Resource Extension Functions: Properties, Inversion, and Generalization to Segments

Stefan Irnich [a,*]

[a] *Deutsche Post Endowed Chair of Optimization of Distribution Networks, RWTH Aachen University, Templergraben 64, D-52056 Aachen, Germany.*

Abstract

The unified modeling and solution framework, presented by Desaulniers *et al.* (1998), is applicable to nearly all types of vehicle-routing and crew-scheduling problems found in the literature thus far. The framework utilizes resource extension functions (REFs) as its main tool for handling complex side constraints that relate to a single vehicle route or crew schedule. The intention of this paper is to clarify which properties of REFs allow important algorithmic procedures, such as efficient representation of (partial) paths, efficient cost computations, and constant time feasibility checking for partial paths (=segments) and their concatenations. The theoretical results provided by the paper are useful for developing highly-efficient solution methods for both exact and heuristic approaches. Acceleration techniques for solving resource-constrained shortest-path subproblems are a key success factor for those exact algorithms which are based on column generation or Lagrangean relaxation. Similarly, those heuristic algorithms which are based on resource-constrained paths can benefit from efficient operations needed to construct or manipulate segments. Fast operations are indispensable for *efficient* local-search algorithms that explore edge-exchange or node-exchange neighborhoods. Efficiency is crucial, since these operations are repeatedly performed in many types of metaheuristics.

Key words: resource-constrained path, resource extension function, column generation, accelerated local search, vehicle routing and scheduling

1 Introduction

The unified modeling and solution framework, presented by Desaulniers *et al.* (1998), is applicable to nearly all types of vehicle-routing problems (VRPs) and scheduling problems for vehicles and crews (SPVCs) presented in the literature thus far. Resource extension functions (REFs) are its main tool for handling complex side constraints that relate to

* Corresponding author.
Email address: sirnich@or.rwth-aachen.de (Stefan Irnich).

Preprint available on www.dpor.rwth-aachen.de *23 February 2007*

a single vehicle route or crew schedule. Essentially, an REF is associated with an arc of the underlying network and describes the update of resources along that arc. Examples of resources are accumulated cost, time, load, distance, and, in general, attributes that describe the state of a vehicle when operating its tour or of a crew member when performing his or her schedule. If VRPs and SPVCs are solved by column generation or Lagrangean relaxation, the subproblem is a so-called shortest-path problem with resource constraints (SPPRC), see (Desrochers and Soumis, 1988; Irnich and Desaulniers, 2005). SPPRCs are extensions of classical shortest-path problems where costs are replaced by multi-dimensional resource vectors. These are updated via REFs along the path and constrained at intermediate nodes.

The intention of this paper is to clarify which properties of REFs allow important algorithmic procedures, such as efficient representation of (partial) paths, efficient cost computations, and constant time feasibility checking for partial paths (=segments) and their concatenations. In particular, for handling concatenations of segments, the generalization of REFs to segments and the inversion of REFs will turn out to be computationally advantageous. At least four areas of application exist for inverse REFs and the generalization of REFs to segments:

First, bidirectional shortest-path algorithms have been used primarily for unconstrained shortest-path algorithms and have proven to be highly effective for speeding up computations in the average case (Ahuja *et al.*, 1993; Helgason *et al.*, 1993). The concept of *inverse REFs* is mandatory for extending bidirectional shortest-path algorithms to SPPRCs with non-additive resource updates. Column-generation subproblems with non-additive resource updates arise in real-world routing and scheduling applications (Desaulniers *et al.*, 1998). For a huge collection of successfully solved real-world problems, we refer to the book by Desaulniers *et al.* (2005) and the references given there. Successful computational tests of bidirectional SPPRC algorithms used in branch-and-price methods for solving several types of VRPs with side constraints were conducted by Salani (2005). Without doubt, these new ideas were rigorous in solving some of the remaining open instances of the (Solomon, 1987) benchmark (see Jepsen *et al.*, 2006; Desaulniers *et al.*, 2006).

Second, Irnich (2007) shows that branch-and-price algorithms, where subproblems require the solution of (o, d)-shortest-path problems (from source o to sink d), can be accelerated by eliminating arcs from the network of the subproblem. The arc-elimination procedure proposed there is exact in the sense that reduced costs of o-d-paths allow the identification of those arcs that cannot be part of an optimal solution. Such arcs are removed from the subproblem network. An *efficient* implementation of the procedure combines forward and backward shortest-path labels, i.e., the solution of o-to-all and inverse d-to-all SPPRCs. The backward labels can only be computed if an inverse of each REF is defined.

Third, the *dynamic aggregation method* of Elhallaoui *et al.* (2005) is an acceleration technique for large-scale column-generation algorithms. As usual, the master program contains variables corresponding to paths (routes or schedules). One assumption of the method is that constraints are mainly of the set-partitioning type. Each set-partitioning constraint models the covering of a task $w \in W$ (a flight segment, a trip segment, a duty etc.), where W is the set of all tasks to be covered. The dynamic aggregation method replaces all tasks $w \in W$ with representatives $w_\ell, \ell \in L$ taken from a partitioning $W = \bigcup_{\ell \in L} W_\ell$. A sequence of aggregated restricted master programs (ARMP), where only the representative tasks are present in the LP, is then solved instead of the full restricted master programm (RMP). The speedup reported by Elhallaoui *et al.* (2005) results from the fact that an ARMP has fewer constraints than the original restricted master and, therefore, tends to be less degenerate. Note that degeneracy typically slows down the convergence of column-generation algorithms (*tailing-off* effect). In order to ensure the convergence towards the non-aggregated original optimal solution, a dynamic update procedures modifies the partitioning (major iterations). During minor iteration steps (when the partitioning is kept fixed), the subproblem is basically solved in order to generate *compatible* negative reduced cost paths. Compatible paths respect the current partitioning, i.e., paths covering each partition W_ℓ either fully or not at all. This is where the generalization of REFs to segments becomes important. Instead of solving the original subproblem over a large-scale network and filtering out compatible paths, one can replace every sequence of nodes corresponding to a partition W_ℓ by a single node. An efficient implementation makes it necessary to generalize REFs from arcs to segments, i.e., to replace a sequence of REFs by a single REF. We expect that this shrinking procedure will notably accelerate the solution process of the subproblem. However, the dynamic aggregation method, as proposed by Elhallaoui *et al.* (2005), needs solutions from the original non-aggregated network, for instance, to decide when to modify the partitioning, to provide an exact stopping rule, and to *disaggregate* dual variables of the ARMP. Therefore, the solution of aggregated subproblems should be treated similarly to the idea of partial pricing (Gamache *et al.*, 1999).

Fourth and finally, the last application uses REFs in a heuristic or metaheuristic context. Efficient primal heuristics for *rich VRPs* (Hasle *et al.*, 2003) can be computed using the modeling and solution framework by Irnich (2006). The framework is based on the giant-tour representation and resource-constrained paths. Nearly all metaheuristics for VRPs rely on the definition of neighbor solutions, e.g., k-Opt, k-Opt* as well as node and string exchanges. Standard local-search procedures iteratively build a neighbor solution first and check its feasibility afterwards. Typically, the feasibility check causes an extra effort bounded by $\mathcal{O}(n)$ for instances of size n. The new search techniques proposed by Irnich (2006) allow searching neighborhoods of size $\mathcal{O}(n^k)$ in $\mathcal{O}(n^k)$ time in the worst case. The fundamental idea is that a move decomposes the giant tour into a small (constant) number of segments, rearranges them, and concatenates the possibly inverted segments. The result is a neighboring giant tour. If a preprocessing procedure for computing segment REFs is performed a priori, the checking of the feasibility of the concatenated segments (=the new giant tour) can be accomplished in constant time. Furthermore, for several classes of VRPs, the *sequential search method* (Irnich *et al.*, 2006) is applicable and allows substantial speedups in the average case. Numerical results are reported in (Irnich, 2006; Hempsch and Irnich, 2007).

This paper is organized as follows: The next section gives basic definitions of resource-constrained paths and REFs for the classical case and extensions. Several examples and references show how real-world side constraints can be modeled using REFs. In Section 3, we analyze important properties of REFs, such as smoothness and linearity, and REFs for the classical and more general non-decreasing case. The computation of an aggregated REF for a segment is considered in Section 4. Section 5 discusses inverse REFs and their defining properties. Section 6 explains the most important findings with a small example. Final conclusions are given in Section 7.

2 Resource-Constrained Paths

Let $G = (V, A)$ be a digraph where V is the set of nodes and A the set of arcs. A *path* $P = (e_1, \dots, e_p)$ is a finite sequence of arcs (some arcs may occur more than once) where the head node of $e_i \in A$ is identical to the tail node of $e_{i+1} \in A$ for all $i \in \{1, \dots, p-1\}$. For the sake of convenience, we assume that G is simple, so that a path can be written as $P = (v_0, v_1, \dots, v_p)$ with the understanding that $e_i = (v_{i-1}, v_i) \in A$ holds for all $i \in \{1, \dots, p\}$.

2.1 Resource-Feasible Paths

Resource constraints can be modeled by means of *(minimal) resource consumptions* and *resource intervals* (e.g., the travel times t_{ij} and time windows $[a_i, b_i]$). Let R be the number of resources. A vector $T = (T^1, \dots, T^R)^\top \in \mathbb{R}^R$ is called a *resource vector* and its components *resource variables* (remark: $x^\top$ denotes the transposed vector to the vector x). T is said to be *not greater* than S if the inequality $T^i \leq S^i$ holds for all components $i \in \{1, \dots, R\}$. We denote this by $T \leq S$. For two resource vectors a and b, the interval $[a, b]$ is defined as the set $\{T \in \mathbb{R}^R : a \leq T \leq b\}$.

Resource intervals, also called *resource windows*, associated with a node $i \in V$ are denoted by $[a_i, b_i]$ with $a_i, b_i \in \mathbb{R}^R$, $a_i \leq b_i$. The changes in the resource consumptions associated with an arc $(i, j) \in A$ are given by a vector $f_{ij} = (f^r_{ij})^R_{r=1}$ of REFs. An REF $f^r_{ij} : \mathbb{R}^R \to \mathbb{R}$ depends on a resource vector $T_i \in \mathbb{R}^R$, which corresponds to the resource consumption accumulated along a path $(s, \dots, i)$ from s to i, i.e., up to the tail node i of arc (i, j). The result $f_{ij}(T_i) \in \mathbb{R}^R$ can be interpreted as a resource consumption accumulated along the path $(s, \dots, i, j)$.

Let P be any path in G. In order to simplify the notation, the nodes are numbered from 0 to p, i.e., $P = (0, 1, \dots, p-1, p)$. Path P is *resource-feasible* if resource vectors $T_i \in [a_i, b_i]$ exist for all $i \in \{0, 1, \dots, p\}$ such that $f_{i,i+1}(T_i) \leq T_{i+1}$ holds for all $i \in \{0, \dots, p-1\}$. For any resource consumption $T'_0 \in [a_0, b_0]$ at the start node 0, the set of all feasible resource vectors at the last node p is given by

$$\mathcal{T}(P, T'_0) = \{T_p \in [a_p, b_p] : \exists\, T_i \in [a_i, b_i] \text{ with } T_0 \geq T'_0 \text{ and } f_{i,i+1}(T_i) \leq T_{i+1} \text{ for all } i \in \{0, 1, \dots, p-1\}\}\,. \tag{1}$$

Additionally, we define $\mathcal{T}(P) = \mathcal{T}(P, a_0)$ and $\mathcal{F}(u, v)$ to be the set of all resource-feasible paths from a node u to a node v. Note that $P \in \mathcal{F}(u, v)$ holds if and only if $\mathcal{T}(P) \neq \varnothing$.

2.2 Cost and Pareto-Optimality

All the applications mentioned in the introduction consider paths within optimization problems. The easiest way of modeling cost is to use a resource $r \in \{1, \dots, R\}$ for this purpose. Two types of optimization problems are associated with REFs: First, a path $P = (0, 1, \dots, p)$ is given and the task is to find an 'optimal schedule' for this path, i.e., to solve $\min_{T \in \mathcal{T}(P)} T^{cost}$. The second type of problem is to find an 'optimal path' (the node sequence and the optimal schedule), i.e., $\min_{P \in \mathcal{F}(s,t)} (\min_{T \in \mathcal{T}(P)} T^{cost})$. It is beyond the scope of this paper to present and classify the different types of problems and associated solution approaches. Details and further references can be found in (Irnich and Desaulniers, 2005).

However, in order to analyze properties of REFs and their impact on different computational procedures, we must stress that there are good reasons for extending the above optimization problems from the one-dimensional 'cost' case to a multi-dimensional setting (see definition of the generic SPPRC in (Irnich and Desaulniers, 2005, p. 41)). If cost is a non-negative linear combination of several other resources (such as distance, time etc.), a cost-minimal path can be found among all Pareto-optimal paths w.r.t. the given resources. Several standard algorithms for solving the two problems 'optimal schedule' and 'optimal path' require the consideration of at least all Pareto-optimal points in $\mathcal{T}(P')$ for all prefix paths P' of P at intermediate nodes. In particular, this is the case if solution approaches, such as dynamic programming algorithms, are used and if all REFs are *non-decreasing* (see definition below).

2.3 Classical REFs

Classical SPPRCs consider REFs of the form

$$f_{ij}(T_i) = T_i + t_{ij} \tag{2}$$

(see Desaulniers *et al.*, 1998) or

$$f_{ij}(T_i) = \max\{a_j, T_i + t_{ij}\} \tag{3}$$

(see Irnich and Desaulniers, 2005), where a_j and $t_{ij} \in \mathbb{R}^R$ are constants associated with node j and arc (i,j), respectively. These *classical REFs* are separable by resources, i.e., no interdependencies exist between the different resources. Note that definitions (2) and (3) are equivalent w.r.t. the definition (1) of $\mathcal{T}(P, T_0)$, since $T_j \geq a_j$ is satisfied.

Note further that (2) and (3) are special cases of an REF where the lower bound depends on the arc (i,j), i.e.,

$$f_{ij}(T_i) = \max\{a_{ij}, T_i + t_{ij}\} \tag{4}$$

(with $a_{ij} = -\infty$ one gets (2) and with $a_{ij} = a_j$ one gets (3)). Defining classical REFs by (4) offers more flexibility for modeling. We will see that using (4) also provides more consistent results w.r.t. the generalization of REFs to segments. Additionally, when an arc (i,j) is used and $T_j \geq f_{ij}(T_i)$ is the resource consumption at node j, it is possible to check T_j against an arc-specific upper bound b_{ij} (instead of b_j). For the sake of simplicity, we do not use this extension in the paper. However, all results can be easily adapted to this case of arc-specific lower and upper bounds.

Well-known examples of real-world constraints that can be modeled with classical REFs are:

(1) **Globally constrained resources accumulated along nodes:** Here, all resource intervals are equal to $[a_i, b_i] = [0, U]$ with a global upper bound $U > 0$ and REFs are of the form $f_{ij}(T_i) = T_i + t_j$ for values $t_j \in \mathbb{R}_+$ and $j \in V$. Limited capacities U and demands t_j are the most prominent examples that occur with the classical VRP and its extensions.

(2) **Globally constrained resources accumulated along arcs:** If the resource consumption is on arcs, the only difference is that REFs are of the form $f_{ij}(T_i) = T_i + t_{ij}$. Examples are path-length constraints, where length is measured in distance, travel

time, fuel consumption, pay toll etc. Note that specialized models and algorithms exist for shortest-path problems that have solely this type of constraint (Beasley and Christofides, 1989; Borndörfer *et al.*, 2001).

