qquad \forall i \in V, \tag{2}$$

$$\sum\nolimits_{j \in V'_{N+1}, i \neq j} x_{ij} \leq 1 \qquad \forall i \in F', \tag{3}$$

$$\sum\nolimits_{i \in V'_{N+1}, i \neq j} x_{ji} - \sum\nolimits_{i \in V'_0, i \neq j} x_{ij} = 0 \qquad \forall j \in V', \tag{4}$$

$$\tau_i + (t_{ij} + s_i) x_{ij} - l_0(1 - x_{ij}) \leq \tau_j \qquad \forall i \in V_0, j \in V'_{N+1}, i \neq j, \tag{5}$$

$$\tau_i + t_{ij} x_{ij} + g(Q - y_i) - M(1 - x_{ij}) \leq \tau_j \qquad \forall i \in F', j \in V'_{N+1}, i \neq j, \tag{6}$$

$$0 \leq u_j \leq u_i - q_i x_{ij} + C(1 - x_{ij}) \qquad \forall i \in V'_0, j \in V'_{N+1}, i \neq j, \tag{7}$$

$$0 \leq y_j \leq y_i - (h \cdot d_{ij}) x_{ij} + Q(1 - x_{ij}) \qquad \forall j \in V'_{N+1}, i \in V, i \neq j, \tag{8}$$

$$0 \leq y_j \leq Q - (h \cdot d_{ij}) x_{ij} \qquad \forall j \in V'_{N+1}, i \in F'_0, i \neq j, \tag{9}$$

$$e_j \leq \tau_j \leq l_j \qquad \forall j \in V'_{0,N+1}, \tag{10}$$

$$x_{ij} \in \{0, 1\} \qquad \forall i \in V'_0, j \in V'_{N+1}, i \neq j, \tag{11}$$

$$\tau_0 = 0, u_0 = C, y_0 = Q. \tag{12}$$

Two Index Formulation. With the exception of x_{ij}, the variables are continuous, modeling visit time, load, and energy level via sequencing constraints. The model represents multiple vehicles by relaxing the unitary out- and in-flow on the start and end depot vertices, respectively. This modeling technique is effective as it does not multiply the number of variables by the number of (symmetric) vehicles.

Problem Variants. The fixed fleet variant can be modeled with the inclusion of a constraint of the form: $\sum_{j \in V'} x_{0j} \leq m$, where m is the fleet size. A variant with heterogeneous vehicles can be modeled for k different vehicle types by adding an index for the vehicle type to the arc, cargo, and energy consumption variables (i.e., x^k_{ij}, u^k_i, and y^k_i), with similar adjustments to the parameters [17]. Additional problem variants, such as partial recharges, can also be considered through the inclusion of additional constraints/variables [10, 7].

3 Related Work

Research on energy-aware, environmentally conscious vehicle routing is a relatively new area with a flurry of research in recent years [24]. In response to a growing commitment within the United States to investigate alternative fuel sources, the vehicle routing literature introduced the GVRP, detailing a MILP formulation, construction heuristic, and clustering algorithm [12]. Since the introduction of the GVRP, EV routing has grown dramatically, with initial EVRPTW work for homogeneous fleets and full re-charges including a MILP model and a hybrid variable neighborhood search/tabu search solution technique [29]. Subsequent research was developed with approaches for heterogeneous vehicles [17], and partial recharging problem variants [13, 10, 19]. Recent work has

This is the author's version of an article that is to appear in the Proceedings of the Sixteenth International Conference on the Integration of Constraint Programming, Artficial Intelligence, and Operations Research (CPAIOR2019).

also been conducted on modeling non-linear energy consumption [27] as well as incorporating richer, industry-driven problem constraints [1].

Although the literature for mathematical programming-based approaches (e.g., MILP, branch-and-price, branch-and-cut) for the GVRP and the EVRPTW is abundant [12, 29, 10, 17], to the authors' knowledge, the use of CP for solving these problems has not yet been investigated. While the performance achieved by sophisticated branch-and-price-and-cut algorithms for EVRPTW [10] is unlikely to be surpassed by monolithic modeling methods utilizing off-the-shelf solvers, the practicality and flexibility of such approaches, including MILP and CP, often translate to more widespread adoption.

In general vehicle routing, CP has been offered as an alternative to mathematical programming approaches for quite some time [30, 2]. Recent applications include work on the multiple traveling salesman problem [31], team orienteering [15], dynamic dial-a-ride routing [3], bike share balancing [11], joint vehicle and crew routing [23], and patient transportation [8, 25], though these efforts do not consider fuel constraints. While CP has not been explicitly proposed for GVRP nor EVRPTW as of yet, previous work on snow plow routing [20] and robot task allocation and scheduling [5, 6] propose CP models with consideration for energy consumption and replenishment.

4 Constraint Programming Approaches

In this section we present two CP formulations for the EVRPTW. Our models are posed as scheduling formulations with optional activities [21, 22]. As is becoming increasingly common in CP-based approaches [5, 8, 25], the proposed models make use of three primary decision variable types, namely: optional interval variables, sequence variables, and cumulative function expressions.

Optional Interval Variables. Formally, optional interval variables are decision variables whose possible values are a convex interval: $\{\perp\} \cup \{[s, \epsilon) | s, \epsilon \in \mathbb{Z}, s \leq \epsilon\}$, where s and ϵ are the start and end values of the interval and $\perp$ is a special value indicating the variable is not present in the solution. The presence (binary), start time, and length of an optional interval variable, var, can be expressed within a CP model using $\texttt{Pres}(var)$, $\texttt{Start}(var)$, and $\texttt{Length}(var)$, respectively. We use the notation $\texttt{optIntervalVar}(p, [s, \epsilon])$ to define these variables in our models, where p is the processing time of the task (and can be variable). Model constraints are only enforced over present interval variables.

