A parallel matheuristic for the technician routing problem with conventional and electric vehicles

Jorge E. Mendoza, Alejandro Montoya, Christelle Guéret, Juan G. Villegas

A parallel matheuristic for the technician routing problem with conventional and electric vehicles

Jorge E. Mendoza[1], Alejandro Montoya[2], Christelle Guéret[3], Juan G. Villegas[4]

[1] Université de Tours - LI (EA6300) - ROOT (ERL CNRS 6305)
64 Av. Jean Portalis, 37200 Tours, France
jorge.mendoza@univ-tours.fr

[2] Departamento de Ingeniería de Producción, Universidad EAFIT
Carrera 49 No. 7 Sur - 50, Medellín, Colombia
jmonto36@eafit.edu.co

[3] Université d'Angers - LARIS (EA 7315)
62 avenue Notre Dame du Lac, 49000 Angers, France
christelle.gueret@univ-angers.fr

[4] Departamento de Ingeniería Industrial, Universidad de Antioquia
Calle 70 No. 52-21, Medellín, Colombia
juan.villegas@udea.edu.co

Abstract

The technician routing problem with conventional and electric vehicles (TRP-CEV) consists in designing service routes taking into account the customers' time windows and the technicians' skills, shifts, and lunch breaks. In the TRP-CEV routes are covered using a fixed and heterogeneous fleet of conventional and electric vehicles (EVs). Due to their relatively limited driving ranges, EVs may need to include in their routes one or more recharging stops. In this talk we present a parallel matheuristic for the TRP-CEV. The approach works in two phases. In the first phase it decomposes the problem into a number of "easier to solve" vehicle routing problems with time windows and solves these problems in parallel using a GRASP. During the execution of this phase, the routes making up the local optima are stored in a long-term memory. In the second phase, the approach uses the routes stored in the long-term memory to assemble a solution to the TRP-CEV. We discuss computational experiments carried on real-world TRP-CEV instances provided by a French public utility and instances for the closely-related electric fleet size and mix vehicle routing problem with time windows and recharging stations taken from the literature.

1 Introduction

The technician routing problem with conventional and electric vehicles (TRP-CEV) can be defined on a directed and complete graph $G = (\mathcal{N}, \mathcal{E})$ where $\mathcal{N}$ is the set of nodes and $\mathcal{E}$ is the set of edges. The set of nodes is defined as $\mathcal{N} = \{0\} \cup \mathcal{C} \cup \mathcal{S}$, where node 0 represents the depot, $\mathcal{C}$ is a set of nodes representing the customers, and $\mathcal{S}$ is a set of nodes representing the charging stations (CSs) where the electric vehicles can recharge their batteries. Each customer $i \in \mathcal{C}$ has a request demanding a skill k_i from a set $\mathcal{K}$ and having a service time p_i and a time window $[ec_i, lc_i]$, where ec_i and lc_i are the earliest and latest possible service start times. For the sake of simplicity, in the remainder of this extended abstract we use the terms customer and request interchangeably. The set of technicians is denoted as $\mathcal{T}$. Each technician $t \in \mathcal{T}$ has: a fixed "utilization" cost c_t; a subset of skills $\mathcal{K}_t \subseteq \mathcal{K}$; a shift $[es_t, ls_t]$, where es_t is the technician's earliest possible departure time from the depot and ls_t is the technician's latest return time to the depot; a lunch break that must start at el_t and end at ll_t; and an *energy consumption factor* cf_t associated to the technician's driving profile (e.g., sportive, normal, eco). To cover their routes, the technicians drive vehicles from a fixed fleet composed by different types of conventional and electric vehicles (hereafter CVs and EVs). The set of vehicle types is defined as $\mathcal{V} = \mathcal{V}_c \cup \mathcal{V}_e$, where $\mathcal{V}_c$ is the set of CV types and $\mathcal{V}_e$ is the set of EV types. For each vehicle type $v \in \mathcal{V}$ there is a unitary travel cost tc_v (expressed in $€/km$) and a fixed and limited number of vehicles m_v. Vehicles of type $v \in \mathcal{V}_e$ additionally have: a fixed cost