(3) **Resources constrained by individual intervals accumulated along arcs and nodes:** The definition (1) of feasible paths is directly motivated by situations where valid service times are given by time windows $[a_i^{time}, b_i^{time}]$, and service and travel times by t_{ij}^{time}. It is possible to wait so that the start of a service may be later than the arrival at the node.

Another example is the cost resource. Costs are typically constrained only at the initial node 0 with $[a_0, b_0] = [0, 0]$, and $[a_i, b_i] = (-\infty, \infty)$ at all other nodes $i \in V, i \neq 0$. In the context of column generation, t_{ij}^{cost} is composed of arc costs c_{ij} and profits for all constraints containing arc (i, j). For instance, if node covering constraints are present, node profits λ_i (dual prices of the covering constraints) yield classical REFs with a reduced cost component defined by $t_{ij}^{cost} = c_{ij} - \lambda_i$.

The modeling capabilities of classical REFs also enable the computation of non-trivial attributes of paths. Several examples are given by Avella *et al.* (2004): They classify resources as *numerical and totalizable* (e.g., length, travel time), *numerical and non-totalizable* (e.g., road width, number of lanes), and *indexed* (e.g., type of road, gradient, parking restrictions). We assume that a path P is given and that the length of an arc (i, j) is l_{ij}. Avella *et al.* show how to formulate the following constraints as globally constrained resources that are accumulated along arcs:

- The average value of a totalizable parameter $p_{ij} \in \mathbb{R}$ over all arcs must not exceed U (or fall below L), i.e., $\sum_{(i,j)\in P} p_{ij} l_{ij} / \sum_{(i,j)\in P} l_{ij} \leq U$ ($\geq L$, respectively).

  For instance, if l_{ij} is the length in kilometers [km] and p_{ij} is the average travel time [h/km] on an arc (i, j), the upper bound U allows the bounding of the average travel time [h/km] along the path, e.g., guaranteeing a minimum speed of $1/U$ [km/h].

- The path has to contain arcs with certain properties f (possibly non-totalizable or indexed) that sum up to a length of at most U (or least L), $\sum_{(i,j)\in P: b_{ij}=1} l_{ij} \leq U$ ($\geq L$, respectively), where coefficient $b_{ij} \in \{0, 1\}$ determines whether $(i, j) \in A$ has property f or not.

  Here one can, for instance, bound the number of kilometers of one-lane roads.

- The path has to contain at least (most) x percent of arcs with a given property f, i.e., $\sum_{(i,j)\in P: b_{ij}=1} l_{ij} \geq \frac{x}{100} \sum_{(i,j)\in P} l_{ij}$ (with $\leq$ for 'at most').

  Relevant constraints of this type are a maximum of x% inner-city streets in a route.

Moreover, classical REFs are useful in the context of *multiple use of vehicles* (Taillard *et al.*, 1996). If a vehicle is used more than once in a planning period, it goes back to the depot for loading/unloading and possibly maintenance. The implication for some of the resources is that they have to be reset to their corresponding lower bound: For instance, a resource for the collected load is reset to zero. This fits in nicely with the above definitions $f_{ij}(T_i) = T_i + t_j$ and $f_{ij}(T_i) = T_i + t_{ij}$ of REFs. In order to reset resource r, one has to set the corresponding component of t_j or t_{ij} to $-\infty$ (any number not greater than

$a_j^r - b_i^r$). Other resources, such as time and cost, are updated in the standard way. Hence, restricting the length of the entire route or the arrival times is fully compatible with the reset of the first resources. The same technique is used in (Irnich, 2006; Hempsch and Irnich, 2007) for modeling a *giant tour* (Christofides and Eilon, 1969) as a single resource-constrained path.

It has been pointed out by Irnich and Desaulniers (2005) that definition (1) captures the case of *minimal* resource consumptions. If one wants to model *exact* resource consumptions instead, the inequalities in (1) have to be replaced by $T_{i+1}^r = f_{i,i+1}^r(T_i)$ for the particular resources r. For the time window case and the resource time, the equality means that waiting is not allowed: The arrival time at each node has to be identical to the start time of the service. Let $\mathcal{R}^=$ ($\mathcal{R}^\leq$) be the resources that force an equality (inequality) in (1). Gamache *et al.* (1998) note that a resource $r \in \mathcal{R}^=$ might equivalently be replaced by two resources $r_1, r_2 \in \mathcal{R}^\leq$. In this case, there are $R+1$ resources and REFs $\tilde{f}_{ij}$ are functions mapping from $\mathbb{R}^{R+1}$ to $\mathbb{R}^{R+1}$ (the $\tilde{}$ symbol refers to the case with the new resources r_1 and r_2). The resource intervals and REFs for r_1 are identical to those for r, while resource intervals for r_2 are $[a_i^{r_2}, b_i^{r_2}] = [-b_i^r, -a_i^r]$. For r_2, the REFs are defined by $\tilde{f}_{ij}^{r_2}(\tilde{T}_i) = -f_{ij}^r(\tilde{T}_i^1, \ldots, \tilde{T}_i^{r-1}, -\tilde{T}_i^{r_2}, \tilde{T}_i^{r+1}, \ldots, \tilde{T}_i^R)$. The new resource windows and REFs do *not* guarantee that $T^{r_1} = -T^{r_2}$ holds and that the resource variables fulfill (1) with equality. However, any path P that is resource-feasible w.r.t. the resources r_1 and r_2 is also feasible w.r.t. $r \in \mathcal{R}^=$, and vice versa. The resource variables T_p^r at the last node of the path can be feasibly chosen from $[-\tilde{T}_p^{r_2}, \tilde{T}_p^{r_1}]$. It is ensured that feasible values for the other resource variables T_i^r exist such that equality holds. The following example of *strict* time windows demonstrates the relationship between r, r_1, and r_2: Let $P = (0,1,2,3)$ be a path, $[0,10]$, $[11,12]$, $[8,24], [15,35]$ be the time windows at the four nodes, and let the travel times be 10 between all pairs of nodes. Trivially, the minimum resource consumptions of resource r_1 along the path P is $(0, 11, 21, 31)$. The resource r_2 uses the lower bounds intervals $[a_i^{r_2}, b_i^{r_2}]$ given by $[-10, 0]$, $[-12, -11]$, $[-24, -8]$, and $[-35, -15]$, and $\tilde{f}_{ij}(T_i)^{r_2} = \max\{a_i^{r_2}, T^{r_2} - 10\}$. The result is the feasible minimum resource consumptions -10, -12, -22, and -32. For the original resource $r \in \mathcal{R}^=$, this result means that any value between 31 and 32 is a feasible service time at node 3. In turn, feasible start times at node 0 are between 1 and 2.

Finally, the modeling of path-structural constraints, such as pairing and anti-pairing, precedence, follower and non-follower constraints as well as elementarity of the path by classical REFs, can be found in (Irnich and Desaulniers, 2005, Section 3).

2.4 General REFs

More general definitions of REFs with non-linear functions and interdependent resources provide a powerful instrument for modeling practically-relevant side constraints.

2.4.1 Load-Dependent Costs

In this paragraph, we consider REFs for routing problems where the cost of traveling along an arc (i, j) depends on the load transported over this arc. An example is the pickup-and-delivery problem presented in (Dumas *et al.*, 1991) where the cost of an arc (i, j) is given by a non-decreasing function $c_{ij} : \mathbb{R}_+ \to \mathbb{R}$ depending on the current load. For the sake of conciseness, we restrict ourselves to the two resources $R = \{cost, load\}$. Each customer node has a demand d_j with $d_j > 0$ for pickups and $d_j < 0$ for deliveries. Load is a restricted resource with resource intervals $[a_i^{load}, b_i^{load}] = [0, Q]$ for a given vehicle capacity Q, while cost is unrestricted. Formally, the REF for an arc (i, j) is given by

$$f_{ij}(T_i^{cost}, T_i^{load}) = (T_i^{cost} + c_{ij}(T_i^{load}), T_i^{load} + d_j). \tag{5}$$

The properties of f_{ij} mainly depend on the cost function c_{ij}. In Sections 4 and 5, we will analyze affine cost functions, polynomial cost functions and piecewise linear cost functions to see whether these can be generalized to segments or be inverted.

2.4.2 Soft Time Windows and Inconvenience Costs

Soft time windows model the fact that some service/visiting times within a given time window $[a_i^{time}, b_i^{time}]$ are more desirable than others. The inconvenience is expressed by a cost or penalty function $\pi_i : [a_i^{time}, b_i^{time}] \to \mathbb{R}_+$ which gives, for each feasible point t in time, the corresponding inconvenience cost. Soft time windows have been considered, e.g., by Sexton and Bodin (1985a,b); Ibaraki *et al.* (2005). Dumas *et al.* (1990) have shown that, for a given path $P = (0, 1, \ldots, p)$ and convex inconvenience cost functions π_i, the 'optimal schedule' problem, i.e.,

$$\begin{aligned} \min \; & \sum_{i=0}^{p} \pi_i(T_i) \\ \text{s.t.} \; & T_{i-1} + t_{i-1,i} \leq T_i \qquad \text{for all } i = 1, \ldots, p \\ & a_i \leq T_i \leq b_i \qquad \text{for all } i = 0, \ldots, p \end{aligned}$$

with $a_i, b_i, T_i \in \mathbb{R}$ for $i \in \{0, 1, \ldots, p\}$, can be solved by an algorithm that takes at most $\mathcal{O}(p)$ uni-dimensional minimizations over convex functions. Details, including the pseudo-code of the algorithm, can be found in (Dumas *et al.*, 1990).

Ibaraki *et al.* (2005) use arbitrary, possibly non-convex piecewise linear inconvenience cost functions π_i. These functions also cover the case of multiple time windows (see also Section 2.4.5), since high inconvenience costs model infeasible service-start times between consecutive time windows at the same location. Among other results, Ibaraki *et al.* (2005) provide a dynamic programming procedure for optimizing the overall cost. This procedure takes $\mathcal{O}(p\delta)$ time if p is the length of the route and δ the total number of pieces of the associated penalty functions.

In a more general setting with several resources, the resource $r = \mathit{cost}$ is updated depending on both resources $r = \mathit{cost}$ and $r = \mathit{time}$, i.e., the cost component of the REF is

$$f_{ij}^{cost}(T_i) = T_i^{cost} + c_{ij} + \pi_i(T_i^{time}), \tag{6}$$

where $c_{ij} \in \mathbb{R}$ is a fixed cost associated with the arc (i, j). The definition (6) adds the penalty when leaving a node and does, therefore, not add an inconvenience cost at the final node. In s-t-shortest-path problems, an alternative definition of the REFs for all arcs ending at the destination node t can solve this defect. Desaulniers *et al.* (1998) also suggest REFs of the form $f_{ij}(T_i, T_j)$ that depend on both resource variables, at node i and node j. Thus, REFs $f_{ij}^{cost}(T_i, T_j) = T_i^{cost} + c_{ij} + \pi_j(T_j^{time})$ would be possible. However, as noted in (Desaulniers *et al.*, 1998, p. 82), such a definition of an REF does, in general, impede the effective computation of intermediate resource consumptions. Computing $\mathcal{T}(P)$ can become (practically) untractable.

2.4.3 VRPSDP

The next example we present is the VRP with simultaneous deliveries and pickups (Min, 1989). Each customer i has to be visited once, and the servicing vehicle has to perform a delivery of demand d_i and a pickup of quantity p_i. An s-t-path $P = (0, 1, \ldots, p)$ is a feasible route if the maximum load on the vehicle does not exceed the vehicle capacity Q at any point in the route. In order to check the feasibility of a route w.r.t. capacity, at least two interdependent resources with non-linear REF are necessary (see Halse, 1992): A first resource $r = pick$ models the amount picked up when leaving a node, i.e.,

$$T_i^{pick} \in [p_i, Q] \quad \text{and} \quad f_{ij}^{pick}(T_i) = T_i^{pick} + p_j, \tag{7a}$$

while a second resource $r = mL$ models the maximum load occurring along the path $P = (s, \ldots, i, j)$, i.e.,

$$T_i^{mL} \in [\max\{p_i, d_i\}, Q] \quad \text{and} \quad f_{ij}^{mL}(T_i) = \max\{T_i^{pick} + p_j, T_i^{mL} + d_j\}. \tag{7b}$$

The REF f_{ij}^{mL} depends on both resources $r = pick$ and $r = mL$ in a non-linear way. The interpretation of f_{ij}^{mL} is the following: The maximum load on a path $P = (o, \ldots, i, j)$ can either occur at the very end when leaving node j, and is then given by the entire picked up load $T_i^{pick} + p_j$. Or the maximum load can emerge on the prefix path $P' = (o, \ldots, i)$. In this case, the maximum load on P comprises the maximum load T_i^{mL} on P' and the amount d_j delivered to node j. This explains the formula (7b).

Note that the VRPSDP generalizes several types of VRPs: The VRP with backhauls and mixed loads (VRPBM) has customers who either have a delivery or pickup demand, but not both, i.e., $d_j p_j = 0$ for all j. If all linehaul customers i ($d_i > 0$ and $p_i = 0$) have to precede the backhaul customers j ($d_j = 0$ and $p_j > 0$), the resulting problem is the VRP with backhauls (VRPB). All of these VRP types can be handled with the two resources *pick* and *mL*, and REFs of the form (7). However, the VRPB is much easier to model than the VRPSDP and VRPBM: It can also be modeled with two *independent* resources for the picked up and delivered quantities (or with only one resource that is reset to zero at the transition from linehaul to backhaul customers). A more detailed classification of VRPs with deliveries and pickups can be found in (Dethloff, 2002).

The case where some customers have positive delivery *and* pickup demand can create another interesting type of VRP: If it is an option to visit customers *once* or *twice* (but with the same vehicle), so-called *lasso tours* can occur. Some customers are first supplied only, then a round trip along customers with simultaneous delivery and pickup is performed, and finally pickups at the first customers are made (visited in the reverse order). The paper by Gribkovskaia *et al.* (2006) shows that such a mixed approach has the potential for notable cost savings. The savings result from better utilization of the vehicle capacity, since performing deliveries at the beginning yields additional space for the collection in the second combined delivery and pickup phase. These VRPs can be modeled as an extension of the VRPBM by splitting all customers j with $d_j p_j > 0$ into two nodes j^+ and j^- with $(d_{j^+}, p_{j^+}) = (d_j, 0)$ and $(d_{j^-}, p_{j^-}) = (0, p_j)$. In addition, pairing constraints have to ensure that j^+ and j^- are visited on the same tour.

2.4.4 Waiting Times and Times on Duty

Another important example of non-linear REFs is the modeling of (limited) waiting times and times on duty. Consider the case where the time of service is given by a resource

$r = time$ with given travel and service times t_{ij}^{time} and time windows $[a_i^{time}, b_i^{time}]$. Any feasible schedule $(T_i^{time})_{i=0}^{p}$ for a path $P = (0, 1, \ldots, p)$ imposes the following values: The time on duty is $d = d(P, (T_i^{time})_{i=1}^{p}) = T_p^{time} - T_0^{time}$, the (overall) waiting time is $w(P, (T_i^{time})_i) = T_p^{time} - T_0^{time} - \sum_{i=1}^{p} t_{i-1,i}^{time}$, and time spent on travel and service is $\sum_{i=1}^{p} t_{i-1,i}^{time}$. By bounding these durations by d^{max}, w^{max} and t^{max}, the determination of an optimal schedule with minimum time on duty and minimum waiting time becomes necessary. Additionally, when waiting is penalized by a constant (node independent) factor c^{wait}, cost-optimal schedules are non-trivial to determine. Desaulniers and Villeneuve (2000) have shown that the computation of cost-minimal (arc costs plus waiting costs) schedules can be performed by means of three resources, two of them having non-linear REFs. Here, we generalize their results in the sense that we model limited waiting times as well as limited times on duty.