Sequence Variables. This variable type is useful for expressing model constraints over a permutation of present (i.e., $\texttt{Pres}(var) = 1$) interval variables. Given the definition of a sequence variable, π, various constraints can be expressed, including those on the interval variable previous to var in the sequence, $\texttt{Prev}_{\pi}(var)$, and temporal constraints such as the $\texttt{NoOverlap}(\pi)$ constraint, which ensures the interval variables in the sequence do not interfere temporally.

This is the author's version of an article that is to appear in the Proceedings of the Sixteenth International Conference on the Integration of Constraint Programming, Artficial Intelligence, and Operations Research (CPAIOR2019).

Cumulative Function Expressions. It is often useful to represent the usage of a renewable resource as the sum of individual interval variable contributions over time. Given a cumulative function expression variable, f, we can express impact on the expression using the $f \pm \texttt{StepAtStart}(var, impact)$ expression, specifying that at the start of interval variable var, function f has an increment (or decrement) of $impact$. The constraint $\texttt{AlwaysIn}(f, [s, \epsilon], [min, max])$ ensures that $min \leq f \leq max$ holds for all time points in s to ϵ and a similar constraint $\texttt{AlwaysIn}(f, var, [min, max])$ ensures that $min \leq f \leq max$ holds during the processing of interval variable var. Cumulative expression variables are useful in representing both the vehicle load and energy constraints, and have been used for similar problems [20, 5].

4.1 Alternative Resource Model

Our first CP model follows the traditional alternative resource model for formulating VRPs in CP [20, 8], and, in contrast to the two-index MILP presented in Section 2, explicitly represents the vehicles. We define an upper bound on the number of vehicles to be equal to the number of customer requests, $|K| = |V|$, representing the worst-case where each customer is serviced by a separate vehicle. For each customer request, $i \in V$, we introduce a mandatory interval variable, $\bar{x}_i$. We create an optional interval variable, x_i^k, for each vertex, $i \in V'$, for each vehicle, $k \in K$. We also introduce start and end interval variables, x_0^k and x_{N+1}^k, with null duration for each vehicle to represent the depot.

The model considers a set of $|V'_{0,N+1}|$ interval variables and a sequence variable, π^k, for each vehicle $k \in K$. Each interval variable x_i^k, for all $i \in V'_{0,N+1}$, represents the time period in which the vehicle visits i. Thus, expressions $\texttt{StartOf}(x_i^k)$ and $\texttt{EndOf}(x_i^k)$ correspond to the arrival and departure time of vehicle k at location i, respectively. The expression $\texttt{Pres}(x_i^k) = 1$ if vehicle $k \in K$ visits location i (i.e., the interval variable is present in the solution), and 0 otherwise. Sequence variable π^k is defined over the set of interval variables involving vehicle k, and represents the sequence of visits. Vehicle load consumption and energy level consumption/replenishment are modeled with cumulative function expressions. We let $\mathcal{C}^k$ and $\mathcal{Q}^k$ be cumulative function expressions representing the load and energy level of vehicle $k \in K$ throughout its route.

Our alternative resource CP model is detailed by Eqns. (13)-(27). Objective (13) represents the minimization of fleet size and distance traveled. Constraint (14) ensures that each customer is serviced by one vehicle and Constraint (15) enforces that tasks assigned to a vehicle, represented by the sequence variable π^k, do not interfere temporally, including travel times. Constraints (16)-(17) ensure that the vehicle load does not fall below zero over the planning horizon, represented as a cumulative function expression with negative impact for served customers. Constraints (18)-(19) ensure vehicle energy stays within permissible limits, also represented as a cumulative function expression with negative impact for travel between locations and a positive impact for vehicle recharging. We note that the impact for energy replenishment tasks includes the negative

This is the author's version of an article that is to appear in the Proceedings of the Sixteenth International Conference on the Integration of Constraint Programming, Artficial Intelligence, and Operations Research (CPAIOR2019).

contribution of the travel to the recharge station. Constraint (20) ensures that during a recharge task, the energy of the vehicle is set to its capacity; whenever a vehicle recharges, it does so fully. Constraint (22) ensures each recharge station is used at most n_f times across the fleet, where $F'(i)$ represents all dummy recharge stations associated with real recharge station $i \in F$. Constraint (21) enforces that the start and end depot instances for a vehicle be first and last in the sequence variable for that vehicle, while Constraints (23)-(27) provide the definitions of the interval and sequence variables.

As the alternative resource formulation explicitly represents each vehicle, the number of variables can become unwieldy for larger problems. Specifically, the formulation has $|V| + |K| \cdot |V'_{0,N+1}|$ interval variables, $|K|$ sequence variables, and $2|K|$ cumulative function expression variables.

$$\min \sum_{k \in K} \left(\alpha \texttt{Pres}(x_0^k) + \beta \sum_{i \in V'_{N+1}} \texttt{Pres}(x_i^k) \cdot d_{\texttt{Prev}_{\pi^k}(i),i} \right) \tag{13}$$

$$\text{s.t. } \texttt{Alternative}(\bar{x}_i, \{x_i^1, \dots, x_i^{|V|}\}) \quad \forall i \in V, \tag{14}$$

$$\texttt{NoOverlap}(\pi^k, \{t_{ij} : (i,j) \in A\}) \quad \forall k \in K, \tag{15}$$

$$\mathcal{C}^k = \texttt{StepAtStart}(x_0^k, C) - \sum\nolimits_{i \in V} \texttt{StepAtStart}(x_i^k, q_i) \quad \forall k \in K, \tag{16}$$

$$\texttt{AlwaysIn}(\mathcal{C}^k, [0, H], [0, C]) \quad \forall k \in K, \tag{17}$$

$$\mathcal{Q}^k = \texttt{StepAtStart}(x_0^k, Q) - \sum\nolimits_{i \in V'_{N+1}} \texttt{StepAtStart}(x_i^k, h \cdot d_{\texttt{Prev}_{\pi^k}(i),i}) + \sum\nolimits_{i \in F'} \texttt{StepAtStart}(x_i^k, g \cdot \texttt{Length}(x_i^k)) \quad \forall k \in K, \tag{18}$$