gc_v for recharging the battery (expressed in €)[1], a battery capacity Q_v (expressed in kWh), a set $\mathcal{S}_v \subseteq \mathcal{S}$ of compatible CSs, and a discrete and non-linear charging function f_{vs} describing the relation between the vehicle's charging time and *state of charge* (SoC) at station $s \in \mathcal{S}_v$. Depending on the context we refer to the SoC as the amount of remaining energy (in kWh) or as the percentage of remaining battery capacity. The charging function is defined as $f_{vs} = \{a_b | b \in \{0, 1, \ldots, 100\}\}$ were a_b is the time needed to take the SoC from 0 to b percent of Q. Finally, set $\mathcal{E} = \{(i, j) : i, j \in \mathcal{N}, i \neq j\}$ denotes the set of arcs connecting nodes in $\mathcal{N}$. Each arc $(i, j) \in \mathcal{E}$ has three associated nonnegative values: a travel time tt_{ij}, a distance d_{ij}, and a nominal energy consumption e_{ijv} for each type of EV $v \in \mathcal{V}_e$.

In the TRP-CEV the objective is to find a set of routes of minimum total cost. The latter is defined as the sum of i) travel costs, ii) fixed charging costs, iii) parking costs, and iv) technician utilization costs. The planned set of routes must satisfy the following constraints: each request is served exactly once within its time window by a technician with the required skill; the level of the battery when the EVs arrive at any vertex is nonnegative; the EVs only charge at compatible CSs; each technician works only during his or her shift; each technician takes the lunch break at the pre-defined times; the number of CVs and EVs used are less or equal than $|\mathcal{V}_c|$ and $|\mathcal{V}_e|$; and each route starts and ends at the depot.

2 Parallel matheuristic

Algorithm 1 describes the general structure of our parallel matheuristic (here after referred to as PMa). The algorithm starts by calling procedure `groupTechnicians(`$\mathcal{T}$`)` – line 2. This procedure groups the technicians sharing the same characteristics (i.e., skills, fixed utilization cost, energy consumption factor, shift, and lunch break) and generates the set $\mathcal{TP}$ of *technician profiles*. Then, the algorithm invokes procedure `buildAssignments(`$\mathcal{TP}, \mathcal{V}$`)` – line 3. The latter builds the set $\mathcal{A}$ containing all possible technician profile-vehicle type assignments. Note that $|\mathcal{A}| = |\mathcal{TP}| \times |\mathcal{V}_c| + |\mathcal{TP}| \times |\mathcal{V}_e|$. Then, the algorithm starts the parallel phase – lines 5 to 8. For each assignment $a \in \mathcal{A}$ the algorithm solves, on a dedicated thread, a vehicle routing problem with time windows and lunch breaks (VRP-TWLB). Let $p(a) \in \mathcal{TP}$ and $v(a) \in \mathcal{V}$ be the technician profile and the type of vehicle involved in assignment a. In the VRP-TWLB for assigment a we assume that i) the fleet is unlimited and composed only of vehicles of type $v(a)$ and that we have an unlimited number of technicians with profile $p(a)$. If $v(a)$ is an EV, then the resulting problem is an electric VRP-TWLB. To solve the $|\mathcal{TP}| \times |\mathcal{V}_c|$ VRPs-TWLB and the $|\mathcal{TP}| \times |\mathcal{V}_e|$ eVRPs-TWLB our approach relies on a GRASP (line 6). The GRASP slightly varies depending on the type of problem being solved (VRP-TWLB or eVRP-TWLB). Figure 1 depicts the components embedded in the two versions. The GRASP returns a set Ω_a containing all the routes found in the local optima reached during the algorithm's execution. The routes in Ω_a join the *long term memory* structure Ω (line 7). After completing the parallel phase, the algorithm calls procedure `setCovering(`$G, \Omega, \mathcal{V}, \mathcal{A}$`)` – line 9 –, which solves an extended set covering formulation over Ω to find a feasible TRP-CEV solution. It is worth noting that it is only at this point that we take into account the constraints on the number of technicians and vehicles. Full details on the GRASP components, specially those used to solve the eVRPs-TWLB, and the parallel implementation will be discussed in the talk.