The REF $f_{ij}(T_i)$ computes the minimum resource consumption along arc (i, j) with the following resources: (1) $r = time$ for the earliest start of service, (2) $r = wait$ the accumulated (minimum) waiting time, (3) an additional resource $r = hlp$ for computing $r = wait$, (4) the overall time on duty $r = duty$, and (5) a second additional resource $r = hlp'$ for computing $r = duty$. The overall time on duty clearly includes travel and service times, but waiting times might be included fully or partially. In order to cover the general case, we introduce the factor $\delta \in [0, 1]$ as the fraction of waiting times relevant for computing the time on duty, i.e., $T^{duty} = T^{travel} + T^{service} + \delta\, T^{wait}$. Using the results given in (Desaulniers and Villeneuve, 2000), the components of $f_{ij}(T_i)$ have to be defined as follows:

$$f_{ij}^{time}(T_i) = T_i^{time} + t_{ij}^{time} \tag{8a}$$
$$f_{ij}^{wait}(T_i) = \max\left\{T_i^{wait}, T_i^{hlp} - t_{ij}^{time} + a_j^{time}\right\} \tag{8b}$$
$$f_{ij}^{hlp}(T_i) = \max\left\{T_i^{wait} - b_j^{time}, T_i^{hlp} - t_{ij}^{time}\right\} \tag{8c}$$
$$f_{ij}^{duty}(T_i) = \max\left\{T_i^{duty} + t_{ij}^{time}, T_i^{hlp'} + (1-\delta)t_{ij}^{time} + \delta a_j^{time}\right\} \tag{8d}$$
$$f_{ij}^{hlp'}(T_i) = \max\left\{T_i^{duty} + (1-\delta)b_j^{time}, T_i^{hlp'} + (1-\delta)t_{ij}^{time}\right\} \tag{8e}$$

The feasible domains of these resource variables are given by the following intervals:

$$T_i^{time} \in [a_i^{time}, b_i^{time}] \quad \text{for all } i \in V \tag{9a}$$
$$T_i^{wait} \in [0, w^{max}] \quad \text{for all } i \in V \tag{9b}$$
$$T_i^{hlp} \in (-\infty, \infty) \quad \text{for all } i \in V \setminus \{0\} \quad \text{and} \quad T_0^{hlp} \in [-b_0^{time}, \infty) \tag{9c}$$
$$T_i^{duty} \in [0, d^{max}] \quad \text{for all } i \in V \tag{9d}$$
$$T_i^{hlp'} \in (-\infty, \infty) \quad \text{for all } i \in V \setminus \{0\} \quad \text{and} \quad T_0^{hlp'} \in [-\delta b_0^{time}, \infty) \tag{9e}$$

Note that resource $r = time$ is independent from the other resources and its update is classical. The two pairs $(wait, hlp)$ and $(duty, hlp')$ are pairwise interdependent resources coupled with a max-term.

As an example, we consider a path $P = (0, 1, 2, 3, 4)$ with time windows $[a_i^{time}, b_i^{time}]$ and travel times $t_{i,i+1}^{time}$ given in columns 2 to 4 of the following table. We further assume that 75% of the waiting time is relevant for computing the time on duty, i.e., $\delta = 0.75$.

Node	Time Window		Travel Time	Resource Variables T_i^r, $r =$				
i	a_i^{time}	b_i^{time}	$t_{i,i+1}^{time}$	$time$	$wait$	hlp	$duty$	hlp'
0	0	2	4	0	0	-2	0	-1.5
1	5	7	2	5	0	-6	4	-0.5
2	9	10	1	9	1	-8	6.75	0
3	11	11	3	11	2	-9	8.5	0.25
4	16	18		16	4	-12	13	1
Total			**10**		**4**		**10+0.75·4**	

First, the path $P_1 = (0,1)$ has no waiting time (travel time and time on duty coincide) because any start time $T_0^{time} \in [1,2] \subset [a_0^{time}, b_0^{time}]$ leads to no waiting at node 1. Second, for the path $P_2 = (0,1,2)$, starting at the latest possible time $T_0^{time} = 2$ results in an arrival at time $T_1^{time} + t_{12}^{time} = 6 + 2 = 8$ at node 2. The minimum waiting time for P_2 is, therefore, equal to 1. Third, path $P = (0,1,2,3,4)$ has a minimum waiting time of 4 because one has to wait at least 1 unit of time at node 2, 1 unit of time at node 3, and 2 units of time at the destination node. The overall time on duty is 13 time units, since traveling and service takes $10 = 4+2+1+3$ units of time and waiting contributes with 3 units (the minimum waiting time is 4 units of time and is considered partially by $3 = 0.75 \cdot 4$).

We will study this type of REF with a pairwise max-term in detail in Sections 4.2 and 5.2.

2.4.5 Multiple Time Windows

Multiple time windows are relevant if the service at a location j has to fall into one out of several different time slots, i.e., into the union of m_j disjoint intervals

$$I_j = [a_{j1}^{time}, b_{j1}^{time}] \cup [a_{j2}^{time}, b_{j2}^{time}] \cup \cdots \cup [a_{j,m_j}^{time}, b_{j,m_j}^{time}].$$

At least two substantially different ways of modeling multiple time windows exist: One possibility is representing the m_j time intervals by m_j different nodes. Instead of visiting location j, one has to visit one of these, i.e., all m_j nodes represent the same task. The second possibility is to have a single node i only, but to use a non-linear REF with a time component of the form

$$f_{ij}^{time}(T_i) = \begin{cases} a_{j1}^{time} & \text{if } T_i^{time} + t_{ij}^{time} < a_{j1}^{time} \\ a_{jk}^{time} & \text{if } T_i^{time} + t_{ij}^{time} \in (b_{j,k-1}^{time}, a_{jk}^{time}) \text{ for some } k > 1 \\ T_i^{time} + t_{ij}^{time} & \text{otherwise} \end{cases} . \tag{10}$$

The REF is non-smooth, but piecewise linear and non-decreasing. Note that it is *not* assured that $T_j^{time} \in I_j$ holds, since inequality (1) just requires $T_j^{time} \geq f_{ij}^{time}(T_i)$ (cf. Section 2.3). However, by using (10), the existence of feasible values $T_j^{time} \in I_j$ is guaranteed. Thus, feasibility w.r.t. multiple time windows is ensured. *Minimum* resource consumptions T_j^{time} fulfill $T_j^{time} \in I_j$.

2.4.6 Time-Dependent Travel Times

Several authors (e.g., Ahn and Shin, 1991; Malandraki and Daskin, 1992; Hill and Benton, 1992) have examined time-dependent travel times. For each arc (i,j) in the network, a function $t_{ij}^{time} : [a_i^{time}, b_i^{time}] \to \mathbb{R}_+$ provides the travel time $t_{ij}^{time}(T^{time})$ for traveling from i to j depending on the time of day T^{time}, i.e., the constant $t_{ij}^{time} \in \mathbb{R}_+$ in the classical case is replaced by a function. The time component of the REF becomes

$$f_{ij}^{time}(T_i) = T_i^{time} + t_{ij}^{time}(T_i^{time}). \tag{11}$$

A consistent definition of travel times $t_{ij}^{time}(T)$ requires that all functions have the so-called *non-overtaking property*

$$S_i^{time} \leq T_i^{time} \quad \Longrightarrow \quad f_{ij}^{time}(S_i^{time}) \leq f_{ij}^{time}(T_i^{time})$$

for all $S_i^{time}, T_i^{time} \in [a_i^{time}, b_i^{time}]$ (in (Ahn and Shin, 1991) the inequalities are stated as strict '$<$'-relations, which is not necessary). This is exactly the definition of a one-dimensional non-decreasing function (see below). If $t_{ij}^{time} : [a_i^{time}, b_i^{time}] \to \mathbb{R}$ is smooth and differentiable, the non-overtaking property is equivalent to that $t_{ij}^{time\prime}(T^{time}) \geq -1$ holds for the derivative and any $T^{time} \in [a_i^{time}, b_i^{time}]$.

2.4.7 Complex Cost Functions

Complex cost functions often occur in crew scheduling applications when modeling complex crew wages. Examples can be found in (Vance *et al.*, 1997; Gamache *et al.*, 1999; Desaulniers *et al.*, 1999). Their modeling by REFs is straightforward.

3 Resource Extension Functions and their Properties

For a fixed path $P = (0, 1, \ldots, p)$ and corresponding REFs $f_{i,i+1}$ for $i \in \{0, 1, \ldots, p-1\}$, the structure of the sets $\mathcal{T}(P)$ and $\mathcal{T}(P, T_0)$ with $T_0 \in \mathbb{R}^R$ can be complex. This has several important consequences. First, checking whether P is resource-feasible ($\mathcal{T}(P, T_0) \neq \varnothing$) or not, can be difficult. Second, as cost is modeled either as a separate resource or as a linear combination of resources, the structure of $\mathcal{T}(P, T_0)$ has implications on cost computations. Third, for algorithmic purposes, a 'simple' representation of $\mathcal{T}(P, T_0)$, e.g., with $\mathcal{O}(R)$ coefficients, is desirable. The next four subsection analyze REFs in terms of these aspects.

3.1 Structure of $\mathcal{T}(P, T_0)$

For any vector $x \in \mathbb{R}^R$, the set $x^{\llcorner}$ is defined as $x^{\llcorner} = \{y \in \mathbb{R}^R : y \geq x\}$. It is a cone with the unique extreme point x and the unit vectors of $\mathbb{R}^R$ as extreme rays. For a set $X \subseteq \mathbb{R}^R$, we define $f(X) = \{f(x) : x \in X\}$ and $X^{\llcorner} = \bigcup_{x \in X} x^{\llcorner}$.

In the following, let $P = (0, 1, \ldots, p)$ be an arbitrary path and $T_0 \in [a_0, b_0]$. For $p \geq 1$, we denote by $P^- = (0, 1, \ldots, p-1)$ the prefix path.

It follows directly from definition (1) that $T \in \mathcal{T}(P, T_0)$ implies $T^{\llcorner} \cap [a_p, b_p] \subset \mathcal{T}(P, T_0)$ holds. This has the following implication for the sets of feasible resource values:

Proposition 1 Let $P = (0, 1, \ldots, p-1, p)$ be a path with $p \geq 1$. The following relation holds for the sets $\mathcal{T}(P, T_0)$ and $\mathcal{T}(P^-, T_0)$:

$$\mathcal{T}(P, T_0) = f_{p-1,p}(\mathcal{T}(P^-, T_0))^{\llcorner} \cap [a_p, b_p] \quad \text{for all } T_0 \in [a_0, b_0] \tag{12}$$

Proofs for the above and all other propositions and theorems can be found in the appendix.

In the following, we will study smooth, linear and non-decreasing REFs and their impact on the sets $\mathcal{T}(P, T_0)$.

Proposition 2 Let $P = (0, 1, \dots, p)$ be a path with smooth REFs $f_{i,i+1}$ for all $i \in \{0, 1, \dots, p-1\}$. Then $\mathcal{T}(P, T_0)$ is compact for all $T_0 \in \mathbb{R}^R$.

The compactness of $\mathcal{T}(P, T_0)$ implies that the minima in the 'optimal schedule' and 'optimal path' problems exist (as long as P is feasible or a feasible path P exists). Otherwise, only an infimum surely exists ($\mathcal{T}(P, T_0) \subset [a_p, b_p]$ is bounded).

Next, we consider the important case where all REFs are non-decreasing functions. A function $f : \mathbb{R}^R \to \mathbb{R}^R$ is *non-decreasing*, if, for any pair $S, T \in \mathbb{R}^R$ with $S \leq T$, the inequality $f(S) \leq f(T)$ holds. Non-decreasing REF arise 'naturally' if an REF is interpreted as the update function for computing *minimum* resource consumptions. If the resource consumption S at node i is not greater (in any component) than T, we expect that a minimum resource consumption after traveling from i to j, i.e., $f_{ij}(S)$ and $f_{ij}(T)$, will also fulfill this relation.

Proposition 3 Let $P = (0, 1, \dots, p)$ be a path with non-decreasing REFs $f_{i,i+1}$ for $i = 0, 1, \dots, p-1$. Then $\mathcal{T}(P, T_0)$ is a possibly empty interval $I \subset \mathbb{R}^R$ for all $T_0 \in \mathbb{R}^R$. If $I \neq \varnothing$, the interval I is given by $I = [\hat{a}_p(T_0), b_p]$ and $\hat{a}_p(T_0) \in \mathbb{R}^R$ can be computed step-by-step using

$$\hat{a}_0(T_0) = \max\{a_0, T_0\} \quad \text{and} \quad \hat{a}_i(T_0) = \max\{a_i, f_{i-1,i}(\hat{a}_{i-1}(T_0))\} \tag{13}$$

for all $i \in \{1, \dots, p\}$.

Note that the second part of the above Proposition 3 requires that the path is feasible, i.e., $\mathcal{T}(P, T_0)$ is non-empty. Otherwise, $[\hat{a}_p(T_0), b_p] \neq \mathcal{T}(P, T_0) = \varnothing$. This can happen if $\hat{a}_i(T_0) \not\leq b_i$ holds for some intermediate node $i \in \{0, 1, \dots, p-1\}$. An example is the segment $P = (0, 1, 2)$ with $[a_0, b_0] = [0, 2]$, $[a_1, b_1] = [1, 1]$, $[a_2, b_2] = [4, 6]$, and $t_{01} = t_{12} = 2$. Formula (13) yields $[\hat{a}_2(T_0), b_2] = [4 + T_0, 6] \neq \varnothing$ but P is infeasible for any initial resource consumption $T_0 \in [0, 2]$.

Since classical REFs are non-decreasing, the results of Proposition 3 hold, i.e., the entire information about the structure of $\mathcal{T}(P, T_0)$ is given by the point $\hat{a}_p(T_0) \in \mathbb{R}^R$ and the upper bound b_p at node p. In order to simplify the notation, we assume *from now on* that all REFs f_{ij} already satisfy $f_{ij}(T_i) \geq a_j$ for all $T_i \in \mathbb{R}^R, (i, j) \in A$. For the classical case, this means that the REF f_{ij} includes the max-term with a_j, i.e., $f_{ij}(T_i) = \max\{a_j, T_i + t_{ij}\}$ (as, e.g., in definition (3)). A direct consequence is that the values $\hat{a}_i(T_0)$ can be represented directly as

$$\hat{a}_i(T_0) = f_{i-1,i} \circ f_{i-2,i-1} \circ \cdots \circ f_{12} \circ f_{01}(T_0) \tag{14}$$

for all $T_0 \geq a_0$ (note the common convention that $g \circ h(T)$ is defined as $g(h(T))$, i.e., the second function is applied first). For $i = 0$, the formula is consistent because the empty concatenation of functions is the identity, so that $a_0(T_0) = T_0$ holds. Note further that, for values $T_0 \not\geq a_0$, the max-term with a_0 is missing, so that Formula (13) can produce a different result.

