European Journal of Operational Research 247 (2015) 782–787

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Production, Manufacturing and Logistics

A speed and departure time optimization algorithm for the pollution-routing problem

Raphael Kramer[a,*], Nelson Maculan[b], Anand Subramanian[c], Thibaut Vidal[d]

[a] Università degli Studi di Modena e Reggio Emilia, Dipartimento di Scienze e Metodi dell'Ingegneria, Italy
[b] Universidade Federal do Rio de Janeiro, Departamento de Engenharia de Sistemas e Computação, Brazil
[c] Universidade Federal da Paraíba, Departamento de Engenharia de Produção, Brazil
[d] Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, Brazil

ARTICLE INFO

Article history:
Received 29 January 2015
Accepted 15 June 2015
Available online 20 June 2015

Keywords:
Routing
Speed optimization
Departure optimization
Green logistics

ABSTRACT

We propose a new speed and departure time optimization algorithm for the pollution-routing problem (PRP), which runs in quadratic time and returns an optimal schedule. This algorithm is embedded into an iterated local search-based metaheuristic to achieve a combined speed, scheduling and routing optimization. The start of the working day is set as a decision variable for individual routes, thus enabling a better assignment of human resources to required demands. Some routes that were evaluated as unprofitable can now appear as viable candidates later in the day, leading to a larger search space and further opportunities of distance optimization via better service consolidation. Extensive computational experiments on available PRP benchmark instances demonstrate the good performance of the algorithm. The flexible departure times from the depot contribute to reduce the operational costs by 8.36% on the considered instances.

1. Introduction

The pollution-routing problem (PRP) is a variant of the vehicle routing problem with environmental considerations, introduced in Bektaş and Laporte (2011), and aiming to minimize operational and environmental costs subject to vehicle capacity and hard time-window constraints. The costs are based on driver wages and fuel consumption, evaluated as a non-linear function of the distance traveled, vehicle load, and vehicle speed. Some recent articles have proposed heuristics for the PRP: an adaptive large neighborhood search in Demir, Bektaş and Laporte (2012), and an ILS with a set-partitioning matheuristic in Kramer, Subramanian, Vidal and Cabral (2015). Other contributions (Bektaş & Laporte, 2011; Franceschetti, Honhon, Van Woensel, Bektaş & Laporte, 2013; Dabia, Demir & Van Woensel, 2014) focused on mathematical formulations and integer programming algorithms based on branch-and-price.

Vehicle-speed decisions play an important role in the PRP, since they do not only affect the total cost, but also the travel times between the locations, with a large impact on time-window feasibility. For this reason, most algorithms for the PRP perform – at regular times during the search – an optimization of vehicle speeds for the current routes. The resulting speed optimization subproblem (SOP), seeks to find the most cost-efficient arc speeds on a given route while respecting arrival-time constraints at each customer.

Some algorithms for the SOP have been recently proposed (Demir et al., 2012; Kramer et al., 2015; Norstad, Fagerholt & Laporte, 2011; Hvattum, Norstad, Fagerholt & Laporte, 2013). These algorithms run in quadratic time, consider identical cost/speed functions for each arc, and assume that the departure time is fixed. Now, considering that the start of the working day (the departure time from the depot) is a decision variable leads to different optimality conditions and speed decisions. This may open the way to significant cost reductions, but also increases resolution complexity. Fewer articles have addressed this aspect. In Dabia et al. (2014), the departure time from the first customer is optimized by means of a golden section search, within a pricing algorithm. Franceschetti et al. (2013) model and solve the PRP with time-dependent travel times, a generalization of the problem considered in this paper. The resulting speed optimization algorithm is, however, more complex due to the presence of three time intervals with different speed/cost functions, involving 24 rules for speed choices. The solution is also not guaranteed to be optimal. Finally, Vidal, Jaillet and Maculan (2014) showed that the optimal SOP solution with deadlines and arc-dependent speed/cost functions can be achieved by solving a hierarchy of resource allocation problems.

This article contributes to the resolution of difficult vehicle routing variants with speed and departure time optimization. We

* Corresponding author. Tel.: +55 8399073669.
E-mail address: raphael.kramer@unimore.it (R. Kramer).

http://dx.doi.org/10.1016/j.ejor.2015.06.037
0377-2217/

consider the case where driver wages are computed from the departure time, hence allowing to better assign human resources to needed deliveries. Note that a fixed departure time can still be obtained by reducing the departure time window to a point. Some routes that were evaluated as unprofitable with a fixed departure time policy can now appear as viable candidates, leading to a larger search space and further opportunities of routing optimization via better service clustering.