$$\texttt{AlwaysIn}(\mathcal{Q}^k, [0, H], [0, Q]) \quad \forall k \in K, \tag{19}$$

$$\texttt{AlwaysIn}(\mathcal{Q}^k, x_i^k, [Q, Q]) \quad \forall i \in F', k \in K, \tag{20}$$

$$\texttt{First}(\pi^k, x_0^k),\ \texttt{Last}(\pi^k, x_{N+1}^k) \quad \forall k \in K, \tag{21}$$

$$\sum_{k \in K} \sum_{j \in F'(i)} \texttt{Pres}(x_j^k) \le n_f \quad \forall i \in F, \tag{22}$$

$$x_i^k : \texttt{optIntervalVar}([0, Q \cdot g^{-1}], [0, H]) \quad \forall i \in F', k \in K, \tag{23}$$

$$x_i^k : \texttt{optIntervalVar}(s_i, [e_i, l_i]) \quad \forall i \in V, k \in K, \tag{24}$$

$$\bar{x}_i : \texttt{intervalVar}(s_i, [e_i, l_i]) \quad \forall i \in V, \tag{25}$$

$$x_0^k : \texttt{intervalVar}(0, [0, 0]), x_{N+1}^k : \texttt{intervalVar}(0, [H, H]) \quad \forall k \in K, \tag{26}$$

$$\pi^k : \texttt{sequenceVar}(\{x_0^k, \dots, x_{N+1}^k\}) \quad \forall k \in K. \tag{27}$$

Model Strengthening

Cumulative Resource Constraint. Similar to a previous CP formulation for patient transportation [8], we strengthen the baseline formulation with a cumulative resource constraint. We define an auxiliary integer variable representing the number of vehicles in the fleet, $z = \sum_{k \in K} \texttt{Pres}(x_0^k)$. The cumulative constraint is then $\texttt{Cumulative}(\bar{x} \cup \{x_i : i \in F'\}, z)$, which expresses that at any time point

This is the author's version of an article that is to appear in the Proceedings of the Sixteenth International Conference on the Integration of Constraint Programming, Artficial Intelligence, and Operations Research (CPAIOR2019).

in the horizon, the total number of customer interval variables, $\bar{x}$, and present recharge interval variables is bounded by the number of vehicles in the fleet.

Symmetry Breaking Constraints. Due to the large number of homogeneous vehicles, the use of symmetry breaking can be effective. We introduce a constraint of the form $\texttt{Pres}(x_0^k) \geq \texttt{Pres}(x_0^{k+1})$, ensuring vehicles are used in a lexicographic order. We then specify that if a vehicle depot task is not present, it cannot be assigned any other activities via $\texttt{Pres}(x_0^k) \geq \texttt{Pres}(x_i^k), \forall i \in V'_{N+1}$.

Energy Expression Tightening. While the energy impact of a customer visit on vehicle energy level is a variable, we can tighten the domain of its impact by reasoning about minimum and maximum travel consumptions to the considered customer location. More specifically, we add the constraint $\min_{i \in V'_0}(h \cdot d_{ij}) \leq \texttt{HeightAtStart}(x_j^k) \leq \max_{i \in V'_0}(h \cdot d_{ij}), \forall j \in V, k \in K$, where the $\texttt{HeightAtStart}(var, f)$ expression evaluates the individual contribution of an interval variable, var, to a cumulative function expression, f.

4.2 Single Resource Model

Our second CP model, inspired by the modeling efficiency of the two-index MILP for homogeneous vehicles, utilizes a single resource transformation to significantly reduce the number of variables. The transformation represents the problem as an interval variable sequence over an augmented horizon and, like the MILP, does not explicitly represent the vehicles. This modeling strategy, while common in MILP models for VRPs, has been rarely used in CP. In previous work on joint vehicle and crew routing, a similar strategy was used to artificially join the end of one route to the beginning of another when using the `Circuit` global constraint [23], which prevents the formation of subtours among a set of integer variables. However, to our knowledge, the single resource transformation has never been proposed for scheduling-based CP models involving interval, sequence, and cumulative function expression variables. The transformation using these formalisms is challenging as the modeling paradigm does not permit the "resetting" of time as in [23]; we detail how this is accomplished in the remainder of this section. The described transformation can also be applied to homogeneous machine scheduling problems, which we leave to future work.

We visualize the single resource model in Figure 1. The transformation augments the problem horizon from H to $|V| \cdot H$, generating a horizon for each potential vehicle used. In addition to the start and end depot instances, v_0 and v_{N+1}, we define a set of auxiliary depot instances, $\mathcal{H} = \{v_{N+2}, \dots, v_{N+|V|}\}$, representing the end depots of the additional horizon segments. We define the notation $V'_{0,N+1,\mathcal{H}} = V'_{0,N+1} \cup \mathcal{H}$ and undirected arcs $A' = \{(i,j) | i,j \in V'_{0,N+1,\mathcal{H}}, i \neq j\}$. Similarly, we define $\mathcal{H}_{N+1} = v_{N+1} \cup \mathcal{H}$ and $\mathcal{H}_{0,N+1} = \{v_0, v_{N+1}\} \cup \mathcal{H}$. A depot instance, represented as an interval variable, x_i, is assigned with null duration for $i \in \mathcal{H}_{0,N+1}$. These interval variables have start time σ_i, such that $\sigma_0 = 0, \sigma_{N+1} = H, \sigma_{N+2} = 2H$, and so forth. We then create a mandatory interval variable, x_i, for each customer request, $i \in V$, and an optional interval

