Polynomial-time algorithms to compute violation in the robust vehicle routing problem with time windows and budget uncertainty

Igor Malheiros[1,2], Michaël Poss[2], Vitor Nesello[1], Artur Pessoa[3], Anand Subramanian[4]

[1] Atoptima, France
{igor.malheiros,vitornesello}@atoptima.com
[2] LIRMM, France
michael.poss@lirmm.fr
[3] Universidade Federal Fluminense, Brazil
arturpessoa@id.uff.br
[4] Universidade Federal da Paraíba, Brazil
anand@ci.ufpb.br

Keywords : *robust optimisation, vehicle routing, dynamic programming.*

1 Introduction

The *vehicle routing problem* (VRP) *with time windows* (VRPTW) is one of the most studied VRP variants [2] due to its numerous practical applications emerging from real-life scenarios.

This work addresses the VRPTW and uncertain travel times under the robust optimization paradigm [1]. The VRPTW is defined on a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, in which $\mathcal{V} = \mathcal{V}^* \cup \{o, d\}$, $\mathcal{V}^*$ is the set of customers to be visited, o is the origin depot and d is the destination depot, whereas $\mathcal{A}$ is the set of arcs. Each vertex $i \in \mathcal{V}$ has a demand q_i and with a time window $[e_i, l_i]$. Each arc has a travel cost c_{ij} and a travel time t_{ij}. In addition, let $\mathcal{K}$ represent the set of homogeneous vehicles with capacity Q.

In our *robust VRPTW* (RVRPTW) model, travel times belong to a known finite set of non-negative vectors for handling the uncertainty. Specifically, let $\bar{t}_{ij}$ and $\hat{t}_{ij}$ denote the nominal travel time and the deviation, respectively, for arc $(i, j) \in \mathcal{A}$, and $\Gamma \in \mathbb{Z}_+$ be a non-negative integer. We introduce the auxiliary vector $\delta \in \{0, 1\}^{|\mathcal{A}|}$ such that the travel time vector is given by $t_{ij}(\delta) = \bar{t}_{ij} + \delta_{ij}\hat{t}_{ij}$ for each of its components $(i, j) \in \mathcal{A}$, and consider the set of scenarios

$$\Delta_\Gamma = \left\{ \delta \in \{0,1\}^{|\mathcal{A}|} : \sum_{(i,j)\in\mathcal{A}} \delta_{ij} \leq \Gamma \right\}.$$

Let x_{ij}^k be the binary variable that is equal to 1 iff the vehicle $k \in K$ traverses the arc $(i, j) \in \mathcal{A}$, and let $y_i(\delta)$ be a non-negative real variable indicating the arrival time at vertex i for a given uncertainty parameter $\delta \in \Delta_\Gamma$. We describe the RVRPTW as follows:

$$\min \sum_{k\in\mathcal{K}} \sum_{(i,j)\in\mathcal{A}} c_{ij} x_{ij}^k \tag{1}$$

$$\text{s.t. } (x_{ij}^k = 1) \implies (y_j(\delta) \geq y_i(\delta) + t_{ij}(\delta)), \quad (i,j) \in \mathcal{A}, k \in \mathcal{K}, \forall \delta \in \Delta_\Gamma \tag{2}$$

$$e_i \leq y_i(\delta) \leq l_i, \quad i \in \mathcal{V}, \forall \delta \in \Delta_\Gamma \tag{3}$$

$$x \in \mathcal{X}, y \in \mathbb{R}_+, \tag{4}$$

where $\mathcal{X} \subseteq \{0,1\}^{|\mathcal{A}|\times|\mathcal{K}|}$ is the set of feasible paths starting at o and ending at d, and covering all nodes of $\mathcal{V}^*$ exactly once.

In local search-based algorithms, one is interested in checking whether a given path π is feasible, or in quantifying its infeasibility if π is not feasible.

Let π contains the nodes $(0, \ldots, n)$. Moreover, let $a_i(\delta)$ be the arrival time at vertex i for scenario $\delta \in \Delta_\Gamma$, which is defined as $a_i(\delta) = e_0$, if $i = 0$ and $\max\{e_i, a_{i-1}(\delta) + t_{i-1,i}(\delta)\}$, otherwise. The violation at node i for $\delta \in \Delta_\Gamma$ can be defined as $f(\max\{a_i(\delta) - l_i\}^+)$ for some non-decreasing function f, such that $f(0) = 0$ and $f(t) > 0$ for $t > 0$, and the violation in π is given by $V(\delta) = \sum_{i=1}^{n} f(\max\{a_i(\delta) - l_i\}^+)$. Overall, the worst-case violation is defined as

$$V^* = \max_{\delta \in \Delta_\Gamma} V(\delta).$$

The work aims to study the computation of V^* for two cases of functions f: (i) the *total tardiness*, where $f(t) = t$, and (ii) the *number of failures*, where $f(0) = 0$ and $f(t) = 1$ if $t > 0$.

2 Polynomial-time algorithms

Number of failures The total number of failures of a path π is bounded by n. This allows one to design a recursive function where $\alpha(i, \gamma, \phi)$ represents the maximum arrival times given that $\phi = 0, \ldots, i$ failures have occurred along the nodes $1, \ldots, i$ and using $\gamma = 0, \ldots, \Gamma$ deviations so far. Therefore, it is possible to implement a dynamic programming algorithm to maximize the total number of failures with a time complexity of $\mathcal{O}(n^2\Gamma)$.

Total tardiness We compute the maximum total tardiness using a dynamic programming algorithm described next. For each $i = 1, \ldots, n$ and $\gamma = 0, \ldots, \Gamma$, we denote by Δ_γ^i the projection of Δ_γ on the components $1, \ldots, i$. Furthermore, for each $i = 1, \ldots, n$, we denote by $\tau_i(\delta) = \max\{a_i(\delta) - l_i, 0\}$ the tardiness at vertex i, and by $\tau^i(\delta) = \sum_{j=0}^{i} \tau_j(\delta)$ the sum over the tardiness for the first i vertices only. Observe that $\tau^i(\delta)$ depends only on the first i components of δ.

Our dynamic programming algorithm is based on the value-function

$$F(i, \gamma, \beta) = \max_{\delta \in \Delta_\gamma^i} \tau^i(\delta) + \beta \cdot \hat{t}_i(\delta),$$

where $\beta = 0, \ldots, n$. Observe that the optimal solution is given by $F(n, \Gamma, 0)$.

Overall, all values of $F(i, \gamma, \beta)$ can be computed in $\mathcal{O}(n^2\Gamma)$.

References

[1] Agostinho Agra, Marielle Christiansen, Rosa Figueiredo, Lars Magnus Hvattum, Michael Poss, and Cristina Requejo. The robust vehicle routing problem with time windows. *Computers & operations research*, 40(3):856–866, 2013.

[2] Guy Desaulniers, Oli BG Madsen, and Stefan Ropke. Chapter 5: The vehicle routing problem with time windows. In *Vehicle Routing: Problems, Methods, and Applications, Second Edition*, pages 119–159. SIAM, 2014.