Lagrangian Approaches for QoS Scheduling in Computer Networks

Published Version:
Frangioni, A., Galli, L., Sorbera, E. (2024). Lagrangian Approaches for QoS Scheduling in Computer Networks. Cham : Springer Nature [10.1007/978-3-031-47686-0_2].

Availability:
This version is available at: https://hdl.handle.net/11585/983163 since: 2025-01-03

Published:
DOI: http://doi.org/10.1007/978-3-031-47686-0_2

Lagrangian approaches for QoS scheduling in computer networks

Antonio Frangioni[1], Laura Galli[1] and Enrico Sorbera[2]

[1] Dipartimento di Informatica, Università di Pisa,
Largo B. Pontecorvo 3, 56127 Pisa, Italy
frangio@di.unipi.it, laura.galli@unipi.it
[2] Dipartimento di Matematica, Università di Pisa,
Largo B. Pontecorvo 5, 56127 Pisa, Italy
enrico.sorbera@hotmail.it

Abstract. We study a routing problem arising in computer networks where stringent Quality of Service (QoS) scheduling requirements ask for a routing of the packets with controlled worst-case "end-to-end" delay. With widely used delay formulæ, this is a shortest-path-type problem with a nonlinear constraint depending in a complex way on the reserved rates on the chosen arcs. However, when the minimum reserved rate in the path is fixed, the Lagrangian problem obtained by relaxing the delay constraint presents a special structure and can be solved efficiently. We exploit this property and present an effective method that provides both upper and lower bounds of very good quality in extremely short computing times.

Keywords: Lagrangian relaxation, Heuristics, Mixed-integer non-linear programs

1 Introduction

Many of todays real-life applications in computer networks (e.g., voice/video streaming, remote operation of industrial/medical tools, etc.) have stringent Quality of Service (QoS) scheduling requirements in terms of controlled "end-to-end" delay, a.k.a. *worst-case delay* (WCD). Considering the packet-based nature of the IP network infrastructure, these applications require the WCD of any packet in a flow (from now on, the WCD of the flow) to be bound by a given threshold. The WCD of a flow is a function of both the (single) path on the network where it is routed (a "discrete" decision), and of the bandwidth allocated to the packets of the flow (a.k.a. *reserved rate*) on each of the arcs of the path (a "continuous" decision). Hence, from an optimization point of view, the Delay Constrained Routing Problem (DCR) consists of computing paths and reserving resources along the IP network subject to WCD constraints. Customarily, the objective function is taken to be the total amount of bandwidth allocated along the path (albeit possibly weighted by arc) in order to leave as much as possible "free space" to other flows to be routed in the future.

A well-known paradigm for QoS scheduling is the Generalized Processor Sharing (GPS) [7], which allows "per-link" WCD bounds to be computed if the traffic arrival rate of the flow at the source is constrained. In particular, Strictly Rate-Proportional (SRP) schedulers are a practical implementation of the GPS paradigm, in which the *latency* (i.e., worst-case scheduling delay at a link) is a closed formula inversely proportional to the reserved rate. Since the WCD of a flow depends on the reserved rates along its path, QoS routing problems with WCD constraints can be mathematically formulated, under the assumption of SRP schedulers in the network. In particular, the single-flow single-path case defined in [1] can be formulated as a Mixed-Integer Second-Order Cone Program (MISOCP), and solved with general-purpose optimization software on realistic instances. This is shown to have positive impacts on the (simulated) functioning of the telecommunication network in [2]. Yet, such a DCR problem is significantly more difficult than usual shortest path routing problems, and it is clearly $\mathcal{NP}$-hard being a generalization of the Constrained Shortest Path problem (CSP); furthermore, due to the nonlinear nature of the WCD constraints, classical approaches for CSP cannot be adapted in a straightforward way. While the use of tight MISOCP models and general-purpose solvers provides optimal solutions quickly in many real or realistic networks, running times do increase substantially as the size of the network does; besides, practical implementations of the concept may have issues (software environment, licensing, ...) with incorporating such complex components as state-of-the art MISOCP solvers.