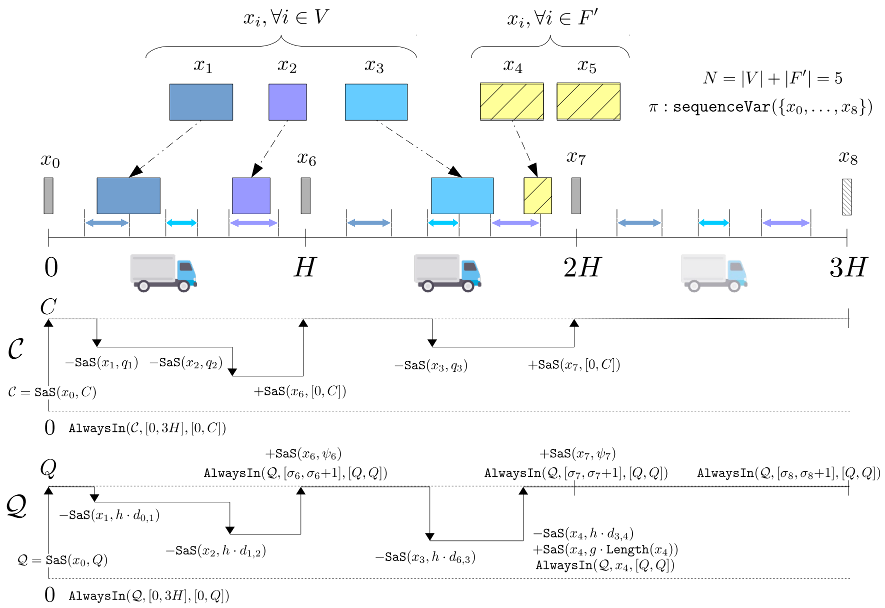

Fig. 1: Single resource transformation for problem with $|V| = 3$ and a single recharge station, $|F| = 1$, with $n_f = 2$ (such that $|F'| = 2$). A horizon segment is created for each potential vehicle and time windows are duplicated. All customer tasks are mandatory with disjoint start time domains and energy tasks (optional) have start time domain of $[0, 3H]$. A cumulative function expression represents vehicle load and energy level (where notation `SaS` corresponds to `StepAtStart` in models). Vehicle assignments can then be inferred by the start times of the tasks themselves. The last horizon segment is not used (set as absent).

variable for each recharge station instance in the augmented set, $i \in F'$. Our model uses a single sequence variable, π, defined over the set of all interval variables, and a single cumulative function expression to model vehicle load, $\mathcal{C}$, with another for energy level, $\mathcal{Q}$. Additionally, at the start of each end depot instance, $i \in \mathcal{H}_{N+1}$, the state of the vehicle must be reset to initial conditions. Thus, the cumulative function expressions for vehicle load and energy have auxiliary positive impacts bringing them to their maximum capacity states. The start time domain for customer requests, $i \in V$, becomes a set of disjoint time windows, where each request time window is replicated over each of the horizon segments. The start domain for customer requests is the entire augmented horizon and the disjoint time windows are enforced with constraints. The start time domain for recharge tasks, $i \in F'$, becomes the entire augmented horizon.

Our single resource CP model is detailed by Eqns. (28)-(42). Objective (28) is our fleet and distance minimization objective function. Constraint (29) enforces temporal feasibility of the interval variable sequence, π, including travel times.

This is the author's version of an article that is to appear in the Proceedings of the Sixteenth International Conference on the Integration of Constraint Programming, Artficial Intelligence, and Operations Research (CPAIOR2019).

To make sure customers are serviced during a valid time window, we use Constraint (30), where $\phi_i = \{0,\ldots,|V|\cdot H\} \setminus \bigcup_{\delta\in\{0,\ldots,|V|\}} \{\delta H + e_i,\ldots,\delta H + l_i + s_i\}$. The `ForbidExtent` constraint prevents an interval variable, x_i, from being scheduled during any time point within the augmented horizon that is not also within one of the disjoint time windows. We discuss a number of alternatives for modeling disjoint time windows with CP in the next section. Constraints (31)-(32) ensure vehicle load feasibility. Constraints (33)-(35) ensure vehicle energy level feasibility while Constraints (36)-(37) dictate any present recharges, as well as horizon end tasks, must charge the vehicle to full energy level. To ensure the resetting of energy level at the end of each horizon, $i \in \mathcal{H}_{N+1}$, we use a positive impact `StepAtStart` with magnitude in $[0, Q - h \cdot d_{\texttt{Prev}_\pi(i),i}]$ expressed by Constraint (34), and the `AlwaysIn` expressed by Constraint (37). These components are illustrated in Figure 1. The position of the start depot in the interval variable sequence, π, is expressed through Constraint (38) and Constraints (39)-(42) identify variable domains.

$$\min \quad \alpha \sum_{i\in\mathcal{H}_{N+1}} \texttt{Pres}(x_i) + \beta \sum_{i\in V'_{\mathcal{H},N+1}} \texttt{Pres}(x_i)\cdot d_{\texttt{Prev}_\pi(i),i} \tag{28}$$

$$\text{s.t.} \quad \texttt{NoOverlap}(\pi, \{t_{ij} : (i,j)\in A'\}) \tag{29}$$

$$\texttt{ForbidExtent}(x_i, \phi_i) \qquad \forall i \in V, \tag{30}$$

$$\mathcal{C} = \texttt{StepAtStart}(x_0, C) - \sum\nolimits_{i\in V} \texttt{StepAtStart}(x_i, q_i) + \sum\nolimits_{i\in\mathcal{H}_{N+1}} \texttt{StepAtStart}(x_i, [0, C]) \tag{31}$$

$$\texttt{AlwaysIn}(\mathcal{C}, [0, |V|\cdot H], [0, C]) \tag{32}$$

$$\mathcal{Q} = \texttt{StepAtStart}(x_0, Q) - \sum\nolimits_{i\in V'} \texttt{StepAtStart}(x_i, h\cdot d_{\texttt{Prev}_\pi(i),i}) + \sum\nolimits_{i\in F'} \texttt{StepAtStart}(x_i, g\cdot \texttt{Length}(x_i)) + \sum_{i\in\mathcal{H}_{N+1}} \texttt{StepAtStart}(x_i, \psi_i) \tag{33}$$