Surprisingly, even if non-decreasingness seems natural, several examples of practically relevant 'optimal schedule' or 'optimal path' problems exist which have REFs with some decreasing component(s). The simplest case are linear REFs which may not be non-decreasing. A function $f : \mathbb{R}^R \to \mathbb{R}^R$ is *affine linear* if a matrix $P \in \mathbb{R}^{R \times R}$ and a vector $q \in \mathbb{R}^R$ exist, such that $f(x) = Px + q$ holds for all $x \in \mathbb{R}^R$. In the case of

$q = 0$, f is *linear*. It has been pointed out that linear node costs 'naturally' arise in column generation subproblems if resource variables appear with non-zero coefficients in the master problem formulation (see Desaulniers *et al.*, 1998). Examples of this type are synchronization of departure or arrival times in vehicle or airline scheduling (Ioachim *et al.*, 1994), combined inventory management and (ship) routing (Christiansen, 1996), and VRP with split delivery (Gendreau *et al.*, 2005).

Proposition 4 Let $P = (0, 1, \ldots, p)$ be a path with (affine) linear REFs $f_{i-1,i}$ for $i \in \{1, \ldots, p\}$. Then $\mathcal{T}(P, T_0)$ is the empty set or a polytope for all $T_0 \in \mathbb{R}^R$.

We end this paragraph with a small example of linear REFs for modeling time (i.e., start of service) and linear cost and inconvenience functions. Consider the path $P = (0, 1, 2)$, two resources $r = time$ and $r = cost$ with resource intervals $[a_0, b_0] = [0, 2] \times [0, 4]$, $[a_1, b_1] = [2, 5] \times [0, 5]$, and $[a_2, b_2] = [5, 10] \times [0, 15]$. Travel times are $t_{01} = 1$ and $t_{12} = 3$, costs are $c_{01} = 4$ and $c_{12} = 6$ and inconvenience cost factors are $\pi_0(T_0^{time}) = -2\,T_0^{time}$ and $\pi_1(T_1^{time}) = -T_1^{time}$, see Formula (6). The REFs are, therefore, defined by $f_{01}(T_0^{time}, T_0^{cost}) = (\max\{2, T_0^{time} + 1\}, T_0^{cost} + 4 - 2\,T_0^{time})$ and $f_{12}(T_1^{time}, T_1^{cost}) = (\max\{5, T_1^{time} + 3\}, T_1^{cost} + 6 - T_1^{time})$ with a partially decreasing second component. Figure 1 shows the two-dimensional polytopes $\mathcal{T}(0, T_0), \mathcal{T}((0, 1), T_0)$, and $\mathcal{T}(P, T_0)$ for $T_0 = a_0 = (0, 0)^\top$. Note that for different values of T_0 the polytope $\mathcal{T}(P, T_0)$ can change w.r.t. the number of faces and extreme points.

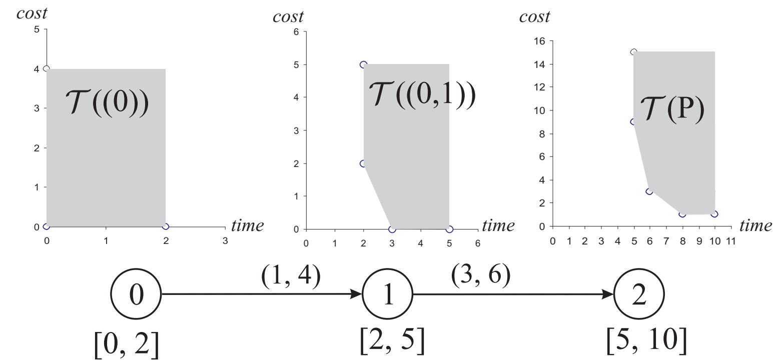

Fig. 1. Example of a Linear REF with a Decreasing Component

3.2 Efficient Representation of $\mathcal{T}(P, T_0)$

Can we efficiently represent $\mathcal{T}(P, T_0)$ in a parametrized form, depending on $T_0 \in \mathbb{R}^R$ (or $\in [a_0, b_0]$)? Obviously, the preceding subsection has given a partial answer to this question.

Provided that all REFs $f_{i-1,i}$ are non-decreasing, the set $\mathcal{T}(P, T_0)$ is either empty or given by $[\hat{a}_p(T_0), b_p]$ with $\hat{a}_p(T_0) = f_{p-1,p} \circ f_{p-2,p-1} \circ \cdots \circ f_{12} \circ f_{01}(T_0)$. If one wants to compute $\hat{a}_p(T_0)$ with an effort independent of the length of the path, one has to find an efficient representation of the composition $f_{p-1,p} \circ f_{p-2,p-1} \circ \cdots \circ f_{12} \circ f_{01}$. This is simply the generalization of p individual REFs for each arc to a single REF for the segment $P = (0, 1, \ldots, p)$. Therefore, an efficient representation of $\mathcal{T}(P, T_0)$ can be gained from any efficient representation of $f_P = f_{p-1,p} \circ f_{p-2,p-1} \circ \cdots \circ f_{12} \circ f_{01}$.

Two cases have to be answered negatively: If an REF has at least one decreasing component, we cannot expect the representation of $\mathcal{T}(P, T_0)$ to be independent of the length p of the path. An example is linear waiting times (Ioachim *et al.*, 1998; Irnich and Desaulniers, 2005) where the polytope $\mathcal{T}(P, T_0)$ can have $\mathcal{O}(p) = \Omega(p)$ extreme points. For more general settings with $R > 2$ resources, the number of extreme points can grow even more rapidly. The second case is that all REFs $f_{i-1,i}$ are non-decreasing but defined differently on $m_i \geq 2$ intervals. Examples are REFs for multiple time windows (see Formula (10)) or soft time windows with $m_i \geq 2$ linear pieces. Here, the number of linear pieces necessary to define f_P can grow in the order of $\Omega(\sum_{i=0}^{p} m_i)$.

3.3 Feasibility Problem

The feasibility problem for a given path $P = (0, 1, \ldots, p)$ and $T_0 \in [a_0, b_0]$ is to answer the question of whether $\mathcal{T}(P, T_0) \neq \varnothing$ holds or not. As shown in (Irnich and Desaulniers, 2005), the feasibility problem can be $\mathcal{NP}$-hard if no additional assumptions about the REFs are given.

Even for the 'most desirable' case where all REFs are non-decreasing, the feasibility problem cannot be solved by considering f_P alone. If $f_P(T_0) \not\leq b_p$, the path P is clearly infeasible. However, $f_P(T_0) \leq b_p$ provides no information about whether P is feasible or not. For the moment, we can just state the following result:

Proposition 5 Let $P = (0, 1, \ldots, p)$ be a path with non-decreasing REFs $f_{i-1,i}$ for $i \in \{1, \ldots, p\}$. Then P is feasible, i.e., $\mathcal{T}(P, T_0) \neq \varnothing$, if and only if $\hat{a}_i(T_0) \leq b_i$ holds for all $i \in \{0, 1, \ldots, p\}$.

Proposition 5 means that we have to perform $p + 1$ comparisons and apply p REF evaluations (on vectors having R components) in order to check whether path P is feasible. We will see in Section 5 that inversion of REFs allows the reduction of the computational effort to a single comparison. However, additional assumptions on the REFs are necessary for ensuring that these can be inverted.

3.4 Cost Computation

In the simplest case, cost is one of the resources, e.g., represented by $r = 1$. Otherwise, cost is modeled implicitly as a non-negative linear combination of two or more of the resources $\{1, \ldots, R\}$. Therefore, the (minimum) cost of a given path $P = (0, 1, \ldots, p)$ with initial resource consumption $T_0 \in [a_0, b_0]$ is $\inf_{T \in \mathcal{T}(P,T_0)} \alpha^\top T$ for some $\alpha \in \mathbb{R}^R_+$. Note that, for non-smooth REFs, the set $\mathcal{T}(P)$ may not be compact so that a minimum may not exist.

The analysis of the structure of $\mathcal{T}(P, T_0)$ in the preceding section directly implies the following two results: If all REFs are non-decreasing, the minimum cost of the path $P = (0, 1, \ldots, p)$ is given by $\alpha^\top \hat{a}_p(T_0)$. If all REFs are linear (but with some decreasing components), the determination of the minimum cost requires the explicit or implicit solution of an LP, since $\mathcal{T}(P, T_0)$ is a polytope. Implicit methods for the description of this polytope have been developed for specialized cases with two resources by Ioachim *et al.* (1998) and three resources by Gendreau *et al.* (2005). These methods also lead to effective dominance rules when the optimal path problem is solved by a dynamic programming (labeling) procedure.

4 Generalization of REFs to Segments

In the following, the term *segment* refers to an arbitrary path that is treated as an indecomposable unit in a given digraph G. The distinction between segments and paths is made for the sake of explanation only: A given path P, e.g., in an optimal schedule problem, can be decomposed into several segments $P = P_1 + P_2 + \cdots + P_\ell$. The modification of a giant tour by local-search moves is another example of segments arising. In an optimal path problem, segments occur if some sequences of nodes or arcs are forced to be visited in a given ordering. Examples are s-t-shortest-path problems in branch-and-price approaches when branching rules fix flows along some arcs, and the solution of the partial pricing problem in the dynamic aggregation procedure (see the introduction).

For this section, let P, P_1 and P_2 be segments in G. We denote by $\alpha(P)$ the first node and by $\omega(P)$ the last node of the segment P. In order to apply Proposition 3 and Formula (14) with a small computational effort, the focus is on analyzing the following properties of a *segment REF* $f_P = f_{p-1,p} \circ f_{p-2,p-1} \circ \cdots \circ f_{12} \circ f_{01}$.

(SCoF): f_P is in the **s**ame **c**lass **o**f **f**unctions as all the arc REFs f_{ij} are.

(FNoC): For all paths P, the segment REFs f_P can be represented with a **f**ixed **n**umber **o**f **c**oefficients (independent of the length of P) so that a function evaluation $f_P(T)$ can be accomplished in $\mathcal{O}(1)$ time and space.

(CJS): For any two segments P_1, P_2 with $\omega(P_1) = \alpha(P_2)$, the **c**omputation of the segment REF $f_{P_1+P_2}$ for the **j**oined **s**egment takes $\mathcal{O}(1)$ time and space.

4.1 Classical REFs

The following propositions show that the classical case can easily be generalized to segments, since REFs for segments and their concatenations are of the same form as REFs for arcs. The coefficients defining the REF for a segment can be computed from the coefficients of the parts (arcs or partial segments) the new segment is constructed of.

Proposition 6 Let f_1 and f_2 be given by $f_1(T) = \max\{a_1, T + t_1\}$ and $f_2(T) = \max\{a_2, T + t_2\}$. Then

$$f_2 \circ f_1(T) = \max\{a, T + t\},$$

with $a = \max\{a_2, a_1 + t_2\} \in \mathbb{R}^R$ and $t = t_1 + t_2 \in \mathbb{R}^R$.

Proposition 7 Let $P = (0, 1, \ldots, p)$ be a segment with classical REFs for all arcs, i.e., $f_{i,i+1}(T) = \max\{a_{i+1}, T + t_{i,i+1}\}$ for all $i \in \{0, 1, \ldots, p-1\}$. The segment REF is

$$f_P(T) = f_{p-1,p} \circ \cdots \circ f_{12} \circ f_{01}(T) = \max\{a_P, T + t_P\} \tag{15a}$$

with

$$a_P = f_{p-1,p} \circ \cdots \circ f_{12} \circ f_{01}(a_0) \quad \text{and} \quad t_P = \sum_{i=0}^{p-1} t_{i,i+1}. \tag{15b}$$

Next, we consider the concatenation of two segments. For two segments $P_1 = (v_0, v_1, \ldots, v_p)$ and $P_2 = (w_0, w_1, \ldots, w_q)$ with $v_p = \omega(P_1) = \alpha(P_2) = w_0$, the segment $P_1 \oplus P_2$ is defined as $(v_0, v_1, \ldots, v_p, w_1, \ldots, w_p)$. Otherwise, if $(v_p, w_0) = (\omega(P_1), \alpha(P_2)) \in A$ then $P_1 + P_2$ denotes the segment $(v_0, v_1, \ldots, v_p, w_0, w_1, \ldots, w_p)$.

Proposition 8 Let P_1, P_2 be segments in G with REFs $f_{P_1}(T) = \max\{a_{P_1}, T + t_{P_1}\}$ and $f_{P_2}(T) = \max\{a_{P_2}, T + t_{P_2}\}$, respectively.

(a) If $\omega(P_1) = \alpha(P_2)$ then $f_{P_1 \oplus P_2}(T) = f_{P_2} \circ f_{P_1}(T) = \max\{a_{P_1 \oplus P_2}, T + t_{P_1 \oplus P_2}\}$, with $a_{P_1 \oplus P_2} = \max\{a_{P_2}, a_{P_1} + t_{P_2}\} \in \mathbb{R}^R$ and $t_{P_1 \oplus P_2} = t_{P_1} + t_{P_2} \in \mathbb{R}^R$.

(b) If $(i, j) := (\omega(P_1), \alpha(P_2)) \in A$ then $f_{P_1+P_2}(T) = f_{P_2} \circ f_{ij} \circ f_{P_1}(T) = \max\{a_{P_1+P_2}, T + t_{P_1+P_2}\}$, with $a_{P_1+P_2} = \max\{a_{P_2}, a_{\alpha(P_2)} + t_{P_2}, a_{P_1} + t_{ij} + t_{P_2}\} \in \mathbb{R}^R$ and $t_{P_1+P_2} = t_{P_1} + t_{ij} + t_{P_2} \in \mathbb{R}^R$.

There is a subtle difference between Formula (2) concerning REFs for arcs and the result of Proposition 8(b) when these are applied to the segments $P_1 = (i)$ and $P_2 = (j)$. The proposition yields $f_P(T) = \max\{\max\{a_j, a_i + t_{ij}\}, T + t_{ij}\}$. This differs from the REF $f_{ij}(T) = \max\{a_j, T + t_{ij}\}$ defined for the arc (i, j). Note that both functions are identical on the interval $[a_i, \infty)$ but may differ on $(-\infty, a_i)$. With $a_{ij} = \max\{a_j, a_i + t_{ij}\}$ the *better* or *more consistent* definition of a classical REF is $f_{ij}(T) = \max\{a_{ij}, T + t_{ij}\}$, see also Formula (4) and the comments there. Anyway, in all cases, f_P is of the required form (15) as stated in Proposition 8.

Summarizing the results, we can state the following theorem:

Theorem 1 Classical REFs $f_{ij}(T) = \max\{a_j, T + t_{ij}\}$ for all $(i, j) \in A$ can be generalized to segments. The segment REFs have the properties (SCoF) and (FNoC), and their coefficients can be computed using Formulas (15). Concatenations of segments with segment REFs of the form $f_P(T) = \max\{a_P, T + t_P\}$ have the property (CJS) and the coefficients of the concatenated segments can be computed by the formulas given in Proposition 8.

4.2 REFs with a Pairwise Max-Term

In Sections 2.4.3 and 2.4.4, we have seen that REFs, where pairs of resource variables are coupled together with a max-term, are useful for modeling simultaneous deliveries and pickups as well as for computing minimum waiting times, minimum times on duty, and travel and waiting costs accumulated along the stops of a vehicle path. For this paragraph, let the number of resources $R = 2U$ be even. All resource vectors are composed of pairs (T, T'), where T and T' are U-dimensional vectors. In the same way, REFs are written as $f : \mathbb{R}^U \times \mathbb{R}^U \to \mathbb{R}^U \times \mathbb{R}^U, (T, T') \mapsto (g(T, T'), h(T, T'))$ with specific functions $g, h : \mathbb{R}^U \times \mathbb{R}^U \to \mathbb{R}^U$. The following proposition shows that REFs with a pairwise max-term have the same 'nice' properties as classical REFs.

