Noname manuscript No.
(will be inserted by the editor)

Delay-Constrained Shortest Paths: Approximation Algorithms and Second-Order Cone Models

Antonio Frangioni* · **Laura Galli** · **Maria Grazia Scutellà**

Abstract Routing real-time traffic with maximum packet delay in contemporary telecommunication networks requires not only choosing a path, but also reserving transmission capacity along its arcs, as the delay is a nonlinear function of both components. The problem is known to be solvable in polynomial time under quite restrictive assumptions, i.e., Equal Rate Allocations (all arcs are reserved the same capacity) and identical reservation costs, whereas the general problem is $\mathcal{NP}$-hard. We first extend the approaches to the ERA version to a pseudo-polynomial Dynamic Programming one for integer arc costs, and a FPTAS for the case of general arc costs. We then show that the general problem can be formulated as a mixed-integer Second-Order Cone (SOCP) program, and therefore solved with off-the-shelf technology. We compare two formulations: one based on standard big-M constraints, and one where Perspective Reformulation techniques are used to tighten the continuous relaxation. Extensive computational experi-

* Corresponding Author

Dipartimento di Informatica, Università di Pisa, Largo B. Pontecorvo 3, 56127 Pisa – Italy

E-mail: [frangio,galli,scut]@di.unipi.it

ments on both real-world networks and randomly-generated realistic ones show that the ERA approach is fast and provides an effective heuristic for the general problem whenever it manages to find a solution at all, but it fails for a significant fraction of the instances that the SOCP models can solve. We therefore propose a three-pronged approach that combines the fast running time of the ERA algorithm and the effectiveness of the SOCP models, and show that it is capable of solving realistic-sized instances with high accuracy at different levels of network load in a time compatible with real-time usage in an operating environment.

Keywords Delay-constrained Routing · Approximation Algorithms · Mixed-Integer NonLinear Programming · Second-Order Cone Model · Perspective Reformulation

Mathematics Subject Classification (2000) 90C11 · 90C25 · 90C30 · 90C90

1 Introduction

The development of computer networks capable to support high bandwidth applications while having stringent Quality of Service (QoS) guarantees is a relevant practical issue, since many applications over IP networks (e.g., industrial control systems, remote sensing and surveillance systems, live Internet Protocol Television and IP Telephony) require real-time guarantees, that is, controlled end-to-end delay. Hence, Internet Service Providers are required to negotiate delay bound within their Service Level Agreements, which in turn requires appropriate traffic engineering support. From an optimization point of view, this implies both computing paths and reserving resources along the paths of the network, since the maximum delay of a flow depends on both.

Even in the single-flow case, this problem is therefore significantly more difficult than usual shortest path routing problems. Several practical approaches have been proposed [1], where delays are assumed to be link-additive in order to simplify the problem; however, delay bounds do depend on the amount of reserved resources at each link, usually in a nonlinear and non-additive way. Efficient algorithms have been devised for the special case where the resource allocation is uniform on all the links of a path, which is called the Equal Rate Allocation (ERA) approach, and when the objective function is basically the arc/node count of the path [2, 3]. However, even for fixed paths ERA has been shown to be highly suboptimal when addressing the more general delay-constrained routing case [4], thus requiring more resources than those strictly necessary to ensure a given delay bound for a given flow, and possibly failing to find feasible delay-constrained routings even when they exist.

In this paper, we mark a first step in the direction of joint path computation and resource reservation under delay bound constraints by considering the more general scenario where the resource allocation may be different on the links of the considered path. We concentrate on the Single-Flow Single-Path Delay-Constrained Routing problem (SFSP-DCR), which is already $\mathcal{NP}$-hard since it generalizes the Constrained Shortest Path problem (CSP) [5–7]; however, due to the nonlinear nature of the delay constraints, adapting known approaches for CSP is not straightforward. We first consider the ERA version of the problem (ERA-SFSP-DCR), i.e., the case where all arcs in the path are allocated the same amount of resource, which is solvable in polynomial time in the case of unit arc costs, and derive a pseudo-polynomial time algorithm for integer arc costs and a FPTAS for general costs. We then consider the general case and we show that the problem can be formulated as a convex Mixed-Integer Non-Linear Optimization problem (MINLP), and in particular, as a Mixed-Integer Second-Order Cone problem (MISOCP) that can be solved by efficient general-purpose tools. We present two MISOCP models for the problem: a straightforward one based on big-M constraints, and an improved one where convex-envelope techniques are used to tighten the continuous relaxation. Extensive computational experiments on both real-world networks and randomly-generated realistic ones show that the exact algorithms for ERA-SFSP-DCR are extremely fast and provide a surprisingly effective heuristic for the general problem whenever they manage to find a solution at all, but they fail for a significant fraction of the instances that the (MI)SOCP models can solve. We therefore propose a three-pronged approach that combines the fast running time of the ERA algorithms and the effectiveness of the SOCP models, and show that it is capable of solving realistic-sized instances with high accuracy at different levels of network load in a time compatible with real-time usage in an operating environment.

2 The Delay-Constrained Routing Problem

A telecommunication network is represented by a directed graph $G = (N, A)$, with $n = |N|$ and $m = |A|$. Our problem is to route one single "new" flow on the network along a minimum cost path, where the cost is any linear function of the reserved capacities on the traversed arcs, with a constraint on the maximum delay that any packet may incur during the trip. For this, we assume our flow to be characterized by an origin $s \in N$, a destination $d \in N \setminus \{s\}$ and, in general, an *arrival curve* $\mathcal{A}(t) : \mathbb{R}_+ \to \mathbb{R}_+$ specifying how many more bits of that flow can enter the origin

s with respect to those entered t instants before; in other words, if the *arrival function* $\mathcal{F}(t)$ measures how many bits have entered the origin at time t, we have $\mathcal{F}(\bar{t}+t) - \mathcal{F}(\bar{t}) \leq \mathcal{A}(t)$ for all $\bar{t}$ and $t \geq 0$. For our purposes, we assume the arrival curve to be entirely specified by the two parameters σ (burst) and ρ (rate) of a *leaky-bucket traffic shaper* [8], so that $\mathcal{A}(t) = \sigma + t\rho$. Each link (arc) $(i, j) \in A$ in the network is characterized by a fixed *link delay* l_{ij}, a *physical link speed* w_{ij}, and a *reservable capacity* c_{ij} ($\leq w_{ij}$, since in general other flows are already present in the network at the time when the new one is routed). Each node $i \in N$ in the network is characterized by a given *node delay* n_i; furthermore, the *maximum transmit unit* L (i.e., the maximal size of any packet) is known and assumed constant. The flow has a *deadline* δ, which bounds from above the maximum time that every bit in the flow is allowed to spend traversing the network prior to reaching the destination; in other words, the *worst-case delay* of the flow must be at most δ. Given *link reservation costs* f_{ij} (i.e., the cost of reserving one unit of capacity on (i, j)), the *Single-Flow Single-Path Delay-Constrained Routing* (SFSP-DCR) problem requires to find one feasible *s*-*d* path and a feasible reservation of capacity for each of its arcs so that the flow can be routed along the path, with the given reserved capacities, by respecting the deadline (delay constraint) δ at the minimum possible reservation cost.

2.1 Delay Modeling

Formulating SFSP-DCR requires to specify how the worst-case delay of the flow is computed. This depends on several factors:

1. the selected *routing* for the flow, i.e., the selected *s*-*d* path P in G;

2. for each arc (i,j) of the chosen path P, the *reserved capacity* (or *rate*) $0 \leq r_{ij} \leq c_{ij}$ $(\leq w_{ij})$ for the flow along the arc;
3. the specific characteristics of the software/hardware systems at the nodes dictating how the flows entering and leaving the nodes are managed (intra-node scheduling of different flows, queues and buffer depths, ...).

The latter point requires a sophisticated analysis, that can be performed, e.g., via network calculus [9]. In all cases of interest here, the delay is finite only if the minimum reserved rate along the arcs of the path is at least as large as the rate ρ of the path, i.e.,

$$r_{ij} \geq \rho \qquad \forall (i,j) \in P \ . \tag{1}$$

Once (1) is satisfied, the general form of the delay for a given routing path P is

$$\frac{\sigma}{\min\{ r_{ij} \ : \ (i,j) \in P \}} + \sum_{(i,j) \in P} \big(\theta_{ij} + l_{ij} + n_i \big), \tag{2}$$

where θ_{ij} is the delay experienced by the flow on traversing the arc (i,j) that is due to the scheduling protocol. The exact form of θ_{ij} depends on the details of the scheduling algorithm at nodes: following $[2,3]$ we assume

$$\theta_{ij} := \frac{L}{r_{ij}} + \frac{L}{w_{ij}}, \tag{3}$$

which corresponds to Strictly Rate-Proportional delay (e.g. $[8, 10, 11]$). Other slightly different forms of delay formulae exist, such that the Weakly Rate Proportional one, that have basically the same algebraic form and therefore could be subject to the same treatment; see [2–4] and the references therein. The fundamental property of (3) in our context is that it is a convex function of r_{ij} when $r_{ij} \geq 0$, which is clearly very useful in order to devise efficient solution approaches.

2.2 Feasibility of SFSP-DCR

While SFSP-DCR is clearly $\mathcal{NP}$-hard (it reduces to the Constrained Shortest Path problem, e.g., if $c_{ij} = \rho$ for all arcs), checking the existence of a feasible solution is easy. Indeed, according to (2)–(3), the delay is a decreasing function of the rates, which means that setting $r_{ij} = c_{ij}$ for each arc (i,j) provides the best (least) possible contribution to the delay.

Let us then define the set $C := \{\, c_{ij} \;:\; (i,j) \in A \,\}$ of all possible arc capacities (note that $|C| \leq m$), and for any $r \in C$, the reduced graph $G^r := (N, A^r)$ where $A^r := \{\, (i,j) \in A \;:\; c_{ij} \geq r \,\}$, i.e., all arcs whose residual capacity is smaller than r are removed. Let us now define the modified arc costs

$$\bar{l}_{ij} := \frac{L}{c_{ij}} + \frac{L}{w_{ij}} + l_{ij} + n_i \;\; ;$$

for future notational convenience, we will denote by $l'_{ij} := L/w_{ij} + l_{ij} + n_i$ the part of $\bar{l}_{ij}$ that does not depend on the choice of r. Solving an s-d shortest path on G^r thus allows one to compute the minimum-delay path P among the paths not containing arcs with capacity smaller than r, and therefore such that $\sigma/r_{min}(P) \leq \sigma/r$, where $r_{min}(P) := \min\{\, r_{ij} \;:\; (i,j) \in P\}$. Clearly, if the cost (delay, in our context) of P is $\leq \delta - \sigma/r$, then a feasible solution has been found. It is easy to show that, by repeating the above process for each $r \in C$ (hence $|C| \leq m$ times), one either finds a feasible solution or proves that none exists. Indeed, the only issue may come from the fact that the minimum-delay path P for some value of r may actually only use arcs with a larger capacity (hence assigned rate) than r: this means that $\sigma/r > \sigma/r_{min}(P)$, possibly leading to declaring P unfeasible while it actually satisfies the delay bound. However, in such a case P is also a path of $G^{\bar{r}}$ (and therefore it remains optimal) for some values $\bar{r} > r$ in C, the largest of which corresponds to $r_{min}(P)$; therefore, the delay of P is correctly evaluated during the iteration corresponding to $r_{min}(P)$.