$$0 \le \psi_i \le Q - h\cdot d_{\texttt{Prev}_\pi(i),i} \qquad \forall i\in\mathcal{H}_{N+1}, \tag{34}$$

$$\texttt{AlwaysIn}(\mathcal{Q}, [0, |V|\cdot H], [0, Q]) \tag{35}$$

$$\texttt{AlwaysIn}(\mathcal{Q}, x_i, [Q, Q]) \qquad \forall i \in F', \tag{36}$$

$$\texttt{AlwaysIn}(\mathcal{Q}, [\sigma_i, \sigma_i + 1], [Q, Q]) \qquad \forall i\in\mathcal{H}_{N+1}, \tag{37}$$

$$\texttt{First}(\pi, x_0) \tag{38}$$

$$x_i \ : \ \texttt{optIntervalVar}([0, Q\cdot g^{-1}], [0, H\cdot|V|]) \qquad \forall i\in F', \tag{39}$$

$$x_i \ : \ \texttt{intervalVar}(s_i, [0, H\cdot|V|]) \qquad \forall i \in V, \tag{40}$$

$$x_i \ : \ \texttt{intervalVar}(0, \sigma_i) \qquad \forall i\in\mathcal{H}_{0,N+1}, \tag{41}$$

$$\pi \ : \ \texttt{sequenceVar}(\{x_0,\ldots,x_{N+|V|}\}). \tag{42}$$

The single resource transformation requires only $|V'_{0,N+1}| + |\mathcal{H}|$ interval variables, one sequence variable, and two cumulative function expression variables, a significant reduction from the alternative resource model.

This is the author's version of an article that is to appear in the Proceedings of the Sixteenth International Conference on the Integration of Constraint Programming, Artficial Intelligence, and Operations Research (CPAIOR2019).

Model Strengthening

Optional Horizon Segments. Initially, a horizon segment must be created for each potential vehicle, recalling that the upper bound used is the number of customer requests, $|V|$. This augmented horizon significantly increases the start time domain of the recharge tasks, even though most high quality solutions only use a small fraction of the vehicles allotted. To improve upon this, we develop a technique, similar to the symmetry breaking in the alternate resource model, where horizon segments can be set absent. First, we set all auxiliary end depot instances, $x_i, \forall i \in \mathcal{H}$, as optional interval variables. Next, we introduce an integer variable for each of the end depot instances, w_i, and constrain its value to be the start time of the interval variable (0 if the variable is set as absent), via $\texttt{StartOf}(x_i) = w_i, \forall i \in \mathcal{H}_{N+1}$. We then constrain the end time of the set of customer and recharge visit tasks to be bounded by the maximum w_i value, $\texttt{EndOf}(x_j) \leq \max_{i \in \mathcal{H}_{N+1}} w_i, \forall j \in V'$. Finally, we impose an ordering on the present depot instances using: $\texttt{Pres}(x_i) \geq \texttt{Pres}(x_{i+1}), \forall i \in \mathcal{H} \setminus v_{N+|V|}$.

Energy Expression Tightening. Similar to the technique presented for the alternative resource model, we introduce energy impact tightening constraints for the single resource model as well, namely: $\min_{i \in V_0'}(h \cdot d_{ij}) \leq \texttt{HeightAtStart}(x_j) \leq \max_{i \in V_0'}(h \cdot d_{ij}), \forall j \in V$.

4.3 Alternate Modeling Strategies

We investigated a number of alternate modeling strategies that were found, through initial experiments, to under-perform the proposed models.

Vehicle Energy and Load. The modeling of energy and vehicle load can also be accomplished via auxiliary tracking variables [20] similar to those used in the MILP. The idea is to introduce a numeric variable for each interval variable representing the load or energy level in the sequence after that particular task. This technique is advantageous in that the exact vehicle load or energy level can be accessed at any point along the route, whereas current implementations of cumulative function expressions, as within CP Optimizer, do not support this.

Disjoint Time Windows. The single resource model results in a set of disjoint time windows for customer tasks.[2] The model uses the $\texttt{ForbidExtent}(var, T)$ constraint, restricting an interval variable var from executing at any time point within the restricted set of time points T. This relationship can also be expressed using interval variables by generating a set of fixed interval variables that occupy all the time points in T. Then, a $\texttt{NoOverlap}(\pi_i')$ is added to the model for each customer request task, $i \in V$, where the sequence variable π_i' contains the set of all customer interval variables and the auxiliary fixed interval variables. Finally, similar to the alternative resource CP model, one can generate an alternative

[2] There also exist VRP variants posed with multiple disjoint time windows [18].

This is the author's version of an article that is to appear in the Proceedings of the Sixteenth International Conference on the Integration of Constraint Programming, Artficial Intelligence, and Operations Research (CPAIOR2019).

task for each of the time windows. Although these alternate techniques perform moderately well on smaller problems, the increased model size was detrimental for larger problems.

4.4 Experimental Analysis

We present an empirical assessment of our models on the benchmark data in Table 2. We explore three different objective functions: fleet distance minimization ($\alpha = 1, \beta = 0$) as reported in [29, 10], fleet size minimization ($\alpha = 0, \beta = 1$), and fleet size minimization with distance minimization as a secondary objective ($\alpha = \xi, \beta = 1$), where ξ is a sufficiently small number to lexicographically order the objective components. We reiterate that the intent of this work is to investigate the performance of off-the-shelf optimization models for EVPRTW; state-of-the-art results for distance minimization are found in [10] using sophisticated branch-price-and-cut techniques bolstered by customized labeling algorithms.