In this paper we provide a first step towards a solver-free solution approach for DCR by showing that, when the minimum allocated rate in the network is fixed, the Lagrangian problem obtained by relaxing the WCD constraint presents a "special" structure and can be solved efficiently. We take advantage of this property to construct an effective method that provides both upper and lower bounds of very good quality in extremely short computing times. This has the potential to form the basis of efficient bespoke approaches for the full DCR problem.

2 Problem definition

In this section we define the (Single-Flow Single-Path) DCR problem. The IP network is represented by a directed graph $G = (N, A)$, where N is the set of nodes, and A is the set of links (i.e., arcs). Each node $i \in N$ is characterized by a fixed *node delay* n_i. Each link $(i, j) \in A$ is characterized by a fixed *link delay* l_{ij}, a *physical link speed* w_{ij}, a *reservable capacity* c_{ij} ($\leq w_{ij}$, since in general other flows may be present in the network when the new one is routed), and a *link reservation cost* f_{ij} (i.e., the cost of reserving one unit of capacity on (i, j)).

We are given a single "new" (unsplittable) flow to be routed on a single path from a source $s \in N$ to a given destination $d \in N \setminus \{ s \}$. The number of bits of the flow expected to enter the origin s (and to be dispatched across the network to the destination d) is described by its *arrival curve* $A(\tau) : \mathbb{R}_+ \to \mathbb{R}_+$ depending on time τ; we assume the classical *leaky-bucket* traffic shaper from the traffic engineering literature, resulting in an affine function $A(\tau) = \sigma + \rho\tau$ with parameters σ and ρ, called, respectively, *burst* and *rate*. Also, the *maximum transmit unit* L (i.e., the maximal size of any packet) is known and assumed constant. The flow has a *deadline* δ, i.e., a threshold which bounds from above the maximum time that every bit in the flow is allowed to spend traversing the network from s to d. In other words, the *worst-case delay* of the flow must be at most δ. Given the costs f_{ij}, the goal of the (DCR) problem is to find one s-d path and to reserve capacity along its arcs in such a way that the corresponding WCD does not exceed the deadline δ and the total reservation cost is minimized.

The WCD of a s-d flow depends on several factors and requires a sophisticated analysis that can be performed via *network calculus* [6]. Here we summarize the key elements: (i) the selected s-d *path* P in G; (ii) the *reserved rate* (i.e., reserved capacity) $0 \leq r_{ij} \leq c_{ij}$ for the arcs of P; (iii) the specific characteristics of the software/hardware components at the nodes of the network that implement the scheduling algorithm used for the flows entering and leaving the nodes (i.e., *schedulers*). For a start, in order to have a finite delay, the minimum reserved rate along the path must be at least as large as the arrival rate ρ of the flow, i.e., $r_{ij} \geq \rho$ for all $(i,j) \in P$. If this condition is satisfied, the general form of the WCD for a given path P is given by

$$\frac{\sigma}{\min\{\, r_{ij} \,:\, (i,j) \in P \}} + \sum_{(i,j)\in P} \big(\, \theta_{ij} + l_{ij} + n_i \,\big) \tag{1}$$

where θ_{ij} is the delay experienced by the flow on traversing the link (i,j) (i.e., the latency), and depends on the scheduling protocol. Indeed, the exact form of θ_{ij} depends on the scheduling algorithm implemented at the nodes of the network. As in [1,2] we assume the latency expression

$$\theta_{ij} = \frac{L}{r_{ij}} + \frac{L}{w_{ij}}, \tag{2}$$

which corresponds to a Strictly Rate-Proportional (SRP) scheduler, bearing its name to the inverse relationship to the reserved rate r_{ij}. The advantage of SRP schedulers is twofold: (i) the latency of a flow does not depend on the other flows in the network, so one can ignore feasibility issues related to the other flows previously routed; (ii) the latency function (2) is a convex function of r_{ij} (if $r_{ij} \geq 0$). Note that, the component $L \,/\, w_{ij}$ of (2) is just an additive constant term, that can be treated in the same way as the link delay l_{ij} and the node delay n_i. Hence, to ease notation, from now on we denote by $\bar{l}_{ij} = L \,/\, w_{ij} + l_{ij} + n_i$ the constant delay component of an arc (i,j), while $\theta_{ij} = L \,/\, r_{ij}$ represents the variable part of the arc latency which depends on the reserved rate r_{ij}.