Proposition 9 Let f_1 and f_2 be given by $f_1(T, T') = (\max\{a_1, T + t_1, T' + u_1\}, \max\{a'_1, T + t'_1, T' + u'_1\})$ and $f_2(T, T') = (\max\{a_2, T + t_2, T' + u_2\}, \max\{a'_2, T + t'_2, T' + u'_2\})$. Then

$$f_2 \circ f_1(T, T') = (\max\{a, T + t, T' + u\}, \max\{a', T + t', T' + u'\}) \tag{16}$$

with $a = \max\{a_2, a_1 + t_2, a'_1 + u_2\}$, $t = \max\{t_1 + t_2, t'_1 + u_2\}$, $u = \max\{u_1 + t_2, u'_1 + u_2\}$, $a' = \max\{a'_2, a_1 + t'_2, a'_1 + u'_2\}$, $t' = \max\{t_1 + t'_2, t'_1 + u'_2\}$, $u' = \max\{u_1 + t'_2, u'_1 + u'_2\}$.

The following theorem is a direct consequence of the preceding proposition.

Theorem 2 REFs of the form $f_{ij}(T, T') = (\max\{a_j, T + t_{ij}, T' + u_{ij}\}, \max\{a'_j, T + t'_{ij}, T' + u'_{ij}\})$ for all arcs $(i, j) \in A$ can be generalized to segments. A segment $P = (0, 1, \ldots, p)$ has an REF of the form

$$f_P(T,T') = f_{p-1,p} \circ \cdots \circ f_{12} \circ f_{01}(T,T')$$
$$= (\max\{a, T+t, T'+u\}, \max\{a', T+t', T'+u'\}) \tag{17}$$

with $(a,a') = f_{p-1,p} \circ \cdots \circ f_{12} \circ f_{01}(a_0, a'_0)$, i.e., properties (SCoF) and (FNoC). Concatenations of segments with segment REFs of the form $f_P(T,T') = (\max\{a, T+t, T'+u\}, \max\{a', T+t', T'+u'\})$ have the property (CJS) and the coefficients of the concatenated segments can be computed by the formulas given in Proposition 9.

VRPSDP The VRP with simultaneous deliveries and pickups has REFs with pairwise max-terms, where $a_i = p_i$, $a'_i = \max\{p_i, d_i\}$, $b_i = b'_i = Q$, $t_{ij} = t'_{ij} = p_j$, $u_{ij} = -\infty$, and $u'_{ij} = d_j$ holds for all nodes $i \in V$. If we assume $p_i, d_i \geq 0$ for all $i \in V$ as well as $T \geq p_i$ and $T' \geq \max\{p_i, d_i\}$ the values a_i, a'_i can be set to 0. This simplifies Formula (17) in the following way: The segment REF for $P = (0,1,\ldots,p)$, $T \geq p_i$, and $T' \geq \max\{p_i, d_i\}$ is

$$f_P(T,T') = \left(T + \sum_{i=1}^{p} p_i, \quad \max\{T + mL^P, T' + \sum_{i=1}^{p} d_i\} \right), \tag{18}$$

where mL^P is a constant representing the maximum load that occurs on segment P. To be more precise, T is the load picked up *after* leaving the first node 0 (i.e., T has to include the pickup quantity p_0 at node 0) and T' is the maximum load on the vehicle that occurs before *arriving at* node 1. Also, mL^P does not consider what happens before node 1, i.e.,

$$mL^P = \max \left\{ \sum_{i=1}^{k} p_i + \sum_{j=k+1}^{p} d_j : k \in \{1,\ldots,p\} \right\}.$$

4.3 Non-decreasing REFs

Besides classical REFs and REFs with a pairwise max-term, we have seen three other types of non-decreasing REFs in Section 2.4: REFs for modeling load-dependent costs, multiple time windows, and time-dependent travel times.

4.3.1 Load-Dependent Costs

Given $P = (0,1,\ldots,p)$ and REFs $f_{i-1,i}$ of the form (5) for $i \in \{1,\ldots,p\}$, it is straightforward to see that

$$f_P(T_0) = f_{p-1,p} \circ \cdots \circ f_{12} \circ f_{01}(T_0)$$
$$= \left(T_0^{cost} + \sum_{i=1}^{p} c_{i-1,i} \left(T_0^{load} + \sum_{j=0}^{i-1} d_j \right), T_0^{load} + \sum_{i=1}^{p} d_i \right)$$

holds. If we want f_P to be a function of the same form as the REF $f_{i-1,i}$ (property (SCoF)), we have to look for sets $\mathcal{C}$ of functions $c \in \mathcal{C}$, $c : \mathbb{R} \to \mathbb{R}$ which are closed under (1) addition of functions, and (2) the shift operation. For any $d \in \mathbb{R}$, a corresponding shift operation maps a function $c \in \mathcal{C}$ to the function defined by $T \mapsto c(T+d)$. In the case where (1) and (2) hold, the above term $T_0^{cost} + \sum_{i=1}^{p} c_{i-1,i} \left(T_0^{load} + \sum_{j=0}^{i-1} d_j \right)$ can be replaced by $T_0^{cost} + c_P(T_0^{load})$.

First, the set $\mathcal{H} = \mathcal{P}^m$ of polynomials of a degree not greater than m is closed under the addition and shift operations. It follows that the cost functions $c_{ij} \in \mathcal{P}^m$ for the load-dependent cost on arcs $(i,j) \in A$ imply REFs for a segment which have a cost function c_P defined by a polynomial in $\mathcal{P}^m$ (property (FNoC)). Since the computation of the corresponding polynomial takes $\mathcal{O}(m^2) = \mathcal{O}(1)$ time and space, property (CJS) is fulfilled. As a special case for $m = 1$ affine linear costs are possible (see Fig. 2(a)). Note that non-decreasingness of the (segment) REFs is satisfied if all cost functions are non-decreasing.

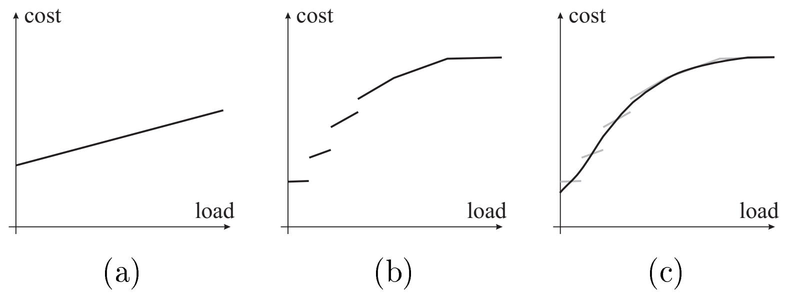

Fig. 2. Load-dependent Costs. (a) Affine Linear, (b) Piecewise Linear with Multiple Pieces, (c) Approximation by a Polynomial

Second, if the c_{ij} are defined piecewise using up to m_{ij} linear functions (as depicted in Fig. 2(b)), the cost function for the segment $P = (0, 1, \dots, p)$ is also piecewise linear (property (SCoF)). However, the number of proper pieces necessary for defining c_P can become $m_{01} \cdot m_{12} \cdot \ldots \cdot m_{p-1,p}$ ((FNoC) does not hold). Consequently, if one is interested in compact representations, one should not use functions c_{ij} defined differently on different intervals. Instead, piecewise defined tariffs should be approximated, e.g., by polynomials leading to segment REFs with a fixed number of coefficients (independent of the length of the segment under consideration). Figure 2(c) shows an approximation of the cost-function depicted in Figure 2(b) by a polynomial.

4.3.2 Multiple Time Windows

For multiple time windows, it is easy to see that the REF for a segment is of the same form as the REF for an arc, see Formula (10). Concerning the effort of representing the REF, Gietz (1994, p. 65) has shown that one has to consider up to $m + (m-1)p$ intervals if up to m time windows are given at each node of a path of length p. Again, (SCoF) holds but (FNoC) does not. Anyway, in real-world problems, Gietz (1994, p. 65 f) could not find instances in which more than nine intervals had to be considered. Thus, for long routes, the worst-case upper bound $m + (m-1)p$ was never reached.

4.3.3 Time-Dependent Travel Times

Even if the REF is non-decreasing (has the non-overtaking property), the generalization to segments is hardly possible. For the sake of simplicity, we assume that all time windows are identical, i.e., $[a_i^{time}, b_i^{time}] = [0, \tau]$, and that all travel time functions $t_{ij}^{time}(T)$ are non-negative. The travel time on a segment $P = (0, 1, \dots, p)$ is

$$
\begin{aligned}
t_P(T_0) = T_0 \quad & + t_{01}(T_0) \\
& + t_{12}(T_0 + t_{01}(T_0)) \\
& + t_{23}(T_0 + t_{12}(T_0 + t_{01}(T_0))) \\
& + t_{34}(T_0 + t_{23}(T_0 + t_{12}(T_0 + t_{01}(T_0)))) \\
& + \ldots
\end{aligned}
$$

(we have omitted the superscript *time*). For the segment P to have a travel time function t_P that is of the same form as the individual travel time functions on the arcs (property (SCoF)), it is required that this class of functions $\mathcal{H}$ be closed under (1) addition of functions, (2) addition of a constant, and (3) concatenation of functions. The set of all polynomials $\mathcal{H} = \mathcal{P}$ satisfies (1)-(3) but the obvious drawback is that t_P has increasing degree (and, therefore, an increasing number of coefficients) when P gets longer. For affine linear functions, i.e., $\mathcal{H} = \{f(T) = cT + d : d \in \mathbb{R}_+, c \in \mathbb{R}, c \geq -d/\tau\}$, the class $\mathcal{H}$ also fulfills (1)-(3). However, these functions are not useful for modeling practically relevant aspects, such as peaks with increased travel time during rush-hours etc. Defining piecewise linear functions does not solve the problem, since one ends up with numerous linear pieces defined on disjoint intervals (such as for load-dependent costs and weight breaks).

We are not aware of any class $\mathcal{H}$ of functions that fulfills property (SCoF), can model practically relevant aspects, and has a compact representation, i.e., property (FNoC).

4.4 Soft Time Windows, Inconvenience Costs, and Linear Node Costs

In the following, we will discuss different cases for the inconvenience functions $\pi_i : [a_i^{time}, b_i^{time}] \to \mathbb{R}_+$. First, if all π_i are non-decreasing, the segment REF f_P is also non-decreasing. Even for the simplest case of linear non-decreasing inconvenience functions $\pi_i(T^{time}) = w_i \cdot T^{time}$ with $w_i \geq 0$ for all $i \in V$, it is difficult to find a compact representation. The reason for this difficulty is that we end up with a piecewise defined REF, i.e.,

$$
f_P(T_0^{cost}, T_0^{time}) = \left(T_0^{cost} + \sum_{i=1}^{p} c_{i-1,i} + \pi_P(T_0), \max\{a_P^{time}, T_0 + \sum_{i=1}^{p} t_{i-1,i}\} \right)
$$

with π_P a piecewise linear inconvenience function. A simple example is $P = (0,1)$, $[a_0^{time}, b_0^{time}] = [0,2]$, $[a_1^{time}, b_1^{time}] = [2,4]$, $t_{01} = 1$. Here, $\pi_P : [0,2] \to \mathbb{R}$ with $\pi_P(T_0) = w_0 T_0 + 2w_1$ for $T_0 \in [0,1]$ and $\pi_P(T_0) = (w_0 + w_1)T_0 + (w_0 + 2w_1)$ for $T_0 \in [1,2]$. It is easy to see that the number of pieces can become $p+1$ for a segment $P = (0,1,\ldots,p)$ with $p+1$ nodes ((FNoC) does not hold). Different non-decreasing functions, such as polynomials, imply that the same discrete cases must be distinguished.

Second, we assume that all inconvenience functions are linear but decreasing, i.e., $\pi_i(T^{time}) = w_i \cdot T^{time}$ with $w_i \leq 0$ for all $i \in V$. These inconvenience functions have been used by Sexton and Bodin (1985a,b). As pointed out previously, $\mathcal{T}(P, T_0)$ is a polytope but not necessarily a (multi-dimensional) interval. Consequently, we cannot use Proposition 3, Formula (14) and the segment REF for representing $\mathcal{T}(P, T_0)$. However, the computation of *cost-minimal* schedules can be accomplished by replacing the REF (6) by a non-decreasing REF, but at the cost of 'inverting' the underlying digraph. In order to see this, note first that the cost-minimal schedule $(T_i)_{i=1}^p$ for $P = (0,1,\ldots,p)$ and $T_0 \in [a_0, b_0]$

visits each node $i \in \{1, \ldots, p\}$ as late as possible. The minimum inconvenience cost of the segment w.r.t. T_0^{time} is given by $\pi_P(T_0) = \min\{w_0 T_0 + \rho_P, \sigma_P\}$, where $\rho_P, \sigma_P \in \mathbb{R}$ are constants depending on the segment P and $w_0 \leq 0$ is the slope of the inconvenience function at the start node 0 of the segment. Here, it is assumed that one arrives as late as possible at the final node p, i.e., $T_p^{time} = b_p^{time}$. Otherwise, for any feasible $T_p^{time} \in [a_p, b_p]$, the minimum cost depends on T_p so that $\pi_P(T_0) = \min\{w_0 T_0 + \rho_P(T_p), \sigma_P(T_p)\}$. The functions $\rho_P(T_p)$ and $\sigma_P(T_p)$ are *not* non-decreasing and depend on resource variables on the final node of the segment. Consequently, we cannot give 'simple' update formulas for inconvenience cost functions when two or more segments are concatenated ((CJS) does not hold). We suggest using the following 'inverse approach' in order to handle linear inconvenience functions with negative slope. Instead of digraph (V, A), we use a new digraph with the same set V of nodes and reverse arcs (j, i) for all $(i, j) \in A$. The resource $r = time$ is replaced by a resource $r = ntime$. This new resource models 'negative points in time' and is constrained by $[a_i^{ntime}, b_i^{ntime}] = [-b_i^{time}, -a_i^{time}]$. The resource update is

$$\begin{aligned} &g_{ji}(T_j^{cost}, T_j^{time}) \\ &= \left(T_j^{cost} + c_{ij} - w_j T_j^{ntime}, \quad \max\{T_j^{ntime} + t_{ij}^{time}, -b_i^{time}\}\right) \end{aligned}$$

It is easy to prove that any schedule $(T_i)_{i=0}^p$ for $P = (0, 1, \ldots, p)$ and the original REFs is resource-feasible if and only if the schedule $(T_i^{cost}, -T_i^{time})_{i=p}^0$ is feasible for the segment $(p, p-1, \ldots, 0)$ with resources *cost* and *ntime* and REFs g_{ji}. The advantage of new REFs is that they are non-decreasing in both components (because of $-w_j \geq 0$). They can be handled like the case of linear non-decreasing inconvenience costs (therefore, we have property (SCoF) for piecewise linear costs, and not (FNoC)).

Third, linear inconvenience cost functions with positive as well as negative slopes at different nodes exceed the complexity of the two cases considered before. These problems need a special algorithmic treatment. Ioachim *et al.* (1998) provide solution approaches for optimal path problems with only two resources (shortest-path problems with time windows and linear node costs) based on the piecewise representation of the lower envelope describing the two-dimensional polytope $\mathcal{T}(P)$. Similarly, the above-mentioned study by Ibaraki *et al.* (2005) considers exactly this case, but the focus is not on analyzing REFs for segments but on procedures for accelerating local search for standard VRP neighborhoods.

Fourth and finally, convex inconvenience cost functions with three or more linear pieces, as suggested by Sexton and Choi (1986), lead to the same type of inconvenience cost function for a segment but with multiple linear pieces. Obviously, this adds another degree of complexity to the three cases considered above.