Set-up. All experiments are implemented in C++ on an Intel Xeon CPU E5-2690 v4 2.60GHz processor and 16GB of RAM running Ubuntu 14.04. We use CP Optimizer for the CP models and CPLEX for the MILP model from the IBM ILOG CPLEX Optimization Studio version 12.8. All experiments are single-threaded with default search and inference settings. A five minute time limit is used for all experiments.

Table 1: Problem instances. Each value represents the number of instances for a given size/characteristic combination. $|V| \leq 15$ are small instances containing 5, 10, and 15 customers. Clustered, random, and mix refer to the geographical distribution of customer vertices. $|F|$ values are averages across the instances.

			Short Horizon			**Long Horizon**		
$\|V\|$	$\|F\|$	Total	Clustered	Random	Mix	Clustered	Random	Mix
$\leq$15	4.2	36	6	6	6	6	6	6
25	21	56	9	12	8	8	11	8
50	21	56	9	12	8	8	11	8

Instances and Implementation. We conduct our analysis on problem instances taken from the literature [29, 10]. Instances vary w.r.t. the number of customer and recharge station vertices, the length of the scheduling horizon (short and long) and the geographical distribution of the customer vertices (random, clustered, and a mixture of both). The benchmark utilized contains a total of 148 instances summarized in Table 1.

Following the procedure outlined in previous work on the same instances [10], we transform floating point parameter values to integer values such that the problems are amenable to CP modeling. As with most integer transformations,

This is the author's version of an article that is to appear in the Proceedings of the Sixteenth International Conference on the Integration of Constraint Programming, Artficial Intelligence, and Operations Research (CPAIOR2019).

the scaling involved in this process results in much larger variable domain ranges which can have a negative impact on CP approaches. Additionally, we heuristically set the number of visits allowable to each recharge station as $n_f = \lceil |V| \cdot 0.2 \rceil$ (i.e., problems with five customers allow a single visit to each recharge station).

During testing, we found that the CP solver, for both formulations, had a difficult time producing initial feasible solutions for the larger problems, $|V| \in \{25, 50\}$. To mitigate this, we seed the CP search with the initial solution found by the MILP presolve routine. We note that we could have used any initial heuristic here to yield the same result; the presolve results used were often trivial (i.e., each customer serviced by a separate vehicle).

Table 2: Experimental results. The best result of each column for each objective function in bold. 'M': method ran out of memory before entering the search.

	Short Horizon									**Long Horizon**								
	# Feasible			# Best			MRE (%)			# Feasible			# Best			MRE (%)		
Method	≤15	25	50	≤15	25	50	≤15	25	50	≤15	25	50	≤15	25	50	≤15	25	50
$\alpha = \mathbf{1}; \beta = \mathbf{0}$																		
MILP	**18**	**29**	**29**	**17**	4	9	**7.3**	63.4	68.1	**18**	**27**	**27**	**18**	**16**	**25**	**3.7**	**50.3**	**51.7**
CP$_{AR}$	**18**	**29**	M	8	0	M	12.6	76.4	M	**18**	**27**	M	3	0	M	17.9	71.3	M
CP$_{SR}$	**18**	**29**	**29**	9	**25**	**20**	9.9	**54.9**	**64.5**	**18**	**27**	**27**	5	11	2	11.0	**50.3**	67.0
$\alpha = \mathbf{0}; \beta = \mathbf{1}$																		
MILP	**18**	**29**	18	14	0	7	24.8	83.5	88.3	**18**	**27**	**27**	15	0	2	17.6	82.0	93.6
CP$_{AR}$	**18**	**29**	M	7	0	M	35.4	89.3	M	**18**	**27**	M	8	3	M	41.7	84.4	M
CP$_{SR}$	**18**	**29**	**23**	**18**	**29**	**17**	**22.8**	**71.7**	**88.2**	**18**	**27**	**27**	**18**	**27**	**25**	**11.1**	**52.2**	**78.9**
$\alpha = \xi; \beta = \mathbf{1}$																		
MILP	**18**	**29**	**29**	**14**	3	11	**22.8**	68.6	66.2	**18**	**27**	**27**	**17**	0	8	**14.0**	65.5	59.5
CP$_{AR}$	**18**	**29**	M	5	0	M	48.6	88.8	M	**18**	**27**	M	2	0	M	56.5	94.4	M
CP$_{SR}$	**18**	**29**	**29**	**14**	**26**	**18**	23.7	**58.8**	**64.0**	**18**	**27**	**27**	5	**27**	**19**	25.0	**40.0**	**57.0**

Results. The results are illustrated in Table 2. The mean relative error (MRE) compares the best solution found by a given technique to the best bound found across all techniques; the results in the table take the average of this across all instances solved by the technique.

The MILP displayed fairly strong performance on the distance minimization objective function, particularly for small problems, where the strong bound is able to effectively direct the search, and on long horizon problems, where CP inference is less effective. While the MILP approach is often able prove optimality for small instances when minimizing travel distance, it struggles to produce high quality solutions and meaningful bounds for the fleet minimization objective functions for medium-to-large problems, with optimality gaps close to 100%.

The alternative resource CP model, CP$_{AR}$, encountered memory issues for $|V| = 50$, while the single resource transformation model, CP$_{SR}$, was able to ini-

This is the author's version of an article that is to appear in the Proceedings of the Sixteenth International Conference on the Integration of Constraint Programming, Artficial Intelligence, and Operations Research (CPAIOR2019).

tiate the search for all problems.[3] Overall, it was found that the single resource model outperformed the alternative model for all problem classes. Additionally, CP_{SR} outperforms the MILP formulation on almost all classes of larger problems ($|V| \in \{25, 50\}$), with the exception of distance minimization over long horizons. The horizon augmentation of the single resource model results in large domain sizes for problems with larger initial scheduling horizons, resulting in weaker inference throughout the search. Both CP approaches tend to produce more meaningful bounds for the fleet minimization problems, where the MILP approach has difficulty producing non-trivial lower bounds.