By simply keeping track of the minimum cost among all feasible paths thusly generated (possibly avoiding to stop as soon as the first feasible path is found), this approach provides a first heuristic for SFSP-DCR. Since all arcs are reserved the maximum possible rate, this heuristic should not be expected to provide particularly good bound (and indeed this is shown to happen in Sect. 5.2); however it can quickly detect unfeasible instances. Furthermore, the heuristic can be improved somewhat using the ideas from the ERA case presented in next section.

3 The Equal Rate Allocation Case

Some polynomial time approaches to SFSP-DCR have been proposed in the literature [2,3] under two strong assumptions. The first one is the Equal Rate Allocation (ERA), i.e., that all the arcs (i, j) of the chosen s-d path P must receive the same resource allocation; therefore, $r_{ij} = r$ $(\geq \rho)$ for a given value r for all $(i, j) \in P$, while of course $r_{ij} = 0$ for $(i, j) \notin P$. Since throughout this section we shall consider the ERA assumption to be in force, we will always refer to "the rate r" as the unique value assigned to all r_{ij}, for $(i, j) \in P$, which of course implies that $r_{min}(P) = r$ as well; the corresponding restricted problem will be denoted as ERA-SFSP-DCR. The second assumption concerns the form of the objective function, as discussed in the following.

3.1 The Equal Costs Case

The ERA-SFSP-DCR problem can be solved in polynomial time if the objective function is nondecreasing with respect to the cardinality of P and the rate r; clearly, this is the case if we take $f_{ij} = 1$ for all $(i,j) \in A$, i.e., we pay the same cost for installing a unit of capacity on each arc, as this means that the objective function has form $r \cdot |P|$, where $|P|$ denotes the number of arcs in P. We will denote this problem by EC-ERA-SFSP-DCR (from "Equal Costs").

The crucial observation is that it is easy to solve EC-ERA-SFSP-DCR for a fixed value of r, as this basically is a hop-constrained shortest path problem. In fact, for a fixed value of r one can define the arc costs

$$l^r_{ij} := L/r + l'_{ij}$$

(cf. Sect. 2.2) and exploit the well-known property of the Bellman-Ford algorithm for the shortest path problem, i.e., that of being able to determine shortest paths with a constraint on the maximum number of hops. This is based on the fact that the Bellman-Ford algorithm works in $n-1$ phases; at the end of the h-th phase, the path currently entering a generic node i is the one having least cost among the paths (from s to i) with at most h arcs. Furthermore, the cost of the considered paths entering i is (obviously) nonincreasing as h grows. Hence, for the fixed value of r one can run the Bellman-Ford algorithm (with root s) on the reduced graph G^r (cf. Sect. 2.2) with the arc costs l^r and easily find the optimal solution to the EC-ERA-SFSP-DCR with the fixed value of r in $O(nm)$ time. This is done by simply checking the cost (that is delay in our context) of the s-d path entering d at the end of each phase: the first time this cost (delay) is $\leq \delta - \sigma/r$ one has found the hop-shortest delay-feasible path for the given value of r. Clearly, if the delay is always $> \delta - \sigma/r$, then no feasible delay-constrained path exists for the given value of r.

This approach has first been analyzed in [2] for the problem of finding the delay-minimal path under the ERA assumption. As in Sect. 2.2, this is done by repeating the above procedure for all values of r in the set C. Furthermore, in [2] it is observed that, by simply keeping track of the hop-shortest delay-feasible path found for each value $r \in C$ (which is freely obtained if the Bellman-Ford algorithm is used) and returning the best (in terms of minimum cardinality) computed path over the values r in C, an exact approach can be immediately derived for determining a feasible delay-constrained path P (if it exists) of minimum cardinality. Note that a simple way to enhance the practical efficiency of this approach is simply to order the values of C in an increasing way, and then applying the Bellman-Ford algorithm on G^r for increasing values of r: since the set A^r is non-increasing when r increases, while the path delays decrease, then the first time a feasible delay-constrained path is determined, this is indeed the hop-shortest delay-feasible path.

Another possibility to speed-up the approach, at the risk of not finding the optimal solution, is to rather use the standard general shortest path scheme [12] where the set Q of candidate nodes is a FIFO list (or queue). This SPT.L.Queue algorithm provides an efficient implementation of the Bellman-Ford algorithm that, despite having the same worst-case time complexity, is typically much faster in practice. Also, each node i is extracted from Q at most $n-1$ times (if there are no negative cost cycles, as it clearly is the case in our application), correspondingly to the phases of the original Bellman-Ford algorithm. Hence, one can just run the SPT.L.Queue algorithm and, each time d exits Q, check the corresponding path. Although by doing this one may fail to explore some of the hop-constrained shortest paths, our experiments showed that this happens very infrequently, while the algorithm is indeed significantly faster. Furthermore, it is easy to see that the modified algorithm surely finds a solution if the original one does: the shortest path corresponds to a feasible solution, if there is any, and both versions reliably find it. This is important because, as we shall see, the ERA approach is very effective for solving the general problem when it does find a solution, but often ERA-SFSP-DCR is empty while SFSP-DCR is not; using SPT.L.Queue cannot worsen this situation, as it does find a feasible solution if one exists. In fact, this variant has shown to be so much preferable in our test set that we will only report results about it.

The above analysis also suggests a similar modification to the feasibility-checking approach of Sect. 2.2: just use SPT.L.Queue to compute the shortest path and, whenever d exits Q, compute the cost and the delay of the current path, saving the best (minimum cost) one obtained. This way one explores several paths for each value of r, instead of just one, and starting with hop-short ones. Clearly, because the value of r_{ij} is not taken to be equal for all arcs, but rather set to its maximum possible value, the number of hops is no longer equivalent (for fixed r) to the objective function value, but one may still hope to generate "good" paths. We call this approach ERA-I (ERA-inspired); its distinctive feature is that it always produces a feasible solution, if one exists. Furthermore, it can be used to compute (at no added extra cost) the least possible feasible value of δ for which a feasible solution exists by just recording the smallest possible delay value generated; this will we useful in our computational experiments, as discussed in Sect. 5.1.

However, the approach above does not necessarily find an optimal solution to EC-ERA-SFSP-DCR when the more general objective function $r\,|P|$ has to be minimized. The obvious counterexample is the one where the computed minimal cardinality path P is such that the delay constraint is not tight: then, r can be suitably reduced by maintaining the path feasibility but without modifying the path cardinality, thus finding a better solution.

This has been addressed in [3], where the following simple modification to the above approach has been proposed. Again, an outer loop is performed where r is chosen in C, the reduced graph G^r built, and the Bellman-Ford shortest path procedure with root s and costs l^r is ran. For each possible path cardinality h, this determines a minimum-delay s-d path P among the paths in G^r having exactly h arcs. If such a path P is found to be feasible, then the algorithm first computes the minimum value of the rate such that the delay constraint related to P is satisfied as an equality: this is simply done by considering that

$$\Delta(r,P) := \frac{\sigma + L|P|}{r} + \sum_{(i,j)\in P} l'_{ij} \leq \delta \tag{4}$$

and noting again that the path delay is nondecreasing with respect to the rate, i.e., that for

$$\tilde{r}(P) := \frac{\sigma + L|P|}{\delta - \sum_{(i,j)\in P} l'_{ij}} \tag{5}$$

one has both $\Delta(\tilde{r}(P),P) = \delta$ and $\tilde{r}(P) \leq r$. Therefore, the algorithm minimizes the cost function, with respect to r, in the rate interval $[\tilde{r}(P), r]$; the approach is described in [3] for slightly more general cost functions, but in our case this simply amounts to picking the value $\tilde{r}(P)$. The following result holds true for the optimal solution (as stated, but not really proven, in [3]):

Proposition 3.1 *The best of the computed pairs* $(\tilde{r}(P),P)$ *is an optimal solution to EC-ERA-SFSP-DCR.*

Proof Consider the optimal solution (r^*, P^*) to the problem, and let r be the smallest element in C larger than (or equal to) r^*; clearly, such an r must exist since otherwise all arcs of P^* should be assigned a capacity strictly larger than the maximal arc capacity. Let us then consider the iteration of the approach where that particular r is chosen: clearly, P^* is a path in G^r (each arc $(i,j) \in P^*$ has capacity at least r^*, hence at least r) and it is delay-feasible for that value of r (since it is delay-feasible for $r^* \leq r$ and the delay decreases with r). Therefore, there exist delay-feasible s-d paths in G^r having exactly $h^* = |P^*|$ arcs. Now, let us consider the path P determined by the algorithm for the rate r and the hop count h^*: since P is the minimum-delay s-d path in G^r with h^* hops, its delay $\Delta(r, P)$ must be smaller than or equal to the delay $\Delta(r, P^*)$ of P^*. However, if by contradiction we had $\Delta(r, P) < \Delta(r, P^*)$, then, since $|P| = |P^*| = h^*$, one would also have

$$\sum_{(i,j)\in P} l'_{ij} < \sum_{(i,j)\in P^*} l'_{ij} \quad \Longrightarrow \quad \delta - \sum_{(i,j)\in P} l'_{ij} > \delta - \sum_{(i,j)\in P^*} l'_{ij}$$

(cf. (4)): the r-dependent term is in fact identical for P and P^*, and hence $\tilde{r}(P) < \tilde{r}(P^*)$. Since $\tilde{r}(P^*) \leq r^*$, then this would imply that $(\tilde{r}(P), P)$ is a better solution than (r^*, P^*). It follows that both P and P^* are optimal solutions, and therefore the best of the computed pairs $(\tilde{r}(P), P)$ is an optimal solution to EC-ERA-SFSP-DCR, as stated. □

Note that the solution (if any) is found in time $O(|C|nm) \leq O(nm^2)$; clearly it is feasible for the general SFSP-DCR, and therefore we can use this as a heuristic for the problem where the r_{ij} are allowed to take on different values (and, possibly, cost coefficients are not all equal). We will refer to this in the following as ERA-H.

3.2 The General Costs Case

An interesting remark, that does not seem to having been done yet in the literature, is that, under some conditions, it is possible to extend ERA-H to the case of non-identical arc reservation costs f_{ij}, thus considering objective functions of form $rf(P)$, where $f(P) := \sum_{(i,j)\in P} f_{ij}$.