5 Inversion of Resource Extension Functions

The main question to be answered in this section is *how to define inverse REFs* so that they are useful for the following two tasks:

- How can we *check the feasibility* of a segment without iteratively applying the REFs for all arcs of the segment and checking intermediate resource consumptions against upper bounds? More precisely, given a segment $P = (0, 1, \ldots, p)$ and an initial resource consumption $T_0 \in [a_0, b_0]$, we want to find out efficiently whether $\mathcal{T}(P, T_0) \neq \varnothing$ holds.

- Define an inverse REF which allows a *reversal of the direction* in which resource variables are considered. Given any arc $(i, j) \in A$, is it possible to define a function which provides the resource consumption at the tail node i if this information is given for the head node j?

The analysis of $\mathcal{T}(P, T_0)$ in Section 3.3 has provided only one 'simple' criterion for the first task, given in Proposition 5. According to the precondition of Proposition 5, we assume for the entire section that *all REFs are non-decreasing.* The following proposition solves the first task, i.e., gives a criterion for the feasibility problem.

Proposition 10 Let $P = (0, 1, \ldots, p)$ be a path with non-decreasing REFs $f_{i-1,i}$ for $i \in \{1, \ldots, p\}$. If functions $f^{inv}_{i-1,i} : \mathbb{R}^R \to \mathbb{R}^R$ exist with properties

(INV)	$f_{i-1,i}(T) \leq T' \iff T \leq f^{inv}_{i-1,i}(T')$	for all $T \in (-\infty, b_{i-1}]$ and all $T' \in [a_i, \infty)$
(UBB)	$f^{inv}_{i-1,i}(T') \leq b_{i-1}$	for all $T' \in \mathbb{R}^R$
(NDI)	$f^{inv}_{i-1,i}$ is non-decreasing	

for all $i \in \{1, \ldots, p\}$, then P is resource-feasible w.r.t. the initial resource consumption T_0, i.e., $\mathcal{T}(P, T_0) \neq \varnothing$ if and only if $T_0 \leq f^{inv}_{01} \circ f^{inv}_{12} \circ \cdots \circ f^{inv}_{p-1,p}(b_p)$ holds.

We call any function f^{inv}_{ij} which satisfies (INV), (UBB), and (NDI) an *inverse REF of* f_{ij}. Note that conditions (NDI) and (UBB) are symmetrical to the conditions on f_{ij}, i.e., the REF has to be non-decreasing and $f_{ij}(T) \geq a_j$ must hold for any $T \in \mathbb{R}^R$. Condition (INV) is weaker than what is classically postulated for an inverse function. This weaker condition makes sense, since one cannot expect f_{ij} or f^{inv}_{ij} to be bijective. For instance, the preimage of a_j under f_{ij}, i.e., $f^{-1}_{ij}(\{a_j\}) \subseteq \mathbb{R}^R$ typically contains more than one point and, in that case, f_{ij} is not injective so that no classical left inverse exists (in terms of set theory).

According to the idea of generalizing REFs to segments, it is straightforward to define *the inverse segment REF* f^{inv}_P of a path $P = (0, 1, \ldots, p)$ as $f^{inv}_{01} \circ f^{inv}_{12} \circ \cdots \circ f^{inv}_{p-1,p}$. Note that f^{inv}_P is (per definition) non-decreasing and fulfills $f_P(T') \geq a_0$ for all $T' \in \mathbb{R}^R$. Proposition 10 provides a criterion that can be checked in $\mathcal{O}(R)$ time (independent of the length of the path) when $f^{inv}_P(b_{\omega(P)})$ is computed a priori. The effort for the computation of $f^{inv}_P(b_{\omega(P)})$ depends on the (inverse) REFs at hand.

If conditions (NDI), (UBB), and (INV) are satisfied for all REFs, we can easily invert the entire graph and associated solution processes for optimal schedule and optimal path problems. In order to see this, define $G' = (V, A')$ with the same set of node and reversed arcs $A' = \{(i, j) : (j, i) \in A\}$. With each arc $(i, j) \in A'$ we associate the REF $f'_{ij} := f^{inv}_{ji}$. For all $i \in V$, let $[a'_i, b'_i] := [a_i, b_i]$ be the resource intervals. Then any path $P = (0, 1, \ldots, p)$ is resource-feasible w.r.t. $(G, f_{ij}, [a_i, b_i])$ if and only if $P' = (p, \ldots, 1, 0)$ is resource-feasible w.r.t. $(G', f'_{ij}, [a'_i, b'_i])$. Classical solution approaches for SPPRCs are mainly based on dynamic programming (Irnich and Desaulniers, 2005). The implication for these approaches is that labels can be extended either forward along arcs (the traditional way) or backward in the opposite direction to the arcs. Furthermore, corresponding forward and backward labels can be compared and provide a simple criterion for checking the feasibility of the associated compound path.

Theorem 3 (Concatenation Theorem)
Let P_1 and P_2 be resource-feasible paths with $\omega(P_1) = \alpha(P_2)$. If P_1 has segment REF f_{P_1} and P_2 has inverse segment REF $f^{inv}_{P_2}$, then $P = P_1 \oplus P_2$ is resource-feasible if and only if $f_{P_1}(a_{\alpha(P_1)}) \le f^{inv}_{P_2}(b_{\omega(P_2)})$.

Our goal is now to develop an easy-to-prove sufficient condition for (INV) to hold. This condition will then be used to show that classical REFs as well as REFs with a pairwise max-term are invertible.

Proposition 11 Let $f_{ij} : \mathbb{R}^R \to \mathbb{R}^R$ and $f^{inv}_{ij} : \mathbb{R}^R \to \mathbb{R}^R$ be non-decreasing functions. A sufficient condition for property (INV) to hold is

$$\begin{aligned} (INV1) \quad & T_i \le f^{inv}_{ij}(f_{ij}(T_i)) \quad \text{for all } T_i \in (-\infty, b_i] \\ \text{and } (INV2) \quad & T_j \ge f_{ij}(f^{inv}_{ij}(T_j)) \quad \text{for all } T_j \in [a_j, \infty). \end{aligned}$$

We will use the above criterion for proving that classical REFs and REFs with pairwise max-term have inverse REFs for arcs as well as for segments.

5.1 *Classical REFs*

Next, we show that classical REFs have an inverse REF and that these can be generalized to segments.

Theorem 4 Classical REFs of the form $f_{ij}(T) = \max\{a_j, T + t_{ij}\}$ for all $(i, j) \in A$ can be inverted and the inverse can be generalized to segments such that properties (SCoF), (FNoC), and (CJS) hold.

(a) The function $f^{inv}_{ij}(T') = \min\{b_i, T' - t_{ij}\}$ is an inverse REF of f_{ij}.

(b) A segment $P = (0, 1, \ldots, p)$ has an REF as in Proposition 7 and an inverse REF of the form

$$f^{inv}_P(T') = \min\{b_P, T' - t_P\} \tag{19}$$

with $b_P = f^{inv}_{01} \circ \cdots \circ f^{inv}_{p-2,p-1} \circ f^{inv}_{p-1,p}(b_p) \in \mathbb{R}^R$ and $t_P = \sum_{i=0}^{p-1} t_{i,i+1} \in \mathbb{R}^R$ (properties (SCoF), (FNoC)).

(c) Let P_1, P_2 be segments in G with inverse REFs $f_{P_1}(T') = \min\{b_{P_1}, T' - t_{P_1}\}$ and $f^{inv}_{P_2}(T') = \min\{b_{P_2}, T' - t_{P_2}\}$, respectively. If $\omega(P_1) = \alpha(P_2)$, then

$$f^{inv}_{P_1 \oplus P_2}(T') = f^{inv}_{P_1} \circ f^{inv}_{P_2}(T') = \min\{b_{P_1 \oplus P_2}, T' - t_{P_1 \oplus P_2}\}$$

holds with $b_{P_1 \oplus P_2} = \min\{b_{P_1}, b_{P_2} - t_{P_1}\} \in \mathbb{R}^R$ and $t_{P_1 \oplus P_2} = t_{P_1} + t_{P_2} \in \mathbb{R}^R$ (this is property (CJS)).

Note that, in general, inverse REFs are not unique, since (INV), (UBB), and (NDI) do not impose strong restrictions on the values of $f^{inv}_{ij}(T')$ for $T' \not\ge a_j$. However, it is easy to show that inverse REFs f^{inv}_{ij} to classical REFs f_{ij} are uniquely defined for $T' \in [a_j, \infty)$: $f^{inv}_{ij}(T_j) = \min\{b_i, T_j - t_{ij}\}$.

5.2 *REFs with a pairwise max-term*

Similar results follow for REFs with a pairwise max-term (as introduced in Section 4.2).

Theorem 5 REFs $f_{ij}(T,T') = (\max\{a_j, T+t_{ij}, T'+u_{ij}\}, \max\{a'_j, T+t'_{ij}, T'+u'_{ij}\})$ for all $(i,j) \in A$ can be inverted and the inverse can be generalized to segments such that properties (SCoF), (FNoC), and (CJS) hold.

(a) The function $f^{inv}_{ij}(S,S') = (\min\{b_i, S-t_{ij}, S'-t'_{ij}\}, \min\{b'_i, S-u_{ij}, S'-u'_{ij}\})$ is an inverse REF of f_{ij}.

(b) A segment $P = (0,1,\ldots,p)$ has an REF as in Theorem 2 and an inverse REF of the form

$$f^{inv}_P(S,S') = (\min\{b, S-t_P, S'-t'_P\}, \min\{b', S-u_P, S'-u'_P\}).$$

with $(b,b') = f^{inv}_{01} \circ f^{inv}_{12} \circ \cdots \circ f^{inv}_{p-1,p}(b_p, b'_p)$ (properties (SCoF) and (FNoC)).

(c) Let P_1, P_2 be segments in G with inverse REFs

$$f^{inv}_{P_1}(S,S') = (\min\{b_{P_1}, S-t_{P_1}, S'-t'_{P_1}\}, \min\{b'_{P_1}, S-u_{P_1}, S'-u'_{P_1}\}),$$
$$f^{inv}_{P_2}(S,S') = (\min\{b_{P_2}, S-t_{P_2}, S'-t'_{P_2}\}, \min\{b'_{P_2}, S-u_{P_2}, S'-u'_{P_2}\}),$$

respectively. If $\omega(P_1) = \alpha(P_2)$, then

$$\begin{aligned} f^{inv}_{P_1 \oplus P_2}(S,S') = (&\min\{b_{P_1\oplus P_2}, S-t_{P_1\oplus P_2}, S'-t'_{P_1\oplus P_2}\},\\ &\min\{b'_{P_1\oplus P_2}, S-u_{P_1\oplus P_2}, S'-u'_{P_1\oplus P_2}\}),\end{aligned}$$

holds with $b_{P_1\oplus P_2} = \min\{b_{P_1}, b_{P_2}-t_{P_1}, b'_{P_2}-t'_{P_1}\}$, $t_{P_1\oplus P_2} = \max\{t_{P_1}+t_{P_2}, u_{P_2}+t'_{P_1}\}$, $t'_{P_1\oplus P_2} = \max\{t_{P_1}+t'_{P_2}, t'_{P_1}+u'_{P_2}\}$, $b'_{P_1\oplus P_2} = \min\{b'_{P_1}, b_{P_2}-u_{P_1}, b'_{P_2}-u'_{P_1}\}$, $u_{P_1\oplus P_2} = \max\{u_{P_1}+t_{P_2}, u'_{P_1}+u_{P_2}\}$, $u'_{P_1\oplus P_2} = \max\{u_{P_1}+t'_{P_2}, u'_{P_1}+u'_{P_2}\}$ (this is property (CJS)).

The special cases of VRPSDP and VRP with limited time on duty or limited waiting time are analyzed in more detail.

VRPSDP It follows from Theorem 5 that

$$f^{inv}_{ij} = (\min\{Q, S-p_j, S'-p_j\}, \min\{Q, S'-d_j\})$$

is an inverse REF of (7) for all $(i,j) \in A$. If we assume $S, S' \le Q$ and that all pickup quantities p_i and delivery quantities d_i are non-negative, the min-term with Q can be omitted (the assumption does not hold for applications with multiple use of vehicles). Formulas for segments $P = (0,1,\ldots,p)$ can be simplified and become

$$f^{inv}_P(S,S') = \left(\min\{S - \sum_{i=1}^{p} p_i, S' - mL^P\}, \quad S' - \sum_{i=1}^{p} d_i\right) \tag{20}$$

where mL^P is the same constant representing the maximum load that occurs in Formula (18). The interpretation of the above formula for the inverse segment REF is that the resource $r = pick$ (the quantity picked up before leaving node 0) is constrained by the entire load picked up on the segment as well as the maximum load that occurs on P. The second resource $r = mL$ for the maximum load is constrained only by the quantity that has to be delivered along P.

Waiting Times and Times on Duty It has been shown in Section 2.4.4 that waiting times, waiting costs, and times on duty can be modeled by means of non-decreasing REFs. The resource $r = time$ for earliest time of service is a resource with a classical update given by (3).

The four resources, $r = wait$, $r = hlp$, $r = duty$, and $r = hlp'$ are two pairs of interdependent resources which depend on each other with REFs as stated in (17). Therefore, an REF with resources $\{time, wait, hlp, duty, hlp'\}$ can be inverted. Hence, all aspects of time on duty and limited waiting times can be handled by REFs that are invertible and can be generalized to segments.

5.3 General REFs

With the cases discussed in Section 4.4, it is easy to see that REFs for modeling soft time windows either have REFs or inverse REFs with at least one decreasing component. The soft time window case never satisfies the conditions of Proposition 10. We therefore discuss the case of load-dependent costs only.

Load-Dependent Costs The only way of defining an inverse REF of (5) for load-dependent costs is

$$f_{ij}^{inv}(T_j^{cost}, T_j^{load}) = \Big(T_j^{cost} - c_{ij}(T_j^{load} - d_j), \quad T_j^{load} - d_j\Big)$$

(for the sake of simplicity, we have omitted the min-term with the upper bounds b_i^{cost} and b_i^{load} for both resources as they are not necessary if all demands and costs are non-negative and $T_j \leq b_j$). As stated before, f_{ij} is non-decreasing if and only if c_{ij} is non-decreasing. The opposite holds for f_{ij}^{inv}, i.e., f_{ij}^{inv} is non-decreasing if and only if c_{ij} is non*in*creasing. Hence, only constant cost functions c_{ij} lead to invertible REFs with property (INV). It is not possible to invert the underlying graph as needed for bidirectional shortest-path algorithms or the second application mentioned in the introduction.

Nevertheless, a proper inversion is typically not necessary for feasibility checking because the resource $r = cost$ is normally not bounded from above. To clarify that point, consider Propositions 10 and 3. Here, the criterion is to check T_0 against an upper bound $f_P^{inv}(b_p)$ (or $f_{P_1}(T_0) \leq f_{P_2}^{inv}(b_p)$, resp.). If the cost component is unconstrained, i.e., $b_i^{cost} = \infty$ for all $i \in V$, then $f_P^{inv,cost}(b_p) = f_{P_2}^{inv,cost}(b_p) = \infty$. Thus, it is possible to restrict the feasibility check to all resources except resource $r = cost$. Note further that a cost computation over several segments can be done using forward segment REFs, e.g., the cost of $P_1 + P_2$ is $(f_{P_2} \circ f_{P_1}(T_0))^{cost}$ so that no inverse REFs are needed for this task.