5 A Hybrid Approach

From the experiments, it is evident that MILP outperforms CP_{SR} for distance minimization for large problems over long scheduling horizons. This finding is similar to that from previous research that demonstrated scheduling-based CP models containing optional activities can suffer from poor inference as the problem scales without good upper bounds on horizon length [4]; the authors of this previous work found that seeding CP with high quality solutions found by a different solver can be significantly beneficial.

Given these findings, we construct a hybrid approach that passes the best solution found by the MILP solver to the CP solver as a starting point. Following previous work [4], we allocate half of the runtime to MILP and half to CP noting that the MILP solution improvement diminishes with time. We apply this hybrid to the larger problem instances, $|V| \in \{25, 50\}$. The remainder of the experimental set-up remains as described in the previous section.

Results. We present the results for our hybrid approach, denoted $\text{MILP}\rightarrow\text{CP}_{SR}$, in Table 3, alongside the original MILP and CP_{SR} results. It is apparent that the hybrid approach is beneficial for the distance minimization objective over long horizons, outperforming the other approaches by a wide margin and improving over MILP MRE values by up to 4.4% ($|V| = 50$, long horizon). However, outside of the large, long horizon distance minimization problems, the hybrid provides improvement in few other areas, and is commonly outperformed by the standalone CP method. Based on this observation, we can conclude that it makes sense to hybridize MILP and CP in particular circumstances, but often standalone CP will produce the best result. In these experiments, a CP-based approach (either hybrid or standalone) provides the best results for every problem class, and the hybrid approach outperforms standalone MILP across nearly all problem classes, with the exception of fleet minimization on short horizons.

[3] In fact, the single resource model required significantly less memory for $|V| = 50$ problems than the alternative resource model did for $|V| = 25$ problems.

This is the author's version of an article that is to appear in the Proceedings of the Sixteenth International Conference on the Integration of Constraint Programming, Artficial Intelligence, and Operations Research (CPAIOR2019).

Table 3: Hybrid results, large problems. Best result of column for each objective function in bold. 'M': method ran out of memory before entering the search.

	Short Horizon						**Long Horizon**					
	# Feas.		# Best		MRE (%)		# Feas.		# Best		MRE (%)	
Method	25	50	25	50	25	50	25	50	25	50	25	50
$\alpha = \mathbf{1}; \beta = \mathbf{0}$												
MILP	**29**	**29**	2	5	63.4	68.1	**27**	**27**	5	8	50.3	51.7
CP_{SR}	**29**	**29**	**14**	**14**	54.9	**64.5**	**27**	**27**	8	2	50.3	67.0
MILP$\rightarrow\text{CP}_{SR}$	**29**	**29**	13	10	**54.6**	**64.5**	**27**	**27**	**14**	**17**	**47.3**	**47.3**
$\alpha = \mathbf{0}; \beta = \mathbf{1}$												
MILP	**29**	18	0	7	83.7	87.6	**27**	**27**	0	2	81.8	92.8
CP_{SR}	**29**	**23**	**23**	**15**	**72.0**	**87.5**	**27**	**27**	**27**	**25**	**51.2**	**77.5**
MILP$\rightarrow\text{CP}_{SR}$	**29**	16	11	3	74.0	86.7	**27**	**27**	24	13	53.3	82.3
$\alpha = \xi; \beta = \mathbf{1}$												
MILP	**29**	**29**	0	4	68.9	66.2	**27**	**27**	0	5	65.5	59.5
CP_{SR}	**29**	**29**	**19**	**14**	**59.1**	64.0	**27**	**27**	**16**	**17**	**40.0**	57.0
MILP$\rightarrow\text{CP}_{SR}$	**29**	**29**	10	11	59.8	**62.9**	**27**	**27**	11	5	47.9	**55.3**

6 Conclusion and Perspective

In this paper we presented the first approaches for solving the Electric Vehicle Routing Problem with Time Windows (EVRPTW) using constraint programming (CP). We present two scheduling-based CP formulations: the initial model uses an alternative resource technique previously applied to other routing problems, while the second uses a single resource transformation for CP models using optional activities, sequence variables, and cumulative function expressions. We detail techniques used to strengthen the formulations and discuss alternate modeling strategies.

Numerical results indicate the superiority of the single resource CP model over the alternative resource model, for all problems, and the MILP formulation, for the majority of medium-to-large problem classes. Recognizing the ability of MILP to quickly produce good quality solutions for large distance minimization problems with long scheduling horizons, we also investigate a hybrid MILP-CP approach where the best solution from the mathematical programming solver is used to seed the CP search. Results indicate the hybrid approach outperforms both of the standalone techniques for the problems that motivated the effort, but is not beneficial overall.

Given the growth of electric vehicle (EV) adoption in the logistics and consumer automotive industries, we believe the study of these problems in the context of CP modeling and solving is a strategic direction. Outside of EVs and transportation, there is considerable opportunity in the highly related field of multi-robot task allocation (MRTA) [16, 5]. Future work will investigate the applicability of the techniques developed in this paper to problems found in MRTA.

Acknowledgment We would like to thank the anonymous reviewers whose detailed feedback helped improve the paper.