In particular, assuming that f_{ij} are positive integers, one may replace the Bellman-Ford shortest path computation at each iteration of the ERA-H algorithm with a standard pseudo-polynomial Dynamic Programming approach to the Constrained Shortest Path problem, thus obtaining a pseudo-polynomial time algorithm (note that, under such a more general objective function, SFSP-DCR is $\mathcal{NP}$-hard, despite the ERA assumption). Specifically, this can be obtained by considering any valid upper bound $\bar{f}$ on the cost of a simple s-d path in G ($\bar{f} \leq (n-1)f_{max}$, where $f_{max} := \max\{\, f_{ij} \,:\, (i,j) \in A \,\}$) and generating the extended Directed Acyclic Graph $\widetilde{G}$ obtained from G by replicating each node i for $\bar{f}+1$ times, producing nodes (i,f) for all (integer) values $f \in \bar{F} := \{0, 1, \ldots, \bar{f}\}$; the (well-known) rationale of this definition is that (i,f) represents the fact that node i has been reached from s with a path of cost f. Each arc (i,j) in G is then replicated as well (at most) $\bar{f}+1$ times to join all nodes (i,f) with $(j, f+f_{ij})$, except of course those such that $f + f_{ij} > \bar{f}$; each of these arcs has the same delay coefficients and reservation capacity of the original arc (i,j). By the outlined transformation, it is easy to see that there is a one-to-one correspondence between the paths of G and these of $\widetilde{G}$ in terms of associated delay, hop count, reservable capacity and cost. It is well-known that, by basically visiting $\widetilde{G}$, in $O(\bar{f}m)$ time one can determine for all possible values $f \in \bar{F}$ the minimum-delay s-d path in G

among the s-d paths with objective function value exactly equal to f. This gives the following:

Theorem 3.1 *If f_{ij} are positive integers, then ERA-SFSP-DCR can be solved in pseudo-polynomial time $O(|C|\bar{f}m) \leq O(nm^2 f_{max})$.*

Proof We adapt ERA-H as follows: for each $r \in C$ we construct the subgraph of $\widetilde{G}$, say $\widetilde{G}^r$, containing only arcs with capacity $\geq r$ (thus still with size bounded by $O(\bar{f}m)$). We then perform a breadth-first visit of $\widetilde{G}^r$ from the node $(s, 0)$; each time a node (d, f) for some value of f is visited we have found, among all s-d paths of cost f, the minimum-delay one having the given number of hops. If that path P is delay-feasible we proceed, as in ERA-H, to find the smallest compatible value of r via (5) and we compare the cost $\tilde{r}(P)f = \tilde{r}(P)f(P)$ with that of the best solution found so far (if any), keeping the best.

One can easily prove that this approach finds the optimal solution of the problem by extending the arguments of the previous section. In particular, it is sufficient to consider the optimal solution (r^*, P^*) of the problem, its path cost $f^* = f(P^*)$ and hop count $h^* = |P^*|$, and the properly chosen $r \geq r^*$: because P^* belongs to the graph $\widetilde{G}^r$ and it has the given function value and hop count, there must be an iteration where node (d, f^*) is visited, providing a path P with $|P| = h^*$. Reasoning as in Sect. 3.1 one has that, if by contradiction we had $\Delta(r, P) < \Delta(r, P^*)$, this would imply $\tilde{r}(P) < \tilde{r}(P^*) \leq r^*$ (note that this goes through (4)–(5) and therefore crucially uses the fact that $|P^*| = |P|$, whence the need to perform a breadth-first visit), hence $\tilde{r}(P)f(P) = \tilde{r}(P)f^* < r^*f^* = r^*f(P^*)$ and the same conclusions stated in Sect. 3.1 follows. Note that the latter relation crucially requires $f(P) = f(P^*)$; in Sect. 3.1 this was actually the same as the condition

$|P^*| = |P|$, but here the two are different (and both needed), which justifies the need of the more involved pseudo-polynomial construction. □

As it often happens, a pseudo-polynomial time algorithm for the integer case can be used to construct a Fully-Polynomial Time Approximation Scheme (FPTAS) for the case where f_{ij} are not (necessarily) integer values.

Theorem 3.2 *If f_{ij} are positive, then ERA-SFSP-DCR admits a FPTAS with time complexity $O(n^2m^3/\varepsilon)$.*

Proof The approach requires the repeated application of the pseudo-polynomial time algorithm of Theorem 3.1 on a suitably defined approximated problem. The "outer loop" of the algorithm cycles over all values of $f \in F := \{ f_{ij} \ : \ (i,j) \in A \}$, i.e., all possible arc costs ($|F| \leq m$). For the currently selected f, one defines the reduced graph G_f where all arcs with cost strictly larger than f are deleted, and defines the scaled costs

$$\tilde{f}_{ij} = \lceil f_{ij}/K \rceil, \qquad \text{where} \qquad K := (\varepsilon f)/(n-1)$$

for all arcs in G_f. Since $f_{ij} \leq f$ for all arcs in G_f, $\tilde{f}_{ij} \leq \lceil n/\varepsilon \rceil$; hence, we can solve the reduced and scaled ERA-SFSP-DCR problem on G_f, with costs $\tilde{f}_{ij}$, by means of the pseudo-polynomial time algorithm of Theorem 3.1, in $O(n^2m^2/\varepsilon)$ time. After this is done for all values of $f \in F$, the minimum cost solution found is ε-optimal for the ERA-SFSP-DCR on the original graph and with the original (unscaled) f_{ij}.

This can be proven similarly to Theorem 3.1: consider the optimal solution (r^*, P^*) to the problem, its maximal arc cost $f_{max}(P^*) := \max\{ f_{ij} \ : \ (i,j) \in$

P^*}, its hop count $h^* := |P^*|$, and its *scaled path cost* $\tilde{f}^* := \tilde{f}(P^*)$. Now consider the outer iteration where $f = f_{max}(P^*)$. Clearly, P^* is a path in G_f, and $f \leq f(P^*)$ (since $f = f_{max}(P^*)$, and the costs are positive). Finally, consider the inner iteration (that must occur) with the appropriate $r \geq r^*$, where node $(d, \tilde{f}^*)$ is extracted from Q providing a path P with $|P| = h^*$. Because P is a minimum delay s-d path with (scaled) cost $\tilde{f}^*$ and hop count h^*, one has $\Delta(r, P) \leq \Delta(r, P^*)$ which, reasoning as in Theorem 3.1, gives $\tilde{r}(P) \leq r^*$. Furthermore, as a result of the rounding operation one has

$$f_{ij} \leq K\tilde{f}_{ij} \leq f_{ij} + K \quad ;$$

summing over P one obtains $f(P) \leq K\tilde{f}(P)$, while summing over P^* one obtains $K\tilde{f}(P^*) \leq f(P^*) + h^*K$. Now, using $\tilde{f}(P) = \tilde{f}^* = \tilde{f}(P^*)$ and the definition of K one obtains $f(P) \leq f(P^*) + h^*K \leq f(P^*) + \varepsilon f$, which using $f \leq f(P^*)$ can be rewritten as

$$f(P) \leq f(P^*)(1 + \varepsilon) \quad ,$$

i.e., P is ε-optimal considering the cost of the path alone. However, since we have already proven that $\tilde{r}(P) \leq r^*$, we can conclude that

$$\tilde{r}(P)f(P) \leq \tilde{r}(P)f(P^*)(1 + \varepsilon) \leq r^* f(P^*)(1 + \varepsilon),$$

i.e., P is ε-optimal by considering the ERA-SFSP-DCR objective function $rf(P)$. Since the objective function value of the best solution found by the outlined approach is less than or equal to $\tilde{r}(P)f(P)$, the thesis follows. The stated approximation result is thus obtained with the announced time complexity, since there are at most m outer iterations, each performed in $O(n^2m^2/\varepsilon)$ time. □

The tricky part of the approach is the selection of the scaling factor f, which must be on one hand "large enough" so that all scaled costs are "small" ($\leq n/\varepsilon$), and on the other hand "small enough" to ensure that $f \leq f(P^*)$; this is guaranteed by iterating over all the possible values of f, which are at most m, although in practice there may be better approaches. For instance, unless the set of arc costs is wildly distributed across a very large interval, just running the pseudo-polynomial time approach once with $f = f_{max}$ (hence $G_f = G$) looks to have pretty good chances to actually provide an ε-optimal solution right away. One may even be able to formally prove this by (approximately) solving the problem of computing the shortest feasible s-d path (in terms of the costs f_{ij}); reasoning as in Sect. 2.2, this can be cast as a standard Constrained Shortest Path problem and thus efficiently tackled by a FPTAS. If the obtained lower bound (by considering the approximation factor) is $\geq f_{max}$, then the single application of the pseudo-polynomial time algorithm is already guaranteed to produce ε-optimal solutions.

However, the approaches outlined before still assume the ERA restriction. Since evidence have been provided [4] (in the multi-flow case but with fixed path) that this can be highly suboptimal, in next section we discuss exact MINLP models of SFSP-DCR that can be used to compute optimal solutions to the more general scenario, and therefore assess the effectiveness of ERA-H when applied to the non-restricted case.

4 Second-Order Cone Models

We now proceed at presenting MISOCP models for the general version SFSP-DCR. For this, we first introduce arc-flow binary variables $x_{ij} \in \{0, 1\}$ indicating whether or not arc (i, j) belongs to the chosen path P, so that we can use the standard flow conservation constraints

$$\sum_{(j,i)\in BS(i)} x_{ji} - \sum_{(i,j)\in FS(i)} x_{ij} = \begin{cases} -1, & \text{if } i = s, \\ 1, & \text{if } i = d, \\ 0, & \text{otherwise} \end{cases} \qquad i \in N \tag{6}$$

to model the s-d–path requirements. We also introduce arc reserve variables r_{ij}, a single variable r_{min} (with obvious meaning) and the corresponding constraints

$$0 \le r_{ij} \le c_{ij} x_{ij} \qquad (i, j) \in A \tag{7}$$

$$\rho \le r_{min} \le r_{ij} + c_{max}(1 - x_{ij}) \qquad (i, j) \in A \tag{8}$$

that ensure on one hand that $r_{ij} = 0$ if $x_{ij} = 0$, and on the other hand that $\rho \le r_{min} \le r_{ij}$ if $x_{ij} = 1$. Note that the finiteness condition (1) is represented in (8), and $c_{max} := \max\{ c_{ij} : (i, j) \in A \}$ is used in (8) to ensure that any arc not in the chosen path ($x_{ij} = 0$) does not contribute to setting r_{min}; using c_{ij} in (8) would not be correct, as it would imply that $r_{min} \le \min\{ c_{ij} : (i, j) \in A \}$, even counting arcs not in the chosen path.