We introduce a simple quadratic time algorithm for the speed and departure time optimization (Section 3). Moreover, we demonstrate the optimality of this algorithm (Section 4). The speed optimization algorithm is embedded into a vehicle routing matheuristic to produce high-quality routing plans. We conduct computational experiments on the classic PRP instances to evaluate the performance of the method, and assess the impact of departure time optimization on cost and pollution emissions (Section 5). The results highlight very significant routing cost reduction, of 8.36% on average. The CPU time of the new metaheuristic remains comparable to current state-of-the-art methods despite the fact that it deals with a more general problem.

2. Problem description

The PRP with flexible departure times can be defined as follows. Let $G = (\mathcal{V}, \mathcal{A})$ be a complete and directed graph with a set $\mathcal{V} = \{0, 1, 2, \ldots, n\}$ of vertices and a set $\mathcal{A} = \{(i, j) : i, j \in V, i \neq j\}$ of arcs. Vertex 0 represents the depot while the remaining vertices are customers. A set of m vehicles with capacity Q is available to service the customers. Each customer i has a non-negative demand q_i, a time window $[a_i, b_i]$ during which service must start, and a service time τ_i. By convention, $q_0 = \tau_0 = 0$ for the depot. Each arc $(i, j) \in \mathcal{A}$ represents a travel possibility from node i to j for a distance d_{ij}, which can be traveled with any speed v_{ij} in the interval $[v_{min}, v_{max}]$. The PRP aims at determining a speed matrix $(\mathbf{v})_{ij}$ for the arcs and a set of feasible routes $\mathbf{R}$ to serve all customers while minimizing environmental and operational costs.

Let $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \ldots, \sigma_{|\sigma|})$ be a route, $f_{\sigma_i\sigma_{i+1}}$ be the vehicle load on arc (σ_i, σ_{i+1}) and t_{σ_i} be the arrival time at customer σ_i. The environmental cost is proportional to fuel consumption, computed as in Eq. (1) where w_1, w_2, w_3, w_4 are parameters based on fuel properties, vehicle and network characteristics. The labor costs are proportional to route duration, computed as the difference between departure and arrival time $t_{\sigma_{|\sigma|}} - t_{\sigma_1}$. Defining ω_{FC} as the fuel cost per liter and ω_{DC} as the driving cost per second, the objective of the PRP is given in Eq. (2):

$$F^{\text{F}}_{\sigma_i\sigma_{i+1}}(v_{\sigma_i\sigma_{i+1}}) = d_{\sigma_i\sigma_{i+1}}\left(\frac{w_1}{v_{\sigma_i\sigma_{i+1}}} + w_2 + w_3 f_{\sigma_i\sigma_{i+1}} + w_4 v^2_{\sigma_i\sigma_{i+1}}\right) \quad (1)$$

$$Z_{\text{PRP}}(\mathbf{R}, \mathbf{v}) = \sum_{\boldsymbol{\sigma}\in\mathbf{R}}\left(\omega_{\text{FC}}\sum_{i=1}^{|\sigma|-1} F^{\text{F}}_{\sigma_i\sigma_{i+1}}(v_{\sigma_i\sigma_{i+1}}) + \omega_{\text{FD}}\,(t_{\sigma_{|\sigma|}} - t_{\sigma_1})\right). \quad (2)$$

3. The proposed speed and departure time optimization algorithm

This section deals with the optimization of speeds and departure times for a fixed route σ. To simplify the exposition, we will omit σ in the notations, and thus assume that customers are indexed by their order of appearance in the route.