References

1. Afroditi, A., Boile, M., Theofanis, S., Sdoukopoulos, E., Margaritis, D.: Electric vehicle routing problem with industry constraints: trends and insights for future research. Transportation Research Procedia 3, 452–459 (2014)
2. Beck, J.C., Prosser, P., Selensky, E.: Vehicle routing and job shop scheduling: What's the difference? In: International Conference on Automated Planning and Scheduling. pp. 267–276 (2003)
3. Berbeglia, G., Cordeau, J.F., Laporte, G.: A hybrid tabu search and constraint programming algorithm for the dynamic dial-a-ride problem. INFORMS Journal on Computing 24(3), 343–355 (2012)
4. Booth, K.E.C., Do, M., Beck, J.C., Rieffel, E., Venturelli, D., Frank, J.: Comparing and integrating constraint programming and temporal planning for quantum circuit compilation. In: International Conference on Automated Planning and Scheduling. pp. 366–374. AAAI Press (2018)
5. Booth, K.E.C., Nejat, G., Beck, J.C.: A constraint programming approach to multi-robot task allocation and scheduling in retirement homes. In: International Conference on Principles and Practice of Constraint Programming. pp. 539–555. Springer (2016)
6. Booth, K.E.C., Tran, T.T., Nejat, G., Beck, J.C.: Mixed-integer and constraint programming techniques for mobile robot task planning. IEEE Robotics and Automation Letters 1(1), 500–507 (2016)
7. Bruglieri, M., Pezzella, F., Pisacane, O., Suraci, S.: A variable neighborhood search branching for the electric vehicle routing problem with time windows. Electronic Notes in Discrete Mathematics 47, 221–228 (2015)
8. Cappart, Q., Thomas, C., Schaus, P., Rousseau, L.M.: A constraint programming approach for solving patient transportation problems. In: International Conference on Principles and Practice of Constraint Programming. pp. 490–506. Springer (2018)
9. Dekker, R., Bloemhof, J., Mallidis, I.: Operations research for green logistics–an overview of aspects, issues, contributions and challenges. European Journal of Operational Research 219(3), 671–679 (2012)
10. Desaulniers, G., Errico, F., Irnich, S., Schneider, M.: Exact algorithms for electric vehicle-routing problems with time windows. Operations Research 64(6), 1388–1405 (2016)
11. Di Gaspero, L., Rendl, A., Urli, T.: Balancing bike sharing systems with constraint programming. Constraints 21(2), 318–348 (2016)
12. Erdoğan, S., Miller-Hooks, E.: A green vehicle routing problem. Transportation Research Part E: Logistics and Transportation Review 48(1), 100–114 (2012)
13. Felipe, Á., Ortuño, M.T., Righini, G., Tirado, G.: A heuristic approach for the green vehicle routing problem with multiple technologies and partial recharges. Transportation Research Part E: Logistics and Transportation Review 71, 111–128 (2014)
14. Feng, W., Figliozzi, M.: An economic and technological analysis of the key factors affecting the competitiveness of electric commercial vehicles: A case study from the usa market. Transportation Research Part C: Emerging Technologies 26, 135–145 (2013)
15. Gedik, R., Kirac, E., Milburn, A.B., Rainwater, C.: A constraint programming approach for the team orienteering problem with time windows. Computers & Industrial Engineering 107, 178–195 (2017)

16. Gerkey, B.P., Matarić, M.J.: A formal analysis and taxonomy of task allocation in multi-robot systems. The International Journal of Robotics Research 23(9), 939–954 (2004)
17. Hiermann, G., Puchinger, J., Ropke, S., Hartl, R.F.: The electric fleet size and mix vehicle routing problem with time windows and recharging stations. European Journal of Operational Research 252(3), 995–1018 (2016)
18. de Jong, C., Kant, G., Van Vlient, A.: On finding minimal route duration in the vehicle routing problem with multiple time windows. Manuscript, Department of Computer Science, Utrecht University, Holland (1996)
19. Keskin, M., Çatay, B.: Partial recharge strategies for the electric vehicle routing problem with time windows. Transportation Research Part C: Emerging Technologies 65, 111–127 (2016)
20. Kinable, J., van Hoeve, W.J., Smith, S.F.: Optimization models for a real-world snow plow routing problem. In: International Conference on AI and OR Techniques in Constriant Programming for Combinatorial Optimization Problems. pp. 229–245. Springer (2016)
21. Laborie, P.: IBM ILOG CP Optimizer for detailed scheduling illustrated on three problems. In: International Conference on AI and OR Techniques in Constriant Programming for Combinatorial Optimization Problems. pp. 148–162. Springer (2009)
22. Laborie, P., Rogerie, J., Shaw, P., Vilím, P.: IBM ILOG CP Optimizer for scheduling. Constraints 23(2), 210–250 (2018)
23. Lam, E., Van Hentenryck, P., Kilby, P.: Joint vehicle and crew routing and scheduling. In: International Conference on Principles and Practice of Constraint Programming. pp. 654–670. Springer (2015)
24. Lin, C., Choy, K.L., Ho, G.T., Chung, S.H., Lam, H.: Survey of green vehicle routing problem: past and future trends. Expert Systems with Applications 41(4), 1118–1138 (2014)
25. Liu, C., Aleman, D.M., Beck, J.C.: Modelling and solving the senior transportation problem. In: International Conference on the Integration of Constraint Programming, Artificial Intelligence, and Operations Research. pp. 412–428. Springer (2018)
26. Luè, A., Colorni, A., Nocerino, R., Paruscio, V.: Green move: An innovative electric vehicle-sharing system. Procedia-Social and Behavioral Sciences 48, 2978–2987 (2012)
27. Montoya, A., Guéret, C., Mendoza, J.E., Villegas, J.G.: The electric vehicle routing problem with nonlinear charging function. Transportation Research Part B: Methodological 103, 87–110 (2017)
28. Rezvani, Z., Jansson, J., Bodin, J.: Advances in consumer electric vehicle adoption research: A review and research agenda. Transportation research part D: transport and environment 34, 122–136 (2015)
29. Schneider, M., Stenger, A., Goeke, D.: The electric vehicle-routing problem with time windows and recharging stations. Transportation Science 48(4), 500–520 (2014)
30. Shaw, P.: Using constraint programming and local search methods to solve vehicle routing problems. In: International Conference on Principles and Practice of Constraint Programming. pp. 417–431. Springer (1998)
31. Vali, M., Salimifard, K.: A constraint programming approach for solving multiple traveling salesman problem. In: The Sixteenth International Workshop on Constraint Modelling and Reformulation (2017)