3 Problem formulation

We now describe a model for the DCR problem, that will be used in the rest of this study. This model is similar to the one derived in [1] that can be reformulated as a MISOCP, but with some specific changes useful for the algorithmic techniques that will be presented later on.

$$\min \sum_{(i,j)\in A} f_{ij} r_{ij} \tag{3}$$

$$\sum_{(j,i)\in BS(i)} x_{ji} - \sum_{(i,j)\in FS(i)} x_{ij} = \begin{cases} -1 & \text{if } i = s \\ 1 & \text{if } i = d \\ 0 & \text{otherwise} \end{cases} \qquad i \in N \tag{4}$$

$$\frac{\sigma}{r_{\min}} + \sum_{(i,j)\in A} \Big(\frac{L {x_{ij}}^2}{r_{ij}} + \bar{l}_{ij} x_{ij} \Big) \leq \delta \tag{5}$$

$$r_{\min} x_{ij} \leq r_{ij} \leq c_{ij} x_{ij} \qquad (i,j) \in A \tag{6}$$

$$x_{ij} \in \{\, 0 \,,\, 1 \,\} \qquad (i,j) \in A \tag{7}$$

$$r_{\min} \in [\, \rho \,,\, c_{max} \,] \tag{8}$$

The arc-flow binary variables x_{ij} indicate whether arc (i, j) belongs to the chosen path, and (4) are the standard flow conservation constraints. Reserved rate variables r_{ij} represent the amount of capacity reserved on arc (i, j), and $r_{\min}$ (in principle, a variable) captures the minimum reserved rate along the path. Denoting by $c_{max} = \max_{(i,j)\in A}\{c_{ij}\}$, constraints (6) ensure that $r_{ij} = 0$ if $x_{ij} = 0$, and $\rho \leq r_{\min} \leq r_{ij}$ if $x_{ij} = 1$. Note that the left-hand side of (6) is nonlinear and nonconvex if $r_{\min}$ is a variable; in fact, in [1] the alternative (linear) form

$$0 \leq r_{ij} \leq c_{ij} x_{ij} \qquad (i,j) \in A \tag{9}$$

$$\rho \leq r_{\min} \leq r_{ij} + c_{max}(1 - x_{ij}) \qquad (i,j) \in A \tag{10}$$

was used. Since in our development $r_{\min}$ is fixed, the (clearly, tighter) version (6) is preferable.

The nonlinear term $\sigma \,/\, r_{\min}$ of the WCD constraint can be expressed via a rotated SOCP constraint, adding an auxiliary variable t

$$t \; r_{\min} \geq \sigma, \quad t \geq 0 \tag{11}$$

Finally, the latency θ_{ij} on arc (i, j), which is expressed by the disjunction

$$\theta_{ij} = \begin{cases} L \,/\, r_{ij} & \text{if } x_{ij} = 1 \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

can be represented in convex way using the *Perspective Reformulation* technique (and another rotated SOCP) as shown in [3]

$$\theta_{ij} r_{ij} \geq L x_{ij}^2 \qquad (i,j) \in A \tag{13}$$

4 Lagrangian relaxation

Our approach starts by *fixing* the value $\bar{r}_{\min} \in [\, \rho \,,\, c_{max} \,]$ in (3)–(8). We refer to the corresponding problem as $DCR(\bar{r}_{\min})$. Then, we relax in a Lagrangian way the WCD constraint (5) with a non-negative multiplier λ, to obtain the Lagrangian function

$$\phi(\lambda) = -\lambda\delta + \frac{\lambda\sigma}{\bar{r}_{\min}} + \min_{(x,r)} \left\{ \sum_{(i,j)\in A} f_{ij} r_{ij} + \frac{\lambda L {x_{ij}}^2}{r_{ij}} + \lambda \bar{l}_{ij} x_{ij} : (4), (6), (7) \right\} \quad (14)$$

and therefore the corresponding (univariate) Lagrangian dual

$$LDCR(\bar{r}_{\min}) \quad \max\{\, \phi(\lambda) \,:\, \lambda \geq 0 \,\} \quad (15)$$