We then introduce θ_{ij} variables to represent the arc-additive part of the delay defined by (2)–(3); with these, the delay constraint can be modeled as

$$t + \sum_{(i,j)\in A} \left(\theta_{ij} + \Big(\frac{L}{w_{ij}} + l_{ij} + n_i \Big) x_{ij} \right) \le \delta \tag{9}$$

$$t\, r_{min} \ge \sigma \quad , \quad t \ge 0, \tag{10}$$

where t is an auxiliary variable needed to express the nonlinear σ / r_{min} term via the (rotated) SOCP constraint (10). The issue now is to represent the fact that θ_{ij} is zero if $x_{ij} = 0$, while it is given by an appropriate (convex) nonlinear expression otherwise.

We will first present the following big-M formulation for this fragment of the problem:

$$0 \leq \theta_{ij} \leq M x_{ij} \qquad (i,j) \in A \tag{11}$$

$$\theta_{ij} \geq s_{ij} - M(1 - x_{ij}) \qquad (i,j) \in A \tag{12}$$

$$s_{ij}\, r'_{ij} \geq L \qquad (i,j) \in A \tag{13}$$

$$s_{ij} \geq 0 \qquad (i,j) \in A \tag{14}$$

$$0 \leq r'_{ij} \leq r_{ij} + M(1 - x_{ij}) \qquad (i,j) \in A. \tag{15}$$

The formulation requires two extra sets of variables. Indeed, one would like to represent the nonlinear $\theta_{ij} \geq L/r_{ij}$ term via the (rotated) SOCP constraint

$$r_{ij}\, \theta_{ij} \geq L,$$

but this is not possible because, since $L > 0$, neither θ_{ij} nor r_{ij} are allowed to be zero, whereas $r_{ij} = \theta_{ij} = 0$ is expected when $x_{ij} = 0$. This is why one introduces:

– constraints (11) to guarantee that $x_{ij} = 0 \Longrightarrow \theta_{ij} = 0$, although these may be also avoided since the model has no incentive in increasing the value of θ_{ij};
– variables $s_{ij} \geq 0$ such that $\theta_{ij} \geq s_{ij}$ if $x_{ij} = 1$, while basically θ_{ij} and s_{ij} are "free" if $x_{ij} = 0$;
– variables $r'_{ij} \geq 0$ such that $r'_{ij} \leq r_{ij}$ if $x_{ij} = 1$, while basically r'_{ij} and r_{ij} are "free" if $x_{ij} = 0$;
– the SOCP constraint (13) ensuring that $s_{ij} \geq L/r'_{ij}$, which of course implies

$$\theta_{ij} \geq s_{ij} \geq L/r'_{ij} \geq L/r_{ij}$$

whenever $x_{ij} = 1$.

All this requires a "big-M" in the constraints, which we claim is best set as $M = \max(\sqrt{L}\,,\, L/\rho)$. The rationale for this choice is as follows:

– When $x_{ij} = 0$, (11)–(15) give

$$0 \geq \theta_{ij} \geq s_{ij} - M \qquad , \qquad s_{ij} \geq L/r'_{ij} \qquad , \qquad r'_{ij} \leq M.$$

(as $r_{ij} = 0$ as well). Since in this case s_{ij} and r'_{ij} can take any value (they do not appear in the objective function nor in any other constraint) we only need to choose a value of M for which a solution exists: this boils down to $M \geq s_{ij} \geq L/r'_{ij} \geq L/M$, hence $M^2 \geq L$.

– When $x_{ij} = 1$ instead, (11)–(15) give

$$M \geq \theta_{ij} \geq s_{ij} \geq L/r'_{ij} \geq L/r_{ij},$$

but $r_{ij} \geq \rho$ from (8), whence $M \geq L/\rho$.

Hence, SFSP-DCR can be modeled as a MISOCP, and therefore solved by off-the-shelf, efficient, general-purpose solvers like `Cplex` or `GUROBI`. However, the thusly proposed formulation has m binary variables and $4m+2$ continuous ones, together with $m+1$ SOCP constraints and, more importantly, several big-M coefficients. It can be expected that such a formulation may quickly become rather difficult to solve.

To avoid some of the issues in the previous formulation, we exploit a well-known reformulation technique known as "Perspective Reformulation", that has been introduced in [13] and used in several applications with success (e.g. [14–18]), although usually in a different form than the one that is presented here. The approach is based on the well-known fact (e.g. [19]) that, given any convex function

$f : \mathbb{R}^q \to \mathbb{R}$ and the two sets

$$\mathcal{P}_0 := \{\, 0 \,\} \qquad , \qquad \mathcal{P}_1 := \{\, v \in \mathbb{R}^q \,:\, l \leq v \leq u \,,\, f(v) \leq 0 \,\} \;\; ,$$

the best possible convex approximation of their (nonconvex) union is

$$conv(\, \mathcal{P}_0 \cup \mathcal{P}_1 \,) = \Big\{\, v \,:\, \lambda l \leq v \leq \lambda u \;,\; \lambda f(v/\lambda) \leq 0 \;,\; \lambda \in [0,1] \,\Big\} \;\; . \tag{16}$$

The above formulation looks ill-defined when $\lambda = 0$, but, as we will see, in practice this is not an issue. We can readily apply this to (3); in particular, we take $v = [\theta_{ij}, r_{ij}]$ and

$$f(\theta_{ij}, r_{ij}) := \frac{L}{r_{ij}} - \theta_{ij} \;\; ,$$

and identify $\lambda = x_{ij}$ to obtain that our requirement can be modeled by the MINLP fragment

$$\begin{aligned} &\rho x_{ij} \leq r_{ij} \leq c_{ij} x_{ij} \\ &0 \leq \theta_{ij} \leq (L/\rho) x_{ij} \\ &\frac{L x_{ij}^2}{r_{ij}} \leq \theta_{ij} \;\; . \end{aligned} \tag{17}$$

The crucial observation is that (17) can be directly modeled as a (rotated) SOCP constraint; thus,

$$\min \; \textstyle\sum_{(i,j) \in A} f_{ij} r_{ij} \tag{18}$$

$$(6) \;\; , \;\; (7) \;\; , \;\; (8) \;\; , \;\; (9) \;\; , \;\; (10)$$

$$\theta_{ij}\, r_{ij} \geq L x_{ij}^2 \qquad\qquad (i,j) \in A \tag{19}$$

$$\theta_{ij} \geq 0 \qquad\qquad (i,j) \in A \tag{20}$$

$$x_{ij} \in \{0,1\} \qquad\qquad (i,j) \in A \tag{21}$$

provides an exact reformulation of the problem. Indeed, the SOCP constraint (19) ensures that $\theta_{ij} \geq L/r_{ij}$ when $x_{ij} = 1$, but simply reduces to $\theta_{ij} \geq 0$ when $x_{ij} = 0$ (which will then mean that $\theta_{ij} = 0$ in any optimal solution since the model does not have any incentive to grow θ_{ij}), thus negating the need for the extra variables s_{ij} and r'_{ij} of the big-M formulation. Clearly, (18)–(21) is a more promising formulation than the one based on (11)–(15): while it has the same number of integer variables and conic constraints, it has only $2m + 2$ continuous variables, i.e., only $m + 1$ more than the structural ones, and clearly the minimum possible number to express the fractional terms in (2)–(3). Furthermore, the continuous relaxation of this formulation is likely to be significantly stronger, since the "optimal" reformulation of some (small) fragments of the model has been used; this has been already shown to yield significant performance improvements in other applications [13–16, 18], and the next section will show that the same holds here.

We finish this section by underlying a potential advantage of using MISOCP models: they could be easily generalized to the case where the cost comprises both reservation costs and fixed costs for the arc selection.

5 Computational Results

We now report our computational experiences aimed at assessing the relative efficiency and effectiveness of the different exact and heuristic approaches to SFSP-DCR. In particular, we compare ERA-H, ERA-I, and the two different MISOCP solvers for the solution of the general model SFSP-DCR. However, we confine ourselves to the case in which all capacity reservation costs f_{ij} are equal. This choice is partly motivated by the fact that, in such a scenario, ERA-H can be implemented simply and runs in (low) polynomial time $O(|C|nm)$, as shown in Sect. 3.1. However, another motivation is that defining sensible weights which measure the different impact of capacity consumption on different arcs is nontrivial, and in want of a specific need to do otherwise, assuming unitary weights is the reasonable option. Thus, while the experimental evaluation of the performances of the Dynamic Programming algorithm and the FPTAS presented in Sect. 3.2 could be interesting, it has not been carried out in this study; hopefully, it will be the subject of future investigation. We will refer to the model (18)–(21) as "P", and to the model using instead the constraints (11)–(15) as "bM". All the experiments have been performed on a (currently, rather low-end) PC with a 2Ghz Opteron 246 processor and 2Gb RAM, running a 64 bits Linux operating system (Ubuntu 12.4). All the codes were compiled with `gcc 4.4.3` and `-O3` optimizations. The two MISOCP models were solved by the two state-of-the-art, off-the-shelf, commercial solvers `Cplex 12.5` and `GUROBI 5.10`. Both solvers were ran without time limit and with default parameters.

5.1 The Instances

Constructing a set of significant DCR instances is a nontrivial exercise; fortunately, the recently released `FNSS` tool [20] provides a number of expert-tuned options to help devising realistic models of current telecommunications networks.

The generation process starts by selecting a network topology (basically, the graph G). For this, we considered two sets of real-world IP network topologies: the `GARR` subset [21] of the Internet Topology Zoo [22], and the `SNDlib` ones [23], which can be downloaded in `gml` format. Furthermore, in order to test our models on larger instances we used also random topologies generated according to the *Waxman model* [24]. This can be done directly by `FNSS`, which allows to generate random `Waxman` topologies simply by specifying the number of nodes n and the (probability) parameter $\alpha \in (0, 1]$, representing the link density: in our experiments we set $n \in \{100, 200\}$ and $\alpha = 0.4$.

Once the topology is loaded in `FNSS` (either by reading a `gml` file or by its internal random generator), one can assign realistic link capacities using one of the three allocation algorithms specifically designed for modeling PoP-level link capacity assignment in ISP backbones. These algorithms exploit the correlation between the amount of capacity assigned to a specific link and three metrics that are meant to capture the importance of the link; in particular, we used the *edge betweenness centrality* metric that corresponds to the number of shortest paths passing through a specific link. In particular, once one has specified a set of possible link capacity values w_{ij} (in our case the standard $\{1, 10, 40\}$ Gbps), the "edge betweeness" algorithm will assign a capacity from the set to all the links of the network proportionally to the edge betweenness centrality.

After this is done, `FNSS` also supports generation of realistic traffic matrices that take into account the capacities of the network. To generate a traffic matrix one needs to specify the mean traffic demand $\mu(T)$ and its standard deviation $\sigma^2(T)$; for our experiments we set $\mu(T) = 0.8$ Gbps and $\sigma^2(T) = 0.05$. We remark that `SNDlib` instances also provide link capacity and (multiple) traffic matrices, but for the sake of uniformity we used randomly-generated data on these, too.