6 An Illustrative Example

The example is defined on the network (V, A) with time windows $[a_i, b_i]$ and the pickup p_i and delivery d_i demands for $i \in V = \{1, 2, \ldots, 8\}$ depicted in Figure 3. The intention is to model a VRP with time windows and simultaneous deliveries and pickups. For the sake of simplicity, we assume that all travel times are $t_{ij} = 2$ and that each visit of a node implies a profit of one unit, i.e., $c_{ij} = -1$ for all $(i, j) \in A$. The vehicle capacity is $Q = 20$.

The non-decreasing REF on an arc $(i, j) \in A$ is given by

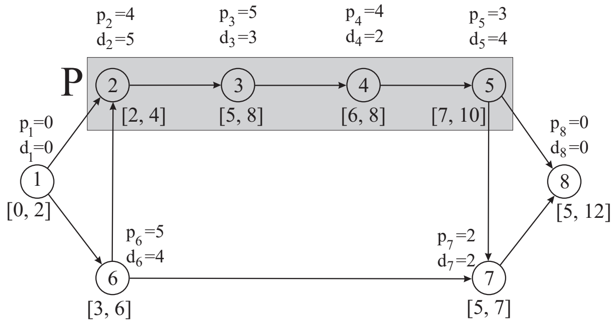

Fig. 3. Example Graph for a VRP with Time Windows and Simultaneous Delivery and Pickup

$$
\begin{aligned}
& f_{ij}(T_i^{cost}, T_i^{time}, T_i^{pick}, T_i^{mL}) \\
& = \Big(T_i^{cost} + c_{ij}, \max\{a_j, T_i^{time} + t_{ij}\}, T_i^{pick} + p_j, \max\{T_i^{pick} + d_j, T_i^{mL} + p_j\}\Big)
\end{aligned}
$$

and resource windows are

$$
(T_i^{cost}, T_i^{time}, T_i^{pick}, T_i^{mL}) \in [0, \infty) \times [a_i, b_i] \times [p_i, Q] \times [\max\{p_i, d_i\}, Q]
$$

We first consider path $P = (2, 3, 4, 5)$ as a (non-decomposable) segment. The computation of the segment REF f_P can use the Formulas (15) and (18) yielding

$$
\begin{aligned}
& f_P(T_2^{cost}, T_2^{time}, T_2^{pick}, T_2^{mL}) \\
& = (T_2^{cost} - 3, \max\{9, T_2^{time} + 6\}, T_2^{pick} + 12, \max\{T_2^{pick} + 13, T_2^{mL} + 9\})
\end{aligned}
$$

The interpretation of these values is as follows: Along path P, the accumulated profit is 3 (cost -3). The earliest arrival time at the destination node 5 of P is 9, while the accumulated travel and service time along P is 6. Concerning the demands, along the path P (excluding the first node 2) a demand of $\sum_{i=3}^{5} p_i = 12$ is picked up and a demand of $\sum_{i=3}^{5} d_i = 13$ is delivered. The maximum load that occurs along the path P depends on the initial conditions at (or more precisely *when leaving*) node 2. If the vehicle is empty (nothing picked up or delivered at node 2), the maximum load of 13 occurs on the arc (4,5), since $p_3 + p_4 = 9$ is already picked up and $d_5 = 4$ has to be delivered to node 5. Furthermore, if the initial conditions are such that $T_2^{mL} \leq T_2^{pick} + 4$ holds, then the maximum load along P is $T_2^{pick} + 13$ and occurs on the arc $(4, 5)$ again. Otherwise, the maximum load is $T_2^{mL} + 9$ and occurs on an arc preceding P. With $a_2 = (0, 2, 4, 5)^\top$, we see that $f_P(a_2) = (-3, 9, 16, 17)^\top$ is the minimum resource consumption at the final node 5 of P.

Using (19) and (20), the inverse segment REF for $P = (2, 3, 4, 5)$ is

$$
\begin{aligned}
& f_P^{inv}(S_5^{cost}, S_5^{time}, S_5^{pick}, S_5^{mL}) \\
& = (T_5^{cost} + 3, \min\{4, T_5^{time} - 6\}, \min\{T_5^{pick} - 12, T_5^{mL} - 13\}, T_5^{mL} - 9\})
\end{aligned}
$$

With $b_5 = (\infty, 10, 20, 20)^\top$, one gets $f_P^{inv}(b_5) = (\infty, 4, 7, 11)^\top \geq a_2$. Therefore, Proposition 3 guarantees that segment P is resource-feasible.

The advantage of having the (inverse) segment REF for P is that multiple REF evaluations and checks against upper bounds can be avoided. A path O preceding P yields a feasible concatenation $O \oplus P$ if the resource consumption at the end of O is less than or equal to $f_P^{inv}(b_5)$ (a single comparison). Additionally, REF and inverse REF for $O \oplus P$ can be computed in constant time from the corresponding REFs of O and P.

In order to compute all resource-feasible paths in the network (V, A) containing P, one simply has to consider the five segments $O = (1, 2)$, $O' = (1, 6, 2)$, $P = (2, 3, 4, 5)$, $Q = (5, 8)$, and $Q' = (5, 7, 8)$. Since $f_{(1,6,2)}(a_1) = f_{62} \circ f_{12}((0, 0, 0, 0)^\top) = (-2, 5, 9, 10)^\top \not\leq f_P^{inv}(b_5) = (\infty, 4, 7, 11)^\top$, the concatenation of O' and P is infeasible. The same holds for the concatenation of P and Q', because of $f_P(a_2) = (-3, 9, 16, 17)^\top \not\leq f_{(5,7,8)}^{inv}(b_8) = f_{57}^{inv} \circ f_{78}^{inv}((\infty, 12, 20, 20)^\top) = f_{57}^{inv}((\infty, 7, 20, 20)^\top) = (\infty, 5, 18, 18)^\top$. Hence, the only path containing P left is $O \oplus P \oplus Q = (1, 2, 3, 4, 5, 8)$. We can easily check that O and Q are resource-feasible. The fact that $f_{12}(a_1) = (-1, 2, 4, 5)^\top \leq f_P^{inv}(b_5) = (\infty, 4, 7, 11)^\top$ shows that $O \oplus P$ is resource-feasible. With Propositions 8(a) and 5(c) we can compute the REF for the concatenation $O \oplus P$,

$$
\begin{aligned}
& f_{O\oplus P}(T_1^{cost}, T_1^{time}, T_1^{pick}, T_1^{mL}) \\
& = (T_1^{cost} - 4, \max\{8, T_1^{time} + 8\}, T_1^{pick} + 16, \max\{T_1^{pick} + 17, T_1^{mL} + 14\}),
\end{aligned}
$$

so that $f_{O\oplus P}(a_1) = f_{O\oplus P}((0, 0, 0, 0)^\top) = (-4, 8, 16, 17)^\top$ holds. Finally, $f_Q^{inv}(b_8) = f_{58}^{inv}((\infty, 12, 20, 20)^\top) = (\infty, 10, 20, 20)^\top \geq f_{O\oplus P}(a_1)$ implies that $O \oplus P \oplus Q$ is resource-feasible.

7 Conclusions

This paper has provided a theoretical foundation for defining, analyzing, and manipulating resource-constrained paths. REFs are the main tool for mathematically describing complex interdependencies between resources. From a modeling point of view, the text has surveyed different types of real-world constraints, mainly taken from the area of vehicle routing and crew scheduling. The unified model by Desaulniers *et al.* (1998) has shown that nearly all types of deterministic VRPs and SPVCs considered in the literature are covered by REF-based models.

From an algorithmic point of view, efficient REF handling is desirable in exact as well as in heuristic solution methods for these problems. Exact solutions procedures (column generation or Lagrangean relaxation integrated into branch-and-bound) require the computation of least cost resource-constrained paths. While efficient solution methods for SPPRC can be found in a separate survey (Irnich and Desaulniers, 2005), the focus here has been on methods that *support several types of acceleration procedures.* Bidirectional shortest-path algorithms (Salani, 2005), reduced cost arc-elimination procedures (Irnich, 2007) and the dynamic aggregation method by Elhallaoui *et al.* (2005) need well-defined concepts for inverting the solution process and for simplifying the computations when segments are shrunk. Local search-based procedures are used in all kinds of traditional and modern metaheuristics. Irnich (2006) explains the usefulness of segment REFs for the evaluation of neighbor solutions if represented by a giant tour and, especially, for developing efficient feasibility checking procedures.

Resource-constrained paths and REFs can be seen as the key concept for bridging the gap between exact and heuristic methods for *rich VRPs* (Hasle *et al.*, 2003). The detailed analysis undertaken here clarifies which types of REFs are well-suited for different algorithmic tasks. Non-decreasing REFs are imperative for the application of easy-to-implement dominance rules, leading to well-performing dynamic programming labeling procedures for solving the subproblem in exact approaches. These methods are applicable to VRPs with (multiple) time windows, path length constraints, multiple use of vehicles, time-dependent travel times, periodic and multiple depot versions of specific VRPs etc. We have pointed out the additional assumptions which guarantee constant time feasibility checks: REFs have to be invertible and segment REFs need to have representations with a fixed number of coefficients (independent of their length, property (FNoC)). In particular, properties (FNoC) and (CJS) enable $\mathcal{O}(1)$ procedures for the determination of whether the concatenation of a priori given segments is a resource-feasible path. The same assumptions (invertible REF, (FNoC), (CJS)) are desirable for arc elimination and the dynamic aggregation method for the acceleration of branch-and-price algorithms. Classical REFs and REFs with pairwise max-term already have these 'good' properties. Several examples of real-world constraints fall into this scheme, for instance, non-trivial time/schedule characteristics (waiting times, times on duty) or simultaneous delivery and pickup requirements. Other REFs cannot guarantee (FNoC), but their generalization to segments imposes REFs of well-defined form (i.e., property (SCoF); for instance, REFs for multiple time windows, certain load-dependent cost functions, and non-decreasing inconvenience costs). Even without guaranteed constant time feasibility checking, these segment REFs and their inverses can still be useful for speeding up exact and heuristic algorithms.

References

Ahn, B. and Shin, J. (1991). Vehicle-routeing with time windows and time-varying congestions. *Journal of the Operational Research Society*, **42**, 393–400.

Ahuja, R., Magnanti, T., and Orlin, J. (1993). *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, Englewood Cliffs, New Jersey.

Avella, P., Boccia, M., and Sforza, A. (2004). Resource constrained shortest path problems in path planning for fleet management. *Journal of Mathematical Modelling and Algorithms*, **3**(1), 1–17.

Beasley, J. and Christofides, N. (1989). An algorithm for the resource constrained shortest path problem. *Networks*, **19**, 379–394.

Borndörfer, R., Grötschel, M., and Löbel, A. (2001). Scheduling duties by adaptive column generation. Technischer Bericht (ZIB-Report) 01-02, Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB), Berlin.

Christiansen, M. (1996). *Inventory and Time Constrained Ship Routing – A Mathematical Approach*. Ph.D. dissertation, Norwegian University of Science and Technology, Trondheim, Norway.

Christofides, N. and Eilon, S. (1969). An algorithm for the vehicle-dispatching problem. *Operational Research Quarterly*, **20**(3), 309–318.

Desaulniers, G. and Villeneuve, D. (2000). The shortest path problem with time windows and linear waiting costs. *Transportation Science*, **34**(3), 312–319.

Desaulniers, G., Desrosiers, J., Ioachim, I., Solomon, M., Soumis, F., and Villeneuve, D. (1998). A unified framework for deterministic time constrained vehicle routing and crew scheduling problems. In T. Crainic and G. Laporte, editors, *Fleet Management and Logistics*, chapter 3, pages 57–93. Kluwer Academic Publisher, Boston, Dordrecht, London.

Desaulniers, G., Desrosiers, J., Lasry, A., and Solomon, M. M. (1999). Crew pairing for a regional carrier. In N. Wilson, editor, *Computer-Aided Transit Scheduling, Lecture Notes in Computer Science 471*, pages 19–41. Springer, Berlin.

Desaulniers, G., Desrosiers, J., and Solomon, M., editors (2005). *Column Generation*. Springer.

Desaulniers, G., Lessard, F., and Hadjar, A. (2006). Tabu search, generalized k-path inequalities, and partial elementarity for the vehicle routing problem with time windows. Les Cahiers du GERAD G-2006-45, GERAD, École des Hautes Études Commerciales, Montréal, Canada.

Desrochers, M. and Soumis, F. (1988). A generalized permanent labelling algorithm for the shortest path problem with time windows. *Information Systems and Operations Research*, **26**(3), 191–212.

Dethloff, J. (2002). Relation between vehicle routing problems: An insertion heuristic for the vehicle routing problem with simultaneous delivery and pick-up applied to the vehicle routing problem with backhauls. *Journal of the Operational Research Society*, **53**, 115–118.

Dumas, Y., Soumis, F., and Desrosiers, J. (1990). Optimizing the schedule for a fixed vehicle path with convex inconvenience costs. *Transportation Science*, **24**(2), 145–152.

Dumas, Y., Desrosiers, J., and Soumis, F. (1991). The pick-up and delivery problem with time windows. *European Journal of Operational Research*, **54**, 7–22.

Elhallaoui, I., Villeneuve, D., Soumis, F., and Desaulniers, G. (2005). Dynamic aggregation of set-partitioning constraints in column generation. *Operations Research*, **53**(4), 632–645.

Gamache, M., Soumis, F., Villeneuve, D., Desrosiers, J., and Gélinas, E. (1998). The preferential bidding systems at Air Canada. *Transportation Science*, **32**(3), 246–255.

Gamache, M., Soumis, F., and Marquis, G. (1999). A column generation approach for large-scale aircrew rostering problems. *Operations Research*, **47**(2), 247–263.

Gendreau, M., Dejax, P., Feillet, D., and Gueguen, C. (2005). Vehicle routing with time windows and split deliveries. Unpublished technical report, Laboratoire d´Informatique d´Avignon, Avignon, France. (Work in progress).

Gietz, M. (1994). *Computergestützte Tourenplanung mit zeitkritischen Restriktionen*. Physica Verlag, Heidelberg.

Gribkovskaia, I., Halskau sr, Ø., Laporte, G., and Vlček, M. (2006). General solutions to the single vehicle routing problem with pickups and deliveries. Technical report, Molde University College, Molde, Norway.

Halse, K. (1992). *Modelling and Solving Complex Vehicle Routing Problems*. PhD dissertation No. 60, IMSOR, Technical University of Denmark, Lyngby, Denmark.

Hasle, G., Kloster, O., Bräysy, O., Gendreau, M., Kjenstad, D., and Riise, A. (2003). A conceptual model for rich VRPs. Presentation at the EURO/INFORMS Joint International Meeting, Istanbul.

Helgason, R., Kennington, J., and Stewart, B. (1993). The one-to-one shortest-path problem: An empirical analysis with the two-tree Dijkstra algorithm. *Computational Optimization and Applications*, **1**, 47–75.

Hempsch, C. and Irnich, S. (2007). Vehicle-routing problems with inter-tour resource constraints. Technical Report 2007-01, Deutsche Post Endowed Chair of Optimization of Distribution Networks, RWTH Aachen University, Aachen, Germany. Available at `www.dpor.rwth-aachen.de`.

Hill, A. and Benton, W. (1992). Modelling intra-city time-dependent travel speeds for vehicle scheduling problems. *Journal of the Operational Research Society*, **43**, 343–351.

Ibaraki, T., Imahori, S., Kubo, M., Masuda, T., Uno, T., and Yagiura, M. (2005). Effective local search algorithms for routing and scheduling problems with general time-window constraints. *Transportation Science*, **39**(2), 206–232.