The important property of (14) is that, when λ is fixed (call it $\bar{\lambda}$), we can "project out" the r_{ij} variables onto the x–space. Indeed, if $x_{ij} = 0$, the corresponding reserved rate is forced to be zero: $r_{ij} = 0$. If, rather, $x_{ij} = 1$, the corresponding optimal r_{ij} value is obtained by solving the one-dimensional convex optimization problem

$$\min \left\{\, h(\, r_{ij}\,) = f_{ij} r_{ij} + \bar{\lambda} L \,/\, r_{ij} \,:\, \bar{r}_{\min} \leq r_{ij} \leq c_{ij} \,\right\} \quad (16)$$

The solution $h'(\, r_{ij}\,) = f_{ij} - \bar{\lambda} L \,/\, r_{ij}^2 = 0$ is $\sqrt{\bar{\lambda} L \,/\, f_{ij}}$ if $f_{ij} > 0$, which then is the (unique) unconstrained minimum of $h(\, r_{ij}\,)$; if $f_{ij} \leq 0$ instead, the derivative is always negative and the optimum of (16) is on the right endpoint of the interval. Thus, the optimal value for r_{ij} is

$$r_{ij}^* = \begin{cases} c_{ij} & \text{if } f_{ij} \leq 0 \\ \min\left\{ c_{ij} \,,\, \max\{\, \bar{r}_{\min} \,,\, \sqrt{\bar{\lambda} L \,/\, f_{ij}} \,\}\right\} & \text{otherwise} \end{cases} \quad (17)$$

After projecting onto the x–space, we can then reformulate (14) as

$$-\bar{\lambda}\delta + \bar{\lambda}\sigma \,/\, \bar{r}_{\min} + \min \left\{\, \textstyle\sum_{(i,j)\in A} \tilde{c}_{ij} x_{ij} \,:\, (4)\,,\, (7) \,\right\} \quad (18)$$

which is a Shortest Path Problem (SPP) with non-negative arc costs $\tilde{c}_{ij} = \bar{\lambda}\bar{l}_{ij} + h(r_{ij}^*)$. Therefore, $\phi(\lambda)$ can be efficiently computed for any value of λ. Note that for a fixed $\bar{r}_{\min}$ the corresponding SPP may turn out to be infeasible: this means that there is no s-d path in G with reservable rates $c_{ij} \geq \bar{r}_{\min}$. This does not depend on λ and it is immediately discovered the first time the SPP is solved, immediately proving that the corresponding $DCR(\bar{r}_{\min})$ is empty.

The efficient solution of the Lagrangian dual (15) is then possible, as this is a one-dimensional optimization problem where the objective to be maximized—being the pointwise minimum of a finite family of affine functions—is a concave nondifferentiable function. The standard approach is the adaptation of the classical Cutting Plane algorithm to the one-dimensional case, giving rise to the well-known line search for concave nondifferentiable functions. To describe it we introduce

$$\beta(x,r) = \sigma \,/\, \bar{r}_{min} - \delta + \textstyle\sum_{(i,j)\in A} L x_{ij}^2 \,/\, r_{ij} + \bar{l}_{ij} x_{ij} \quad (19)$$

$$\alpha(x,r) = \textstyle\sum_{(i,j)\in A} f_{ij} r_{ij} \quad (20)$$

which enable us to rewrite (14) as

$$\phi(\lambda) = \alpha\big(\, x(\lambda)\, ,\, r(\lambda)\,\big) + \lambda\beta\big(\, x(\lambda)\, ,\, r(\lambda)\,\big)$$

where $\big(\, x(\lambda)\, ,\, r(\lambda)\,\big)$ is any optimal solution to (14); $\beta\big(\, x(\lambda)\, ,\, r(\lambda)\,\big)$ is well-known to be a subgradient of $\phi(\lambda)$ in λ. At each iteration of the *line search* we need two values of λ, $0 \leq \lambda^+ < \lambda^-$, such that $\beta(\, x(\lambda^+)\, ,\, r(\lambda^+)\,) > 0$ and $\beta(\, x(\lambda^-)\, ,\, r(\lambda^-)\,) < 0$. This allows us to construct two lines