Basically, the above set of parameters (together with arc costs) define an instance of a Multicommodity Min Cost Flow (MMCF) problem; in order to standardize and ease the distribution of our instances we thus created a corresponding set of MMCF instances in the well-known `Mnetgen` format [25]. We remark that `FNSS` generates by default n^2 traffic flows, i.e., one for each possible origin-destination pair in the network; while this results in an acceptable number of flows in all the real-world instances, the same cannot be said for the `Waxman` ones, that would in this way get the order of 10000 flows. Restricting the number of flows in `FNSS` is possible but complex; thus, we rather exploited this "translation stage" to select a subset of the `FNSS` generated flows, limiting the number to $n \log n$.

The last step of the generation process takes in input any MMCF instance and defines reasonable values for the missing parameters, basically the delay-related ones. For this, we implemented a `DCR-generator` that generates the remaining network parameters according to the suggestions of telecommunication network experts. In particular, the MTU L is set to 1500 bytes, since nearly all IP over Ethernet implementations use the Ethernet V2 frame format. Node delays n_i and link delays l_{ij} are then set equal to L/w_{ij}; individual reservation capacities c_{ij} are taken to be all equal to the mutual reservation capacitiy w_{ij} at this stage. Flow bursts σ are set to 3 times the MTU value. Finally, to define flow deadlines δ, we calculate the least possible value δ_{min}, under which no routing is possible, and the maximum possible value δ_{max}, over which the delay constraint becomes redundant. As mentioned in Section 2.2, δ_{min} can be computed using the ERA-I algorithm; as for δ_{max}, one can use an analogous approach where each r_{ij} is set to its minimum possible value ρ (as opposed to its maximum possible value c_{ij} as in ERA-I). Then, δ is randomly chosen uniformly within the interval $[\,\delta_{min}\,,\,(\delta_{max} - \delta_{min})\beta\,]$ for a fixed parameter β; in our experiments we used $\beta = 0.2$.

All the produced files are freely available at [25], and the `DCR-generator` will also be made available in due time. We remark that we tested, on a small subset of topologies, several other combinations of the generation parameters at the various steps (traffic matrices, delay, ...) but the general flavor of the results did not change significantly, so we believe that the ones reported in next section can be considered fairly typical.

5.2 Computational Experiments

In a first set of experiments, we assumed link speed w_{ij} and link capacity c_{ij} to coincide; in other words, each flow is individually routed in an "empty" network. Because of our generation process (cf. Sect. 5.1), this means that each corresponding instance is feasible.

A first set of results related to the performance of the heuristics ERA-I and ERA-H is reported in Table 1. For each instance of the three test sets (visually separated by an horizontal line) we report the size of the graph and the number of flows (k); each line of the table refers to the solution of all flows in the instance, one by one, as SFSP-DCR. For both heuristics, we report the average and the maximum (among all the flows of the instance) gap between the optimal value, as computed by the SOCP models, and the value of the solution returned by the heuristic. We do not report running times because, for both heuristics, they were negligible, always less than 0.001 seconds; furthermore, they will be reported later on (cf. Table 3). However, for ERA-H we report the failure rate (column "inf"), i.e., the fraction of the instances (flows) for which ERA-H was not able to find a feasible solution. We don't do this for ERA-I because, as the theory predicts, it was always capable of finding a feasible solution. Of course, the average and the maximum for ERA-H were only computed for those flows for which it did produce a feasible solution.

The table clearly depicts a rather awkward picture, whereby ERA-I always produces solutions of rather abysmal quality (average gaps almost always larger than 50% and maximal gaps on the region of 90%) but solves all flows, whereas ERA-H consistently produces solutions of extremely good quality in average: as the "max" column shows gaps can be quite large at times, but the very low average means that this occurs very rarely. However, ERA-H fails to find solutions in a significant fraction of the cases (up to 85%), despite all instances being guaranteed to be feasible.

We then move on to Table 2, which reports the behavior of the two general-purpose solvers for the solution of the two MISOCP models P and bM. Since we did not set any time limit all solvers were capable of solving all instances, so we only report the (average and maximum) running time ("t") and the number of nodes ("n") they required. We do not report again instance information since the rows are organized exactly as in Table 1 and have the same meaning.

The table clearly shows that—how it should be expected—model P is much better than model bM. On the real-world networks, the first is between 2 and 6 times faster on average for `Cplex` and between 3 and 12 times faster on average for `GUROBI`, with similar (albeit often somewhat smaller) improvement rates on the maximum time. For the largest networks, the ratios climb to a factor of 10 and 15 for the average and to a factor of 20 and 35 for the maximum, respectively for `Cplex` and `GUROBI`. Hence, there is no reason not to use model P. The comparison between the two solvers is less clear: `GUROBI` is often somewhat faster, but also somewhat less consistent (although `Cplex` have occasionally shown numerical issues). Incidentally, these results probably depend on somewhat different strategies, as shown by the fact that `GUROBI` enumerates significantly more nodes, but it is often faster in doing so, which probably implies it being less reliant on strategies to improve the lower bound, such as valid inequalities; indeed, this is the typical approach that the folklore would associate to a faster behavior on "easy" instances but a less consistent one on"harder" ones. Yet, the two solvers are largely equivalent, and the results bode relatively well for the use of the P model in a real-world operating environment, with average and even maximum (except for a few cases for `GUROBI`) running times in the split-second range. However, as the size grows average (and especially maximum) running times become unfeasibly large. Admittedly, one could experiment with setting a tight time limit and/or a coarser optimality tolerance to the MISOCP solvers to determine whether or not good feasible solutions can be obtained (although not proven optimal) in much less time; however, it is fair to say that these results already start to show the limitations of an approach entirely relying on general-purpose tools.

Given these results, for our final set of experiments we focused only on the P model. The rather peculiar behavior of the ERA-H heuristic, which is very effective when it does deliver a solution, but also rather prone to failure, suggests to try to combine the best characteristics of all the available approaches. One simple way to do that is to develop a three-pronged approach ("3P" in the following) that proceeds as follows:

1. initially it runs the very quick ERA-I, and if the instance is found to be unfeasible it terminates;

2. otherwise it runs ERA-H: if a solution is found it is reported and the approach terminates;
3. if all else fails, then model P is ran and its solution is reported.

This is clearly not the most sophisticated approach: one could for instance choose to always run at least the root node of the P model to try to determine whether the current instance is one of the (very) few where ERA-H finds solution of bad quality, or more in general run the MISOCP solvers on tight time limits giving them the ERA-H solution as cutoff. However, we decided to limit ourselves to the simplest solution and test it on a somewhat more "realistic" environment. In particular, we fixed in four possible ways (0, 0.2, 0.4, 0.8) a maximum level γ of arc load, and for each level we subtracted to the arc capacity an amount uniformly drawn at random in $[0, \gamma w_{ij}]$ to simulate a more realistically loaded network. We then compared three approaches in all these four scenarios: ERA-H, the use of the MISOCP solvers (obviously with model P), and the 3P approach. The results are shown in Tables 3, 4, 5, and 6 for $\gamma = 0$ (i.e., the "unloaded" network of Tables 1 and 2), $\gamma = 0.2$, $\gamma = 0.4$ and $\gamma = 0.8$, respectively. The rows of the tables are all organized in the same way as the previous ones. In the leftmost part of each table we report the (average and maximum) running times of the 3P approach, with both solvers, as compared to that of the direct MISOCP approach. In the middle part we report the (average and maximum) gap of 3P, which is of course the same for the two solvers, since that of the MISOCP is always zero. Finally, in the leftmost part we report the average (when it is larger than `1e-6` seconds) and maximum running time of ERA-H, as well as the corresponding fraction of "failed" instances. This is just the number of flows for which a solution was not found when $\gamma = 0$, but for larger values of γ some of the instances actually do not have a solution; thus, in this case we report the fraction of the feasible instances (for which MISOCP and 3P can find a solution) that cannot be solved by ERA-H. Note that in one case (entry "***" in Table 6) not a single flow was feasible, and therefore this fraction had no meaning. Moreover, note that we do not report gaps for ERA-H since we can estimate them to be very close to these of 3P; the latter are bound to be slightly smaller, because 3P solves more instances than ERA-H and these in the difference set are solved with guaranteed zero gap, but the difference is negligible.

The results show that, for $\gamma = 0$, 3P is not much faster than the MISOCP on the `GARR` instances; this is not surprising, because the failure rate of ERA-H in these is very large, meaning that for more than 75% of the flows one actually ends up performing both approaches. However, on the same instances 3P is significantly faster than P for $\gamma > 0$: this is due to the fact that the percentage of unfeasible instances increases with γ, and these are quickly identified by ERA-I without a need to invoke neither of the other two components (although, infeasible instances are quickly identified by the general-purpose solvers as well, as it is easy to see since their running time also decreases).

On the `SNDlib` instances and on the Waxman-100 one, 3P most often requires a substantially smaller average running time than MISOCP (typically one order of magnitude less), while obtaining a very low average gap (less than 1%) in spite of the occasionally substantial (but, clearly, very rare) maximum gaps. For the `SNDlib` instances, the running time of ERA-H is significantly smaller; however, the heuristic fails in a significant number of cases. Furthermore, while for $\gamma = 0$ ERA-H is still two orders of magnitude faster on the Waxman-100 instance, when

$\gamma > 0$ the difference is much smaller. This should be expected in view of the fact that its running time depends on $|C|$, and while for $\gamma = 0$ we have $|C| = 3$ in our instances, in the other (more realistic) cases $|C| \approx m$. We remark that these results are strongly dependent on using the SPT.L.Queue approach in implementing ERA-H (and therefore 3P), since otherwise the approaches would be significantly less competitive: using the Bellman-Ford algorithm typically results in increased running times by two orders of magnitude, especially as the load of the network increases.

The effect of network load is even more apparent in the Waxman-200 instance: indeed, while for $\gamma = 0$ ERA-H requires about 0.01 seconds, for $\gamma > 0$ its average running time is about 8 seconds, and the maximum about 10. For `GUROBI` this is actually larger than the mean running time, so that 3P turns out to be actually slower than P on average, although it is still significantly faster when the maximum is taken into account; things are different with `Cplex` only because for this instance it is significantly slower than `GUROBI`. Yet, all this is scarcely relevant: none of the proposed techniques can solve SFSP-DCR instances of that size efficiently enough.

6 Conclusions

Routing under QoS constraints is a new, interesting application that motivates the development of MINLP models with novel structures. In particular, the SFSP-DCR problem is an interesting optimization model that shows both a “classical” flow/path structure and a pretty uncommon nonlinear (albeit, fortunately, convex) resource constraint. This peculiar combination allows for the development of specialized approaches, largely based on shortest paths computations, for the case where the "nonlinear" features of the problem can be dealt with easily, such as when one restricts all the resource allocations to be equal; however, the general case gives rise to complex MISOCP models that require sophisticated reformulation techniques to be solved efficiently enough with general-purpose tools.