Ioachim, I., Desrosiers, J., Soumis, F., and Bélanger, N. (1994). Fleet routing with schedule synchronization. Les Cahiers du GERAD G-94-48, École des HEC, Montréal, Canada, H3T 2A7.

Ioachim, I., Gélinas, S., Desrosiers, J., and Soumis, F. (1998). A dynamic programming algorithm for the shortest path problem with time windows and linear node costs. *Networks*, **31**, 193–204.

Irnich, S. (2006). A unified modeling and solution framework for vehicle routing and local search-based metaheuristics. Technical Report 2006-02, Deutsche Post Endowed Chair of Optimization of Distribution Networks, RWTH Aachen University, Aachen, Germany. Available at `www.dpor.rwth-aachen.de`.

Irnich, S. (2007). Speeding up column-generation algorithms by using reduced costs of paths to eliminate arcs. Technical Report 2007-02, Deutsche Post Endowed Chair of Optimization of Distribution Networks, RWTH Aachen University, Aachen, Germany. Available at `www.dpor.rwth-aachen.de`.

Irnich, S. and Desaulniers, G. (2005). Shortest path problems with resource constraints. In Desaulniers *et al.* (2005), chapter 2, pages 33–65.

Irnich, S., Funke, B., and Grünert, T. (2006). Sequential search and its application to vehicle-routing problems. *Computers & Operations Research*, **33**(8), 2405–2429.

Jepsen, M., Spoorendonk, S., Petersen, B., and Pisinger, D. (2006). A non-robust branch-and-cut-and-price algorithm for the vehicle routing problem with time windows. DIKU Technical Report no. 06/03, Dept. of Computer Science, University of Copenhagen, Copenhagen, Denmark.

Malandraki, C. and Daskin, M. (1992). Time dependent vehicle routing problems: Formulations, properties and heuristic algorithms. *Transportation Science*, **26**(3), 185–200.

Min, H. (1989). The multiple vehicle routing problem with simultaneous delivery and pick-up points. *Transportation Research*, **23**, 377–386.

Salani, M. (2005). *Branch-and-price algorithms for Vehicle Routing Problems*. Ph.D. dissertation, Università degli Studi di Milano, Facoltà die Scienze Matematiche, Fisiche e Naturali, Dipartimento di Tecnologie dell'Informatione, Milan, Italy.

Sexton, T. and Bodin, L. (1985a). Optimizing single vehicle many-to-many operations with desired delivery times: I. Scheduling. *Transportation Science*, **19**(4), 378–410.

Sexton, T. and Bodin, L. (1985b). Optimizing single vehicle many-to-many operations with desired delivery times: II. Routing. *Transportation Science*, **19**(4), 411–435.

Sexton, T. and Choi, Y. (1986). Pickup and delivery of partial loads with "soft" time windows. *American Journal of Mathematical and Management Sciences*, **6**(3&4), 369–398.

Solomon, M. (1987). Algorithms for the vehicle routing and scheduling problem with time window constraints. *Operations Research*, **35**(2), 254–265.

Taillard, D., Laporte, G., and Gendreau, M. (1996). Vehicle routeing with multiple use

of vehicles. *Journal of the Operational Research Society*, **47**(8), 1065–1070.
Vance, P., Barnhart, C., Johnson, E., and Nemhauser, G. (1997). Airline crew scheduling: A new formulation and decomposition algorithm. *Operations Research*, **45**(2), 188–200.

Appendix

The appendix contains proofs for all propositions and theorems stated in the text.

PROOF of Proposition 1. Immediately from definition (1). ✓

PROOF of Proposition 2. The set $\mathcal{T}((0), T_0)$ is identical to $[T_0, b_0]$ and, therefore, compact. Using identity (12), the fact that operator $\llcorner$ maps compact sets to closed sets, and that the intersection with an interval maps closed sets to compact sets, yields that $\mathcal{T}((0,1), T_0)$ is compact. Iterating the same chain of arguments for all prefix paths $P' = (0, 1, \ldots, p')$, $p' \leq p$ of P implies that each set $\mathcal{T}(P', T_0)$ is compact, too. The proposition follows with $p' = p$. ✓

PROOF of Proposition 3. It follows from definition (1) that the statement is true for $P = (0)$, i.e. for $p = 0$. By induction, we can assume that the prefix path $P^- = (0, 1, \ldots, p-1)$ fulfills $\mathcal{T}(P^-, T_0) = [\hat{a}_{p-1}(T_0), b_{p-1}]$ with $\hat{a}_{p-1}(T_0)$ defined as stated above. From Proposition 1 it follows that $\mathcal{T}(P, T_0) = f_{p-1,p}([\hat{a}_{p-1}(T_0), b_{p-1}])^{\llcorner} \cap [a_p, b_p]$ holds. Since $f_{p-1,p}$ is non-decreasing, we have $f_{p-1,p}(\hat{a}_{p-1}(T_0)) \leq f_{p-1,p}(T)$ for all $T \in [\hat{a}_{p-1}(T_0), b_{p-1}]$. Hence, $f_{p-1,p}([\hat{a}_{p-1}(T_0), b_{p-1}])^{\llcorner} = f_{p-1,p}(\hat{a}_{p-1}(T_0))^{\llcorner}$ (note that on the LHS $\llcorner$ is applied to a set and on the RHS it is applied to a vector). The definition of the operator $\llcorner$ implies $x^{\llcorner} \cap [a, b] = [\max\{a, x\}, b]$ for all vectors $x, a, b \in \mathbb{R}^R$ and, therefore, $f_{p-1,p}(\hat{a}_{p-1}(T_0))^{\llcorner} \cap [a_p, b_p] = [\max\{a_p, f_{p-1,p}(\hat{a}_{p-1}(T_0))\}, b_p]$. The left bound of the interval coincides with $\hat{a}_p(T_0)$ and the proposition is proven. ✓

PROOF of Proposition 4. Induction over the length p of P: Obviously, for $P = (0)$ the set $\mathcal{T}(P, T_0)$ is $[T_0, b_0]$, which is empty for $T_0 \not\leq b_0$ and an R-dimensional interval, otherwise.

For $p \geq 1$, let P' be the prefix path $(0, 1, \ldots, p-1)$. The assumption of the induction is that $\mathcal{T}(P', T_0)$ is the empty set or a polytope. In the latter case, it follows that $f_{p-1,p}$ map $\mathcal{T}(P', T_0)$ to a polytope (because each lineare function maps a polytope $conv(\{T^1, \ldots, T^q\})$ to a polytope $conv(\{f(T^1), \ldots, f(T^q)\})$). Using identity (12) for the polytope $f(\mathcal{T}(P', T_0))$, the operator $\llcorner$ transforms it to a polyhedron with the unit vectors $e_1, e_2, \ldots, e_R \in \mathbb{R}^R$ as extreme rays. Finally, the intersection of $f_{p-1,p}(\mathcal{T}(P', T_0))^{\llcorner}$ with $[a_p, b_p]$ cuts the polyhedron to a polytope again. ✓

PROOF of Proposition 5. If $\hat{a}_i(T_0) \leq b_i$ holds for all $i \in \{0, 1, \ldots, p\}$ one can use these vectors as resource vectors T_i in definition (1) to see that P is feasible, i.e., $\hat{a}_i(T_0) \in \mathcal{T}(P, T_0) \neq \varnothing$ holds. In contrast, if $\mathcal{T}(P, T_0) \neq \varnothing$, valid resource vectors T_i, $i \in \{1, \ldots, p\}$ exist fulfilling definition (1). Starting with $T_1 \geq \max\{a_1, f_{0,1}(T_0)\} = \max\{a_1, f_{0,1}(\hat{a}_0(T_0))\} = \hat{a}_1(T_0))$, the same arguments and the non-decreasing REF imply that $T_i \geq \hat{a}_i(T_0)$ holds for all $i \in \{1, \ldots, p\}$. Therefore, $\hat{a}_i(T_0) \leq T_i \leq b_i$ holds for all $i \in \{1, \ldots, p\}$. ✓

PROOF of Proposition 6. Direct calculus. ✓

PROOF of Proposition 7. For a segment $P = (0)$ consisting of a single node, Formula (15b) gives $a_P = a_0$ and $t_P = 0$, i.e., $f_P(T) = \max\{a_0, T\}$. This is the correct REF for the segment (0). A segment $P = (0, 1)$ coincides with an arc and the corresponding REF is of the required form. The statement for longer segments P follows by induction from Proposition 6 with the REF for the segments $P_1 = (0, \ldots, p-1)$ and $P_2 = (p-1, p)$. ✓

PROOF of Proposition 8. (a) follows directly from Proposition 6.
(b) follows from (a) with the three segments $P_1, (i, j)$, and P_2. ✓

PROOF of Theorem 1. Direct consequence of Propositions 7 and 8. ✓

PROOF of Proposition 9. Direct calculus. ✓

PROOF of Theorem 2. Follows from Proposition 9. For the sake of brevity, we do not present *closed* formulas for the computation of the coefficients t, u, t', u'. Nevertheless, Proposition 9 shows that property (CJS) holds, too. ✓

PROOF of Proposition 10. According to Proposition 5 and equality (14) the segment P is feasible w.r.t. T_0 if and only if $f_{i-1,i} \circ \cdots \circ f_{01}(T_0) \leq b_i$ holds for all $i \in \{0, \ldots, p\}$. Using property (INV) this is equivalent to $T_0 \leq f_{01}^{inv} \circ \cdots \circ f_{i-1,i}^{inv}(b_i)$ for all $i \in \{0, \ldots, p\}$. Finally, using properties (UBB) and (NDI), we can establish the following inequality

$$
\begin{aligned}
& f_{01}^{inv} \circ f_{12}^{inv} \circ \cdots \circ f_{p-2,p-1}^{inv} \circ \underbrace{f_{p-1,p}^{inv}(b_p)}_{\leq b_{p-1}} \\
\leq\ & f_{01}^{inv} \circ f_{12}^{inv} \circ \cdots \circ \underbrace{f_{p-2,p-1}^{inv}(b_{p-1})}_{\leq b_{p-1}} \\
\leq\ & \ldots \\
\leq\ & f_{01}^{inv} \circ \underbrace{f_{01}^{inv}(b_2)}_{\leq b_1} \\
\leq\ & f_{01}^{inv}(b_1) \\
\leq\ & b_0
\end{aligned}
$$

for all the values T_0 is compared with. Summing up, P is feasible w.r.t. T_0 if and only $T_0 \leq f_{01}^{inv} \circ \cdots \circ f_{p-1,p}^{inv}(b_p)$ holds. ✓

PROOF of Theorem 3. Let for the sake of brevity, let $0 = alpha(P_1)$ and $p = \omega(P_2)$. The inequality $f_{P_1}(a_0) \leq f_{P_2}^{inv}(b_p)$ is equivalent to $a_0 \leq f_{P_1}^{inv} \circ f_{P_2}^{inv}(b_p) = f_P^{inv}(b_p)$. With $T_0 = a_0$ this is the criterion given in Proposition 10. ✓

PROOF of Proposition 11. We have to show property (INV), i.e., the equivalence $f_{ij}(T_i) \leq T_j \iff T_i \leq f_{ij}^{inv}(T_j)$.

'$\Longrightarrow$': $T_i \overset{(INV1)}{\leq} f_{ij}^{inv}(f_{ij}(T_i)) \overset{(NDI), f_{ij}(T_i) \leq T_j}{\leq} f_{ij}^{inv}(T_j)$

'$\Longleftarrow$': $f_{ij}(T_i) \overset{f_{ij} n.d., T_i \leq f_{ij}^{inv}(T_j)}{\leq} f_{ij}(f_{ij}^{inv}(T_j)) \overset{(INV2)}{\leq} T_j$ ✓

PROOF of Theorem 4. (a) Any function of the form $\min\{b, T - t\}$ is non-decreasing in T for arbitrary $b, t \in \mathbb{R}^R$. It remains to show that the above definition of f^{inv} has properties (INV1) and (INV2). Property (INV1) follows from

$$\begin{aligned} f^{inv}(f(T)) &= \min\{b_i, f(T) - t_{ij}\} = \min\{b_i, \max\{a_j, T + t_{ij}\} - t_{ij}\} \\ &= \min\{b_i, \max\{a_j - t_{ij}, T\}\} \geq \min\{b_i, T\} \\ &\geq T. \end{aligned}$$

Note that the last inequality holds because of $T \in (-\infty, b_i]$. Property (INV2) follows from

$$\begin{aligned} f(f^{inv}(T')) &= \max\{a_j, f^{inv}(T') + t_{ij}\} = \max\{a_j, \min\{b_i, T' - t_{ij}\} + t_{ij}\} \\ &= \max\{a_j, \min\{b_i + t_{ij}, T'\}\} \leq \max\{a_j, T'\} \\ &\leq T' \end{aligned}$$

Again, the last inequality holds, since $T' \in [a_j, \infty)$ is assumed.

(b) Follows from arguments analogue to those used in Proposition 7.

(c) Direct calculus. ✓

PROOF of Theorem 5. (a) We show that properties (INV1) and (INV2) hold for $(T, T') \in (-\infty, (b_i, b_i')]$ and $(S, S') \in [(a_j, a_j'), \infty)$. First,

$$\begin{aligned} & f_{ij}^{inv}(f_{ij}(T, T')) \\ &= (\min\{b_i, \max\{a_j, T + t_{ij}, T' + u_{ij}\} - t_{ij}, \max\{a_j', T + t_{ij}', T' + u_{ij}'\} - t_{ij}'\}, \\ &\quad \min\{b_i', \max\{a_j, T + t_{ij}, T' + u_{ij}\} - u_{ij}, \max\{a_j', T + t_{ij}', T' + u_{ij}'\} - u_{ij}'\}) \\ &= (\min\{b_i, \max\{a_j - t_{ij}, T, T' + u_{ij} - t_{ij}\}, \max\{a_j' - t_{ij}', T, T' + u_{ij}' - t_{ij}'\}, \\ &\quad \min\{b_i', \max\{a_j - u_{ij}, T + t_{ij} - u_{ij}, T'\}, \max\{a_j' - u_{ij}'\}, T + t_{ij}' - u_{ij}'\}, T'\}) \\ &\geq (\min\{b_i, T, T\}, \min\{b_i', T', T'\}) \\ &= (T, T'), \end{aligned}$$

which proves (INV1) and second

$$\begin{aligned} & f_{ij}(f_{ij}^{inv}(S, S')) \\ &= (\max\{a_j, \min\{b_i, S - t_{ij}, S' - t_{ij}'\} + t_{ij}, \min\{b_i', S - u_{ij}, S' - u_{ij}'\} + u_{ij}\}, \\ &\quad \max\{a_j', \min\{b_i, S - t_{ij}, S' + t_{ij}'\} + t_{ij}', \min\{b_i', S - u_{ij}, S' + u_{ij}'\} + u_{ij}'\}) \\ &= (\max\{a_j, \min\{b_i + t_{ij}, S, S' - t_{ij}' + t_{ij}\}, \min\{b_i' + u_{ij}, S, S' - u_{ij}' + u_{ij}\}\}, \\ &\quad \max\{a_j', \min\{b_i + t_{ij}', S - t_{ij} + t_{ij}', S'\}, \min\{b_i' + u_{ij}', S - u_{ij} + u_{ij}', S'\}\}) \\ &\leq (\max\{a_j, S, S\}, \max\{a_j', S', S'\}) \\ &= (S, S'), \end{aligned}$$

which proves (INV2).

(c) Direct calculus. Repeated application of (c) for all arcs of the path P yields (b). ✓