$$r^+(\lambda) = m^+\lambda + q^+ \qquad r^-(\lambda) = m^-\lambda + q^-$$

where $m^\pm = \beta(\, x(\lambda^\pm)\, ,\, r(\lambda^\pm)\,)$, $q^\pm = \alpha(\, x(\lambda^\pm)\, ,\, r(\lambda^\pm)\,)$, which approximate the Lagrangian function $\phi(\lambda)$ from above. Next, we compute the point

$$\hat{\lambda} = \frac{q^- - q^+}{m^+ - m^-} \in [\, \lambda^+\, ,\, \lambda^-\,]$$

corresponding to the intersection of the two lines and the corresponding $x(\hat{\lambda})$, $r(\hat{\lambda})$: if $\beta(\, x(\hat{\lambda})\, ,\, r(\hat{\lambda})\,) < 0$ then $\hat{\lambda}$ replaces λ^-, otherwise it replaces λ^+. The process is iterated until either the distance between λ^+ and λ^- is small enough, or a subgradient $\beta\big(\, x(\lambda)\, ,\, r(\lambda)\,\big) \approx 0$ is found.

To initialise the process we need to find two starting values for λ^+ and λ^-. For the former we can always take $\lambda^+ = 0$: indeed, if $\beta(\, x(0)\, ,\, r(0)\,) \leq 0$ we can immediately conclude that the optimum is $\lambda^* = 0$. For the latter, we start with any strictly positive value (say, $\lambda^- = 1$) and we iteratively increase (say, double) it until $\beta(\, x(\lambda^-)\, ,\, r(\lambda^-)\,) \leq 0$ holds. The latter condition will surely be finitely (and rapidly) obtained if $\phi(\lambda)$ is bounded above; but this may not necessarily be the case, signalling that $DCR(\bar{r}_{\min})$ is empty (even if, possibly, the SPP is feasible). The only crucial point of the initialization phase is therefore to find a condition to stop increasing λ^- and finitely conclude that $LDCR(\bar{r}_{\min})$ is unbounded above "without waiting for λ^- to go all the way to $+\infty$".

This can be easily done since the feasible region of $DCR(\bar{r}_{\min})$ is compact. Therefore, one can easily compute a valid *upper bound* on its optimum value *if the latter is finite* by solving

$$U = \max\big\{\, \textstyle\sum_{(i,j)\in A} f_{ij} r_{ij} \;:\; \bar{r}_{min} x_{ij} \leq r_{ij} \leq c_{ij} x_{ij}\, ,\, x_{ij} \in [\, 0\, ,\, 1\,] \quad (i,j) \in A \,\big\}$$

which, being trivially separable in the $(\, x_{ij}\, ,\, r_{ij}\,)$ pairs, can obviously be done by inspection. Now, if $DCR(\bar{r}_{\min})$ is nonempty, its optimum value v^* necessarily must satisfy $v^* \leq U$; on the other hand, by weak duality it must be that $\phi(\lambda) \leq v^*$ for all $\lambda \geq 0$. Thus, if during the initialization phase we find a λ^- such that $\phi(\lambda^-) > U$, then we have immediately proved that $DCR(\bar{r}_{\min})$ is empty (and $LDCR(\bar{r}_{\min})$ unbounded above). In this way we are sure that the initialization phase always finitely terminates, and therefore so does the solution of $LDCR(\bar{r}_{\min})$.

As a by-product of solving $LDCR(\bar{r}_{\min})$, feasible solutions to $DCR(\bar{r}_{\min})$ are necessarily obtained (unless the latter is empty and the former unbounded).

Indeed, anytime we find a point λ with negative subgradient we know that the corresponding constraint (5) is satisfied in $(\, x(\lambda)\, ,\, r(\lambda)\,)$, as

$$\beta(\, x(\lambda)\, ,\, r(\lambda)\,) \leq 0 \iff \sigma \,/\, \bar{r}_{min} + \sum_{(i,j)\in A} Lx_{ij}^2(\lambda) \,/\, r_{ij}(\lambda) + \bar{l}_{ij}x_{ij}(\lambda) \leq \delta$$

and thus represents a feasible solution to $DCR(\bar{r}_{\min})$. Thus, our Lagrangian approach naturally also doubles as a matheuristic that produces feasible solutions at some Lagrangian iterations (save for the case where (15) is unbounded above, as previously discussed).