3 Computational experiments

We implemented our PMa in Java (jre V.1.8.0) and used Gurobi Optimizer (version 6.0) to solve the set covering formulation (Algorithm 1, line 9) and to explore the Global Charging Improvement neighborhood (bottom-right of Figure 3). We ran experiments on two sets of instances. The first set is made up of 24 "real-world" TRP-CEV instances (10 small with proven optima + 14 large) built using data provided by French electricity giant ENEDIS. The reader can find a full description of these instances in [2, chap. 5]. For the 10 small instances our PMa was able to find the optimal solutions. For the remaining 14

[1]This cost accounts for the long-term battery degradation cost

Algorithm 1 Parallel matheuristic: general structure

function ParallelMatheuristic(G,$\mathcal{T}$,$\mathcal{V}$)
  $E \longleftarrow$ `groupTechnicians`($\mathcal{T}$)
  $P \longleftarrow$ `buildAssignments`($\mathcal{TP}$, $\mathcal{V}$)
  $\Omega \longleftarrow \emptyset$
  parallel for each $a \in \mathcal{A}$
    $\Omega_a \longleftarrow$ `GRASP`(a, G)
    $\Omega \longleftarrow \Omega \cup \Omega_a$
  end for
  $\sigma \longleftarrow$ `setCovering`(G, Ω, $\mathcal{V}$, $\mathcal{A}$)
  return σ
end function

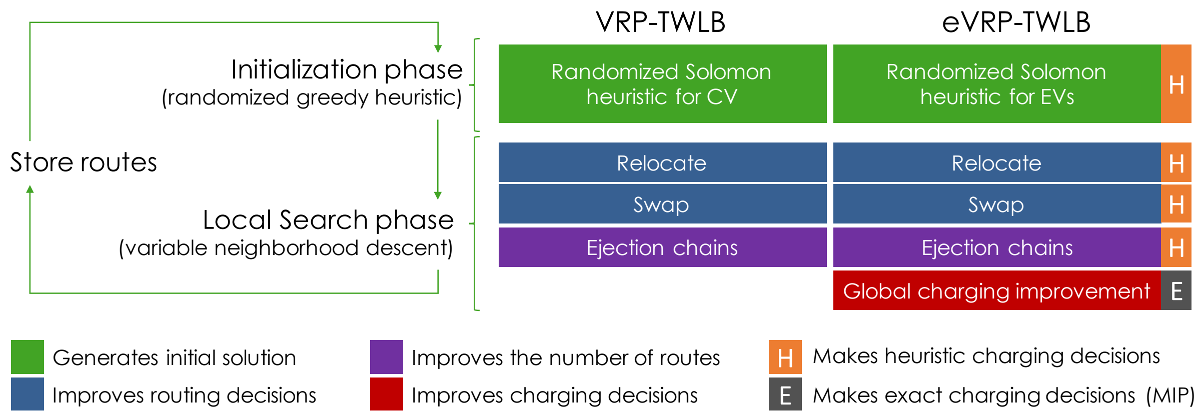

Figure 1: Algorithmic components embedded in the two GRASP versions

instances our method reported average improvements of 6.5% with respect to the solutions delivered by the commercial software currently used at ENEDIS. The second set consists of the 276 instances (108 small with proven optima + 168 large) proposed in [1] for the closely-related electric fleet size and mix vehicle routing problem with time windows and recharging stations. Although our method was not tailored for this problem, it was able to deliver competitive perfomances with respect to the state-of-the-art Adaptive Large Neighborhood Search (ALNS) proposed in [1]. On the small instances our PMa found 81/108 optimal solutions and reported better avg. gaps (0.32% vs 0.55%) and execution times (0.06min vs. 0.32min) than ALNS. On the large instances, our method unveiled 61 new best known solutions but reported larger avg. gaps than ALNS (2.20% vs. 1.18%) with comparable execution times.

References

[1] G. Hiermann, J. Puchinger, S. Ropke, and R. F. Hartl. 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.

[2] A. Montoya. *Electric Vehicle Routing Problems: models and solution approaches*. PhD thesis, Université d'Angers, 2016.