The fuel consumption per distance unit $F^{\text{F}}_{i,i+1}(v_{i,i+1})$ is a convex function. The speed value v^*_{F} that minimizes fuel costs is given in Eq. (3). Similarly, for any arc $(i, i+1)$, assuming that there is no waiting time in the route after i, the speed value v^*_{FD} that minimizes fuel plus driver costs is expressed in Eq. (4):

$$\frac{dF^{\text{F}}_{i,i+1}}{dv_{i,i+1}}(v^*_{\text{F}}) = 0 \Leftrightarrow v^*_{\text{F}} = \left(\frac{w_1}{2w_4}\right)^{1/3} \quad (3)$$

$$v^*_{\text{FD}} = \left(\frac{\frac{\omega_{\text{DC}}}{\omega_{\text{FC}}} + w_1}{2w_4}\right)^{1/3}. \quad (4)$$

For a fixed route, the speed and departure time optimization problem consists of finding the departure time from the depot and the optimal speeds for each arc while respecting customers' time windows. To solve this problem, we propose an optimal recursive algorithm that extends those presented in Demir et al. (2012); Hvattum et al. (2013) and Kramer et al. (2015). It solves in quadratic time a special case of the time-dependent SOP of Franceschetti et al. (2013).

The algorithm relies on a general divide-and-conquer strategy, which iteratively solves a relaxed SOP obtained by ignoring time windows at intermediate destinations. If the resulting solution satisfies all constraints, then it is returned. Otherwise the customer p with maximum time-window violation is identified and its arrival time is set to its closest feasible value. Fixing this decision variable creates two sub-problems which are recursively solved (Algorithm 1, lines 20–21). The novelty of this algorithm is the way it manages departure or arrival-time fixing within subproblems to converge towards optimal departure and speed decisions.

Algorithm 1 Speed and departure-time optimization algorithm (SDTOA).

1: Procedure $SDTOA(s, e)$
2: $p \leftarrow violation \leftarrow maxViolation \leftarrow 0$
3: $D \leftarrow \sum_{i=s}^{e-1} d_{i,i+1}$
4: $T \leftarrow \sum_{i=s}^{e-1} \tau_i$
5: **if** $s = 1$ **and** $e = n_\sigma$ **then**
6: $\quad t_1 = a_1$
7: **if** $e = n_\sigma$ **then**
8: $\quad t_e = \min\{\max\{a_e, t_s + D/v^*_{\text{FD}} + T\}, b_e\}$
9: **if** $s = 1$ **then**
10: $\quad t_s = \min\{\max\{a_s, t_e - D/v^*_{\text{FD}} - T\}, b_s\}$
11: $v_{\text{REF}} \leftarrow D/(t_e - t_s - T)$
12: **for** $i = s + 1 \ldots e$ **do**
13: $\quad t_i = t_{i-1} + \tau_{i-1} + d_{i-1,i}/v_{\text{REF}}$
14: $\quad violation = \max\{0, t_i - b_i, a_i - t_i\}$
15: $\quad$ **if** $violation > maxViolation$ **then**
16: $\quad\quad maxViolation = violation$
17: $\quad\quad p = i$
18: **if** $maxViolation > 0$ **then**
19: $\quad t_p = \min\{\max\{a_p, t_p\}, b_p\}$
20: $\quad SDTOA(s, p)$
21: $\quad SDTOA(p, e)$
22: **if** $s = 1$ **and** $e = n_\sigma$ **then**
23: $\quad$ **for** $i = 2 \ldots n_\sigma$ **do**
24: $\quad\quad v_{i-1,i} = max\{d_{i-1,i}/(t_i - t_{i-1} - \tau_{i-1})), v^*_{\text{F}}\}$

For a route with n_σ nodes (including the departure and return to the depot), Algorithm 1 is applied by setting the *start* s to 1 and the *end* e to n_σ. The departure time is first set to the earliest possible value $t_1 = a_1$ (Algorithm 1, line 6). This decision will be revised later on. The arrival times at each customers are then derived as follows. The arrival time at the last customer when traveling at speed v^*_{FD} is determined and, in case of violation, updated to its closest time-window bound (Algorithm 1, line 8). This leads to a reference speed v_{REF} on the route (Algorithm 1, line 11) which is used to compute the arrival time at each customer as well as the maximum time-window violation (Algorithm 1, lines 12–17).

In case of violation, two subproblems are recursively solved. Any subproblem starting at the depot is now solved without fixing the departure time. Indeed, the arrival time to the last customer of this sub-problem is already fixed, such that it is possible to evaluate the reference speed "backwards", deriving the best departure time at the depot (Algorithm 1, line 10), and the customer arrival times. The other sub-problems are similarly solved. The recursion is repeated until all constraints are satisfied. Finally, when arrival times are known for all customers, the associated speeds are revised in such a way that any