5 Computational experiments

In this section we report the results of our computational results, in all of which $\bar{r}_{\min}$ is always considered to be *fixed*, hence a parameter of our models. The primary goal is to compare the continuous relaxation of (3)–(5),(7),(9)–(10) (the model in the original [1]) with the Lagrangian dual (15) in terms of both the quality of the bound and the required running time. The former is solved using `Cplex` on the corresponding MISOCP formulation, the latter is solved using our implementation of the method described in Section 4. A secondary goal is to evaluate the effectiveness of the Lagrangian heuristic naturally embedded in the solution of the Lagrangian dual, as compared (in both gap and time) with the optimal solution of the MISOCP model obtained by `Cplex`. All the experiments have been performed on a VM running on a 2.20GHz Intel Xeon 5120 CPU with 64Gb RAM, under `Ubuntu 22.4`. All the codes were compiled with `gcc 11.3.0`, `Cplex` version was `12.10`.

The test instances were generated as in [1], but here we briefly review the procedure for ease of reading. We considered both real-world and synthetic network topologies. The real-world ones are taken from the `GARR` subset [4] of the Internet Topology Zoo [5] and the `SNDlib` topologies [9], while the random topology was generated using the *Waxman* model [10], with $n = 100$ and density $n/m = 0.4$. The information on the topology was combined with the data generated via the `FNSS` tool [8], which provides a number of expert-tuned options to create realistic telecommunications networks parameters as well as traffic matrices (i.e., the flows). All the link capacities were chosen among $\{1, 10, 40\}$ Gbps, according to the link's *edge betweeness* [8]. We set $L = 1500$ bytes at all links. Link delays l_{ij} and node delays n_i were set equal to L/w_{ij}. Bursts σ were set to $3L$, while flow rates ρ were taken from a lognormal distribution with $\mu = 0.8$ Gbps and $\sigma^2 = 0.05$ [8]. To define the flow deadlines δ we proceeded in two steps: first we computed a lower and an upper bound on the WCD, namely δ_{min} under which no routing is possible, and δ_{max}, over which the deadline constraint becomes redundant; the δ was chosen uniformly within the interval $[\,\delta_{min}\,,\, \delta_{min} + (\delta_{max} - \delta_{min})\pi\,]$ for a fixed parameter $\pi = 0.2$, which produces fairly tight deadlines.
The instances can be downloaded from https://commalab.di.unipi.it/datasets/mmcf/#FNSS.

Table 1 shows the results obtained by our optimisation approach. We consider 7 topologies with the following features:

- `abilene`: 12 nodes, 15 links, 31 flows
- `atlanta`: 15 nodes, 22 links, 45 flows
- `Cogentco`: 197 nodes, 486 links, 1000 flows
- `Colt`: 153 nodes, 354 links, 1000 flows
- `Garr199904`: 23 nodes, 50 links, 506 flows
- `Garr200109`: 22 nodes, 48 links, 462 flows
- `w1-100-04`: 100 nodes, 414 links, 664 flows

where `w1-100-04` is a randomly generated Waxman [10] topology. We fixed 10 different value of $\bar{r}_{\min}$ in the interval $[\,\rho\,,\,c_{max}\,]$ (which depends on the specific flow considered) as $\bar{r}_{\min} = \rho + \alpha(c_{max} - \rho)$, where α ranges between 0 and 0.9 with spacing 0.1. Therefore, an instance is defined by the topology, one of its flows, and a value of α (which determines $\bar{r}_{\min}$). Each row corresponds to the average results among all the flows of the given topology. However, several different values of α (especially "large" ones) gave rise to basically the same results, with very minor changes; for the sake of readability we reported results only for the smallest among these values of α.