Our computational results show that one can solve SFSP-DCR with high efficiency for networks of realistic size, in particular if it is possible to cope with occasional (but very rare) suboptimal solutions; in this case, the "three pronged" approach that combines combinatorial heuristics and the use of MISOCP models seems to be a promising option. Let us mention that split-second running times on ordinary hardware is feasible for practical applications, because routing decisions can nowadays be demanded to a specialized Path Computation Element (PCE) [26] that, unlike ordinary routers, can be computationally powerful and run a significant amount of non-routing-related software such as a general-purpose optimization solver. Besides, only one PCE per network is required, thus hardware, software and maintenance costs would not be a serious issue. Thus, the approaches presented in the paper could, at least in principle, be feasibly implemented in a real-world operating environments.

However, our results also show that there is still ample room for improvement. When the size of the network increases, all the approaches become excessively slow. This is true not only for the MISOCP models, but also for the (otherwise very fast) combinatorial heuristics, even in its best case of all-equal costs; while efficient (approximated) versions could be devised for general costs, it must be expected that their practical performances be significantly slower than these for the all-equal case. Hence, we believe that the study of nonlinearly-constrained shortest path (or flow) models is a promising new research venue that can both lead to significant methodological advances and foster practically useful applications.

Acknowledgements

We are very grateful to Giovanni Stea for numerous suggestions and helpful discussions, and to Lorenzo Saino for his precious assistance in using `FNSS`. This research has been partly funded by the Italian Ministry of Education, University and Research (MIUR) under grant PRIN 2009XN4ZRR.

References

1. Saad, M., Leon-Garcia, A., Yu, W. Optimal network rate allocation under end-to-end quality-of-service requirements. IEEE Transactions on Network and Service Management **4**(3), 40–49 (2007)
2. Ma, Q., Steenkiste, P. Quality-of-service routing for traffic with performance guarantees. In Proc. IFIP International Workshop on Quality of Service, 115–126 (1997)
3. Orda, A. Routing with end-to-end QoS guarantees in broadband networks. IEEE/ACM Trans. on Networking **7**(3), 365–374 (1999)
4. Lori, A., Stea, G., Vaglini, G. Towards Resource-Optimal Routing Plans for Real-Time Traffic. In Leveraging Applications of Formal Methods, Verification, and Validation, Margaria, T. and Steffen, B., editors, volume 6415 of Lecture Notes in Computer Science, 214–227 (2010)
5. Di Puglia Pugliese, L., Guerriero, F. A survey of resource constrained shortest path problems: Exact solution approaches. Networks **62**(3), 183–200 (2013)
6. Irnich, S. Resource extension functions: Properties, inversion and generalization to segments. OR Spectrum **30**, 113–148 (2008)
7. Irnich, S., Desaulniers, G. Shortest Path Problems with Resource Constraints. In Desaulniers, G., Desrosiers, J., Solomon, M. (eds): Column Generation, Springer, 33–65 (2005)

8. Lenzini, L., Mingozzi, E., Stea, G. A methodology for computing end-to-end delay bounds in FIFO-multiplexing tandems. Performance Evaluation **65**, 922–943 (2008)
9. Lenzini, L., Martorini, L., Mingozzi, E., Stea, G. Tight end-to-end per-flow delay bounds in FIFO multiplexing sink-tree networks. Performance Evaluation **63**, 956–987 (2006)
10. Lenzini, L., Mingozzi, E., Stea, G. Eligibility-based round robin for fair and efficient packet scheduling in interconnection networks. IEEE Transactions on Parallel and Distributed Systems **15**(3), 254–266 (2004)
11. Zhang, L. Virtual clock: a new traffic control algorithm for packet switching networks. ACM SIGCOMM Computer Communication Review **20**(4), 19–29 (1990)
12. Gallo, G., Pallottino, S. Shortest path methods: A unifying approach. Mathematical Programming Studies **26**, 38–64 (1986)
13. Frangioni, A., Gentile, C. Perspective cuts for a class of convex 0–1 mixed integer programs. Mathematical Programming **106**(2), 225–236 (2006)
14. Frangioni, A., Gentile, C. A computational comparison of reformulations of the perspective relaxation: SOCP vs. cutting planes. Operations Research Letters **37**(3), 206–210 (2009)
15. Frangioni, A., Gentile, C., Lacalandra, F. Tighter approximated MILP formulations for unit commitment problems. IEEE Transactions on Power Systems **24**(1), 105–113 (2009)
16. Frangioni, A., Gentile, C., Grande, E., Pacifici, A. Projected perspective reformulations with applications in design problems. Operations Research **59**(5), 1225–1232 (2010)
17. Günlük, O., Linderoth, J. Perspective Reformulation and Applications. In Lee, J., Leyffer, S. (eds): Mixed Integer Nonlinear Programming, volume 154 of The IMA Volumes in Mathematics and its Applications, 61–89 (2012)
18. Hijazi, H., Bonami, P., Cornuejols, G., Ouorou, A. Mixed integer nonlinear programs featuring “On/Off” constraints: Convex analysis and applications. Electronic Notes in Discrete Mathematics **36**(1), 1153–1160 (2010)
19. Tawarmalani, M., Sahinidis, N. Convex extensions and envelopes of lower semi-continuous functions. Mathematical Programming **93**, 515–532 (2002)
20. Saino, L., Cocora, C., Pavlou, G. A toolchain for simplifying network simulation setup. In Proceedings of the 6th International ICST Conference on Simulation Tools and Techniques, SIMUTOOLS ’13 (ICST, ICST, Brussels, Belgium (2013)
21. Garr. `http://www.garr.it`

22. The internet topology zoo. `http://www.topology-zoo.org/`

23. Sndlib. `http://www.sndlib.zib.de`

24. Waxman, B. Routing of multipoint connections. IEEE Journal on Selected Areas in Communications **6**(9), 1617–1622 (1988)

25. Multicommodity problems. `http://www.di.unipi.it/optimize/Data/MMCF.html`

26. Paolucci, F., Cugini, F., Giorgetti, A., Sambo, N., Castoldi, P. A survey on the path computation element (PCE) architecture. IEEE Communications Surveys Tutorials, **to appear** (2013)

				ERA-I		ERA-H		
instance	n	m	k	avg	max	avg	max	inf
garr 1999-01	16	36	240	0.65	0.88	0.000	0.001	0.02
garr 1999-04	23	50	506	0.57	0.94	0.000	0.001	0.75
garr 1999-05	23	50	506	0.55	0.94	0.000	0.000	0.75
garr 2001-09	22	48	462	0.60	0.94	0.000	0.000	0.74
garr 2001-12	24	52	552	0.59	0.94	0.000	0.000	0.75
garr 2004-04	22	48	462	0.56	0.94	0.000	0.000	0.75
garr 2009-08	54	136	2862	0.65	0.94	0.001	0.386	0.85
garr 2009-09	55	138	2970	0.67	0.94	0.000	0.000	0.85
garr 2009-12	54	136	2862	0.67	0.94	0.001	0.240	0.85
garr 2010-01	54	136	2862	0.67	0.94	0.001	0.241	0.85
abilene	12	15	31	0.52	0.92	0.000	0.000	0.06
atlanta	15	22	45	0.57	0.88	0.000	0.000	0.07
cost266	37	57	120	0.48	0.95	0.000	0.000	0.17
dfn-bwin	10	45	45	0.03	0.06	0.000	0.000	0.00
dfn-gwin	11	47	53	0.16	0.86	0.000	0.000	0.02
di-yuan	11	42	58	0.48	0.90	0.000	0.000	0.12
france	25	45	66	0.44	0.90	0.000	0.000	0.02
geant	22	36	63	0.46	0.89	0.000	0.001	0.06
germany50	50	88	276	0.50	0.90	0.000	0.001	0.21
giul39	39	172	1482	0.67	0.97	0.011	0.570	0.10
india35	35	80	195	0.53	0.93	0.000	0.000	0.11
janos-us	26	84	650	0.71	0.95	0.004	0.275	0.18
janos-us-ca	39	122	1482	0.68	0.95	0.010	0.289	0.23
newyork	16	49	89	0.50	0.90	0.000	0.000	0.03
nobel-eu	28	41	106	0.55	0.93	0.000	0.000	0.23
nobel-ger	17	26	51	0.49	0.93	0.000	0.000	0.10
nobel-us	14	21	24	0.35	0.90	0.000	0.001	0.00
norway	27	51	341	0.71	0.94	0.000	0.000	0.12
pdh	11	34	54	0.64	0.90	0.000	0.001	0.04
pioro40	40	89	204	0.40	0.89	0.000	0.000	0.25
polska	12	18	24	0.59	0.90	0.000	0.000	0.00
sun	27	102	702	0.76	0.95	0.008	0.431	0.06
ta2	65	108	388	0.45	0.92	0.000	0.000	0.31
w1-100-04	100	414	664	0.77	0.95	0.015	0.739	0.07
w1-200-04	200	1550	1528	0.71	0.96	0.015	0.814	0.05

Table 1 Behavior of ERA-I and ERA-H

`Cplex` P				`Cplex` bM				`GUROBI` P				`GUROBI` bM			
average		maximum		average		maximum		average		maximum		average		maximum	
t	n	t	n	t	n	t	n	t	n	t	n	t	n	t	n
0.022	0.017	0.13	1	0.09	0.21	0.33	1	0.034	0.5	0.09	9	0.096	6.6	0.38	17
0.029	0.000	0.07	0	0.10	0.07	0.45	3	0.016	1.9	0.11	26	0.115	2.7	0.55	35
0.029	0.004	0.09	1	0.10	0.08	0.40	3	0.018	2.0	0.08	25	0.139	3.5	0.79	36
0.030	0.000	0.10	0	0.11	0.10	0.44	3	0.020	2.0	0.09	19	0.156	4.0	0.97	29
0.029	0.000	0.08	0	0.09	0.16	0.32	3	0.015	0.0	0.04	0	0.116	0.1	0.31	17
0.028	0.000	0.18	0	0.09	0.05	0.31	3	0.021	3.0	0.06	14	0.128	3.5	0.57	27
0.087	0.005	0.46	2	0.57	0.47	1.99	27	0.070	7.6	0.72	124	0.776	18.8	5.39	164
0.089	0.011	0.62	4	0.60	0.61	2.19	36	0.071	7.6	0.59	202	0.918	21.8	4.85	212
0.090	0.013	0.78	4	0.60	0.59	2.47	44	0.071	7.6	0.55	123	0.920	22.7	6.21	352
0.093	0.013	0.50	4	0.61	0.57	2.32	32	0.073	7.6	0.68	114	0.916	22.8	5.76	339
0.011	0.000	0.03	0	0.02	0.03	0.09	1	0.011	0.0	0.03	0	0.032	0.1	0.06	3
0.015	0.044	0.18	1	0.03	0.07	0.17	1	0.012	0.5	0.03	8	0.044	1.6	0.08	15
0.015	0.017	0.06	1	0.05	0.03	0.26	1	0.012	0.4	0.05	11	0.099	0.8	0.30	27
0.012	0.000	0.03	0	0.05	0.02	0.11	1	0.007	0.0	0.01	0	0.068	0.0	0.08	0
0.020	0.151	0.10	1	0.05	0.00	0.16	0	0.017	0.0	0.04	0	0.104	0.1	0.31	4
0.051	1.190	0.34	18	0.11	1.36	0.62	31	0.028	2.0	0.21	46	0.116	4.9	0.46	74
0.014	0.000	0.05	0	0.04	0.02	0.16	1	0.011	0.3	0.03	6	0.079	1.2	0.18	17
0.011	0.016	0.06	1	0.03	0.03	0.19	1	0.011	0.7	0.04	11	0.062	1.2	0.17	22
0.024	0.025	0.10	1	0.09	0.06	0.70	1	0.016	1.1	0.26	34	0.166	2.5	0.93	52
0.245	0.547	0.99	13	1.27	15.33	6.68	610	0.424	67.6	6.69	1308	1.795	138.5	30.02	2212
0.021	0.036	0.27	1	0.08	0.07	0.58	4	0.014	0.4	0.12	14	0.132	1.8	0.34	29
0.093	0.108	0.63	7	0.43	2.65	1.55	30	0.150	21.2	2.14	767	0.717	85.4	16.54	1168
0.141	0.138	0.83	8	0.80	5.76	2.76	243	0.285	47.1	7.87	916	1.741	158.4	25.93	1595
0.018	0.034	0.14	1	0.07	0.05	0.28	1	0.013	0.8	0.04	14	0.091	2.2	0.22	22
0.016	0.009	0.08	1	0.04	0.05	0.26	1	0.013	0.2	0.09	9	0.080	0.4	0.25	31
0.011	0.020	0.04	1	0.04	0.08	0.24	3	0.012	0.4	0.04	11	0.056	1.4	0.33	38
0.015	0.083	0.10	1	0.04	0.04	0.19	1	0.012	0.8	0.05	11	0.047	0.9	0.15	11
0.035	0.079	0.32	8	0.11	0.36	0.96	8	0.033	2.8	0.44	30	0.141	7.7	0.63	55
0.042	0.444	0.38	8	0.11	0.74	0.38	13	0.023	4.6	0.09	47	0.081	7.1	0.23	45
0.019	0.039	0.27	1	0.10	0.14	0.57	6	0.015	0.6	0.09	13	0.160	2.6	0.57	44
0.020	0.042	0.11	1	0.03	0.08	0.09	1	0.010	0.5	0.03	7	0.038	1.2	0.06	9
0.165	0.587	0.89	13	0.65	7.68	2.36	257	0.189	39.6	0.76	282	0.961	126.9	5.68	583
0.020	0.015	0.13	1	0.12	0.08	0.89	4	0.018	0.6	0.12	27	0.214	1.9	1.52	33
1.854	3.176	43.14	85	8.88	164.49	43.87	2585	2.372	159.3	7.09	703	14.064	407.2	110.36	5339
24.231	25.366	413.95	4075	231.09	2714.68	9088.54	127429	9.575	241.4	63.37	1395	134.145	637.0	2384.84	10943

Table 2 Behavior of MISOCP models

Cplex				GUROBI								
SOCP		3P		SOCP		3P		Gaps		ERA-H		
avg	max	avg	max	avg	max	avg	max	avg	max	avg	max	inf
0.025	0.12	0.001	0.03	0.035	0.10	0.001	0.03	0.00	0.00	4e-5	0.01	0.02
0.030	0.08	0.022	0.06	0.017	0.12	0.016	0.10	0.00	0.00	4e-5	0.01	0.75
0.028	0.08	0.021	0.06	0.018	0.08	0.016	0.08	0.00	0.00	6e-5	0.01	0.75
0.026	0.09	0.021	0.08	0.022	0.09	0.018	0.09	0.00	0.00	4e-5	0.01	0.74
0.027	0.07	0.022	0.07	0.016	0.04	0.012	0.04	0.00	0.00	4e-5	0.01	0.75
0.026	0.17	0.020	0.05	0.022	0.06	0.019	0.06	0.00	0.00	4e-5	0.01	0.75
0.084	0.44	0.075	0.44	0.069	0.70	0.065	0.71	0.00	0.39	2e-4	0.01	0.85
0.086	0.62	0.078	0.62	0.069	0.56	0.063	0.57	0.00	0.00	2e-4	0.01	0.85
0.088	0.75	0.078	0.73	0.071	0.52	0.061	0.50	0.00	0.24	2e-4	0.01	0.85
0.087	0.46	0.076	0.45	0.074	0.61	0.066	0.59	0.00	0.24	2e-4	0.01	0.85
0.009	0.02	0.001	0.01	0.009	0.02	0.001	0.01	0.00	0.00		0.00	0.06
0.016	0.16	0.001	0.02	0.010	0.03	0.001	0.02	0.00	0.00		0.00	0.07
0.013	0.05	0.002	0.03	0.012	0.04	0.003	0.04	0.00	0.00		0.00	0.17
0.011	0.02	0.000	0.00	0.007	0.01	0.000	0.01	0.00	0.00		0.00	0.00
0.019	0.09	0.000	0.01	0.015	0.04	0.000	0.01	0.00	0.00		0.00	0.02
0.050	0.35	0.017	0.35	0.028	0.22	0.012	0.23	0.00	0.00		0.00	0.12
0.015	0.04	0.000	0.01	0.010	0.03	0.000	0.01	0.00	0.00		0.00	0.02
0.013	0.05	0.001	0.01	0.010	0.04	0.001	0.03	0.00	0.00		0.00	0.06
0.021	0.09	0.005	0.08	0.017	0.24	0.007	0.27	0.00	0.00	7e-5	0.01	0.21
0.254	1.01	0.019	0.66	0.449	7.57	0.087	6.52	0.01	0.57	3e-4	0.01	0.10
0.019	0.25	0.002	0.04	0.016	0.11	0.002	0.07	0.00	0.00		0.00	0.11
0.091	0.62	0.013	0.33	0.153	2.25	0.051	2.19	0.00	0.28	1e-4	0.01	0.18
0.144	0.84	0.026	0.49	0.298	9.59	0.118	7.70	0.01	0.29	2e-4	0.01	0.23
0.017	0.13	0.000	0.02	0.015	0.04	0.001	0.02	0.00	0.00		0.00	0.03
0.014	0.05	0.004	0.05	0.016	0.09	0.005	0.09	0.00	0.00		0.00	0.23
0.010	0.03	0.002	0.03	0.015	0.04	0.002	0.04	0.00	0.00		0.00	0.10
0.013	0.09	0.000	0.00	0.014	0.05	0.000	0.00	0.00	0.00		0.00	0.00
0.032	0.30	0.005	0.25	0.035	0.32	0.005	0.13	0.00	0.00	6e-5	0.01	0.12
0.034	0.30	0.001	0.02	0.026	0.10	0.002	0.10	0.00	0.00		0.00	0.04
0.019	0.27	0.007	0.25	0.018	0.09	0.007	0.09	0.00	0.00	5e-5	0.01	0.25
0.016	0.09	0.000	0.00	0.014	0.03	0.000	0.00	0.00	0.00		0.00	0.00
0.154	0.89	0.006	0.36	0.188	0.87	0.009	0.40	0.01	0.43	2e-4	0.01	0.06
0.019	0.12	0.008	0.05	0.020	0.13	0.009	0.13	0.00	0.00	8e-5	0.01	0.31
1.906	46.7	0.034	1.84	2.354	8.35	0.150	3.54	0.01	0.74	2e-3	0.01	0.07
23.660	357.7	0.247	54.29	9.033	63.19	0.399	12.36	0.01	0.81	1e-2	0.02	0.05

Table 3 Comparison of the P model and 3P for $\gamma = 0$

Cplex				GUROBI								
SOCP		3P		SOCP		3P		Gaps		ERA-H		
0.024	0.11	0.001	0.05	0.032	0.11	0.001	0.03	0.00	0.00	3e-4	0.01	0.05
0.040	0.12	0.003	0.05	0.003	0.09	0.003	0.07	0.00	0.00	5e-4	0.01	0.79
0.037	0.12	0.004	0.05	0.004	0.05	0.003	0.04	0.00	0.00	5e-4	0.01	0.82
0.046	0.15	0.004	0.08	0.005	0.07	0.003	0.06	0.00	0.00	4e-4	0.01	0.73
0.035	0.12	0.004	0.06	0.003	0.04	0.003	0.03	0.00	0.00	5e-4	0.01	0.76
0.035	0.11	0.003	0.05	0.003	0.04	0.002	0.04	0.00	0.00	4e-4	0.01	0.73
0.132	0.89	0.033	0.29	0.024	0.31	0.027	0.34	0.00	0.00	7e-3	0.02	0.74
0.134	0.96	0.035	0.37	0.025	0.36	0.029	0.37	0.00	0.00	7e-3	0.02	0.76
0.129	0.76	0.035	0.51	0.026	0.33	0.028	0.34	0.00	0.24	7e-3	0.02	0.76
0.131	0.80	0.036	0.51	0.026	0.30	0.031	0.33	0.00	0.24	7e-3	0.02	0.76
0.010	0.04	0.000	0.01	0.005	0.02	0.001	0.02	0.00	0.00		0.00	0.04
0.015	0.10	0.001	0.02	0.009	0.04	0.001	0.03	0.00	0.00		0.00	0.05
0.014	0.06	0.002	0.04	0.010	0.06	0.002	0.06	0.00	0.00	3e-4	0.01	0.10
0.021	0.05	0.000	0.00	0.001	0.01	0.001	0.01	0.00	0.00	2e-4	0.01	0.00
0.032	0.08	0.001	0.02	0.011	0.03	0.001	0.02	0.00	0.00	2e-4	0.01	0.05
0.044	0.19	0.011	0.18	0.026	0.20	0.012	0.21	0.00	0.00	2e-4	0.01	0.15
0.019	0.06	0.001	0.01	0.008	0.03	0.000	0.01	0.00	0.00	3e-4	0.01	0.00
0.014	0.04	0.000	0.01	0.007	0.04	0.001	0.01	0.00	0.00		0.00	0.02
0.025	0.12	0.004	0.12	0.013	0.09	0.005	0.10	0.00	0.00	1e-3	0.01	0.13
0.257	1.21	0.057	1.02	0.424	7.08	0.100	7.07	0.01	0.57	2e-2	0.03	0.11
0.025	0.20	0.002	0.05	0.015	0.11	0.004	0.04	0.00	0.00	1e-3	0.01	0.09
0.103	0.50	0.018	0.33	0.155	1.84	0.041	1.84	0.00	0.28	2e-3	0.01	0.16
0.170	0.78	0.044	0.81	0.274	3.34	0.113	3.30	0.01	0.26	6e-3	0.02	0.22
0.020	0.10	0.001	0.06	0.014	0.05	0.002	0.03	0.00	0.00	4e-4	0.01	0.03
0.015	0.06	0.003	0.03	0.014	0.07	0.004	0.07	0.00	0.00	2e-4	0.01	0.17
0.013	0.04	0.000	0.01	0.011	0.04	0.001	0.02	0.00	0.00		0.00	0.03
0.013	0.07	0.001	0.02	0.007	0.03	0.001	0.02	0.00	0.00		0.00	0.08
0.032	0.26	0.006	0.27	0.034	0.27	0.008	0.26	0.00	0.00	7e-4	0.01	0.12
0.034	0.17	0.001	0.03	0.023	0.07	0.003	0.08	0.00	0.00		0.00	0.04
0.020	0.09	0.003	0.08	0.013	0.06	0.004	0.07	0.00	0.00	1e-3	0.01	0.18
0.017	0.08	0.001	0.02	0.013	0.04	0.002	0.04	0.00	0.00		0.00	0.05
0.154	0.82	0.013	0.42	0.187	1.45	0.020	0.57	0.00	0.23	4e-3	0.01	0.08
0.025	0.11	0.007	0.11	0.013	0.13	0.008	0.13	0.00	0.00	2e-3	0.01	0.25
1.48	46.0	0.42	3.5	2.286	10.51	0.52	3.62	0.01	0.65	0.17	0.26	0.09
31.38	291.1	16.66	208.5	9.772	97.03	16.50	33.57	0.01	0.83	8.29	10.18	0.07

Table 4 Comparison of the P model and 3P for $\gamma = 0.2$

Cplex				GUROBI								
SOCP		3P		SOCP		3P		Gaps		ERA-H		
0.025	0.18	0.002	0.04	0.029	0.07	0.002	0.06	0.00	0.00	2e-4	0.01	0.07
0.010	0.09	0.001	0.03	0.001	0.04	0.001	0.04	0.00	0.00	2e-4	0.01	0.62
0.010	0.08	0.001	0.04	0.002	0.04	0.001	0.04	0.00	0.00	2e-4	0.01	0.68
0.011	0.08	0.001	0.04	0.002	0.03	0.001	0.03	0.00	0.00	2e-4	0.01	0.53
0.009	0.08	0.001	0.04	0.002	0.03	0.001	0.03	0.00	0.00	2e-4	0.01	0.65
0.010	0.12	0.001	0.03	0.002	0.04	0.001	0.05	0.00	0.00	2e-4	0.01	0.48
0.039	0.36	0.008	0.18	0.010	0.29	0.009	0.28	0.00	0.00	3e-3	0.02	0.57
0.037	0.42	0.009	0.13	0.010	0.25	0.010	0.25	0.00	0.00	3e-3	0.02	0.60
0.036	0.38	0.008	0.32	0.010	0.21	0.010	0.21	0.00	0.24	3e-3	0.01	0.58
0.036	0.37	0.008	0.32	0.010	0.23	0.010	0.24	0.00	0.24	3e-3	0.02	0.58
0.009	0.03	0.000	0.00	0.007	0.03	0.000	0.00	0.00	0.00		0.00	0.00
0.012	0.05	0.001	0.02	0.009	0.04	0.002	0.04	0.00	0.00		0.00	0.06
0.011	0.04	0.001	0.02	0.007	0.04	0.001	0.03	0.00	0.00	3e-4	0.01	0.09
0.007	0.03	0.000	0.00	0.000	0.01	0.000	0.00	0.00	0.00		0.00	0.00
0.014	0.05	0.001	0.02	0.004	0.02	0.000	0.01	0.00	0.00	2e-4	0.01	0.07
0.027	0.12	0.003	0.12	0.014	0.06	0.002	0.06	0.00	0.00		0.00	0.09
0.015	0.07	0.001	0.01	0.007	0.03	0.001	0.01	0.00	0.00	3e-4	0.01	0.00
0.012	0.03	0.001	0.01	0.007	0.04	0.000	0.01	0.00	0.00	2e-4	0.01	0.03
0.019	0.08	0.003	0.05	0.010	0.09	0.005	0.09	0.00	0.00	9e-4	0.01	0.16
0.241	1.02	0.053	1.05	0.365	9.72	0.089	8.41	0.00	0.34	1e-2	0.03	0.13
0.018	0.07	0.001	0.06	0.011	0.09	0.002	0.04	0.00	0.00	7e-4	0.01	0.06
0.093	0.44	0.013	0.35	0.121	1.40	0.023	1.42	0.00	0.24	2e-3	0.01	0.15
0.141	0.63	0.030	0.56	0.223	3.88	0.063	3.95	0.00	0.24	5e-3	0.01	0.22
0.016	0.08	0.001	0.02	0.012	0.04	0.001	0.03	0.00	0.00	2e-4	0.01	0.06
0.013	0.04	0.002	0.03	0.010	0.07	0.003	0.06	0.00	0.00	9e-5	0.01	0.14
0.009	0.03	0.001	0.02	0.009	0.04	0.001	0.03	0.00	0.00		0.00	0.11
0.010	0.06	0.000	0.00	0.006	0.04	0.000	0.00	0.00	0.00		0.00	0.00
0.029	0.32	0.006	0.26	0.032	0.24	0.010	0.23	0.00	0.00	5e-4	0.01	0.17
0.032	0.21	0.000	0.02	0.024	0.11	0.001	0.03	0.00	0.00		0.00	0.02
0.015	0.13	0.003	0.13	0.010	0.08	0.003	0.08	0.00	0.00	6e-4	0.01	0.19
0.014	0.06	0.000	0.01	0.012	0.03	0.000	0.00	0.00	0.00		0.00	0.00
0.140	0.63	0.017	0.50	0.186	0.85	0.025	0.65	0.00	0.59	3e-3	0.01	0.11
0.016	0.11	0.003	0.10	0.009	0.05	0.004	0.05	0.00	0.00	1e-3	0.01	0.18
1.86	53.2	0.42	4.3	2.30	11.0	0.55	4.84	0.01	0.54	0.17	0.26	0.12
23.57	332.5	16.22	145.2	10.41	131.5	15.99	40.51	0.01	0.84	7.97	9.65	0.10

Table 5 Comparison of the P model and 3P for $\gamma = 0.4$

Cplex				GUROBI								
SOCP		3P		SOCP		3P		Gaps		ERA-H		
0.029	0.08	0.003	0.04	0.018	0.11	0.005	0.12	0.00	0.00	2e-4	0.01	0.22
0.004	0.07	0.000	0.02	0.001	0.06	0.000	0.04	0.00	0.00	8e-5	0.01	0.50
0.004	0.06	0.000	0.02	0.001	0.05	0.000	0.03	0.00	0.00	1e-4	0.01	0.57
0.004	0.06	0.000	0.01	0.001	0.02	0.000	0.02	0.00	0.00	9e-5	0.01	0.28
0.003	0.03	0.000	0.02	0.001	0.03	0.000	0.02	0.00	0.00	9e-5	0.01	0.43
0.004	0.05	0.000	0.02	0.001	0.03	0.000	0.02	0.00	0.00	9e-5	0.01	0.38
0.016	0.20	0.002	0.14	0.005	0.27	0.004	0.26	0.00	0.00	1e-3	0.01	0.54
0.016	0.23	0.003	0.25	0.005	0.17	0.004	0.18	0.00	0.00	1e-3	0.01	0.56
0.014	0.20	0.003	0.12	0.005	0.16	0.004	0.15	0.00	0.00	1e-3	0.02	0.57
0.014	0.19	0.003	0.12	0.005	0.22	0.004	0.21	0.00	0.00	1e-3	0.02	0.57
0.007	0.02	0.000	0.01	0.004	0.02	0.000	0.01	0.00	0.00		0.00	0.06
0.013	0.06	0.002	0.02	0.008	0.05	0.003	0.05	0.00	0.00		0.00	0.15
0.010	0.03	0.001	0.03	0.005	0.04	0.001	0.04	0.00	0.00	2e-4	0.01	0.13
0.003	0.01	0.000	0.00	0.000	0.01	0.000	0.00	0.00	0.00		0.00	***
0.007	0.04	0.000	0.00	0.001	0.01	0.000	0.00	0.00	0.00		0.00	0.00
0.019	0.06	0.000	0.02	0.007	0.05	0.001	0.04	0.00	0.00		0.00	0.04
0.013	0.05	0.001	0.02	0.004	0.02	0.001	0.03	0.00	0.00	2e-4	0.01	0.09
0.010	0.04	0.001	0.01	0.005	0.04	0.000	0.01	0.00	0.00		0.00	0.03
0.017	0.15	0.004	0.15	0.006	0.05	0.003	0.05	0.00	0.00	6e-4	0.01	0.22
0.270	1.61	0.070	1.66	0.285	2.28	0.090	2.40	0.01	0.69	1e-2	0.03	0.27
0.015	0.07	0.002	0.04	0.008	0.04	0.002	0.03	0.00	0.00	5e-4	0.01	0.13
0.092	0.61	0.017	0.55	0.090	0.43	0.023	0.40	0.01	0.41	2e-3	0.01	0.24
0.142	1.08	0.039	1.08	0.150	0.85	0.065	0.89	0.01	0.77	4e-3	0.02	0.38
0.013	0.05	0.001	0.03	0.008	0.04	0.002	0.04	0.00	0.00	1e-4	0.01	0.10
0.010	0.05	0.001	0.02	0.005	0.06	0.001	0.06	0.00	0.00	9e-5	0.01	0.12
0.008	0.03	0.002	0.03	0.007	0.04	0.003	0.04	0.00	0.00		0.00	0.26
0.009	0.08	0.000	0.00	0.005	0.03	0.000	0.00	0.00	0.00		0.00	0.00
0.027	0.23	0.007	0.23	0.024	0.23	0.010	0.24	0.00	0.27	4e-4	0.01	0.24
0.026	0.15	0.001	0.02	0.018	0.07	0.001	0.03	0.00	0.00		0.00	0.05
0.010	0.06	0.002	0.04	0.006	0.05	0.003	0.04	0.01	0.30	3e-4	0.01	0.25
0.010	0.02	0.000	0.00	0.008	0.02	0.000	0.00	0.00	0.00		0.00	0.00
0.139	0.82	0.023	0.56	0.162	0.90	0.037	0.74	0.01	0.57	3e-3	0.01	0.21
0.012	0.06	0.002	0.04	0.005	0.05	0.003	0.04	0.00	0.00	6e-4	0.01	0.26
1.82	38.3	0.55	21.5	2.126	17.2	0.67	6.71	0.02	0.60	0.17	0.25	0.21
28.83	373.6	15.48	206.6	9.670	136.5	15.00	49.36	0.03	0.74	7.73	9.24	0.36

Table 6 Comparison of the P model and 3P for $\gamma = 0.8$