The columns are divided into two groups: *feasible* and *unfeasible* flows. We distinguish the two groups because, on one hand, we want to measure the quality of the bounds found with our method when the instance is feasible, and on the other hand it is interesting to report the effectiveness of our approach in detecting unfeasible instances. In each group, the last three columns report the (average) computing times in seconds, where "tL" is that for solving the Lagrangian dual using the method described in Section 4, "tC" is that for solving the continuous relaxation of the MISOCP model, and "tE" that for solving the MISOCP model to integer optimality, in both cases using `CPLEX`. In the former group we also report the percentage $\%f$ of feasible flows and the quality of the bounds with respect to the optimal solution of $DCR(\bar{r}_{\min})$, where "gapL" is the gap of the Lagrangian bound, "gapC" is the gap of the continuous relaxation bound, and "gapH" is the gap of the best solution found by the Lagrangian heuristic. Gaps smaller than `1e-6` (the optimality tolerance for `CPLEX`) are clearly meaningless and are therefore tabulated as blanks. In the latter group we rather report the percentage of unfeasible flows detected, respectively, by the Lagrangian method ("%uL") and the continuous relaxation ("%uC').

The Table paints a clear picture whereby the computing time of the Lagrangian approach is several orders of magnitude smaller with respect to the continuous relaxation, and of course a fortiori the exact solution, via `CPLEX`. As for the gaps, in the vast majority of the cases the Lagrangian bound is exact, i.e., equal to the optimal value: correspondingly, *all* unfeasible instances are detected as such by the Lagrangian approach. The continuous relaxation bound is also exact in a good number of cases, especially in some topologies and for large values of α (that lead to many unfeasible flows), but much less so than the Lagrangian bound; in many cases the gap is rather higher, even of the order of `1e-1`, and

	α	feasible flows %f	gapL	gapC	gapH	tL	tC	tE	unfeasible flows %uL	%uC	tL	tC	tE
abilene	0	0				1.00e-6	0.00	0.00	100	100	1.47e-5	0.002	0.005
	0.1	32		4.52e-2	1.46e-6	1.35e-5	3.49	8.83	100	95	1.04e-5	0.002	0.487
	0.2	45		7.86e-2		1.35e-5	5.77	10.78	100	94	9.12e-6	0.003	0.421
	0.3	41				1.13e-5	4.15	7.95	100	83	6.50e-6	0.003	0.223
	0.4	48				1.07e-5	4.86	8.58	100	81	5.94e-6	0.002	0.667
	0.5	48				7.40e-6	5.08	9.67	100	81	5.75e-6	0.002	0.341
	0.6	48				6.67e-6	5.52	11.59	100	100	6.00e-6	0.002	0.005
	0.7	48				7.13e-6	7.06	13.34	100	100	5.94e-6	0.002	0.005
atlanta	0	0				1.00e-6	0.00	0.00	100	100	1.73e-5	0.002	0.005
	0.1	46		7.80e-2	1.43e-6	1.54e-5	2.86	6.94	100	88	1.24e-5	0.003	0.606
	0.2	75		1.07e-1		9.85e-6	3.77	7.60	100	73	6.82e-6	0.002	0.826
	0.3	13				7.00e-6	6.70	11.76	100	69	5.51e-6	0.003	0.579
	0.4	15				8.71e-6	5.06	7.97	100	71	5.55e-6	0.002	0.608
	0.5	15				6.57e-6	4.27	12.11	100	74	5.63e-6	0.002	0.593
	0.6	15				7.00e-6	4.28	11.32	100	84	5.71e-6	0.002	0.567
	0.7	15				6.43e-6	5.84	11.69	100	100	5.79e-6	0.002	0.005
Cogentco	0	56	1.75e-3	1.75e-3	5.42e-3	4.24e-3	2.16	4.89	100	100	2.74e-3	0.126	0.246
	0.1	6	7.73e-5	3.52e-2	6.25e-2	4.18e-4	2.27	5.69	100	17	2.22e-5	0.007	1.968
	0.2	6	4.51e-4	2.07e-1	7.25e-2	1.18e-4	2.25	9.06	100	22	2.10e-5	0.030	1.904
	0.3	0				1.00e-6	0.00	0.00	100	41	1.79e-5	0.079	1.460
	0.4	0				1.00e-6	0.00	0.00	100	60	2.15e-5	0.100	1.026
	0.5	0				1.00e-6	0.00	0.00	100	71	2.49e-5	0.121	0.772
	0.6	0				1.00e-6	0.00	0.00	100	100	1.73e-5	0.031	0.035
Colt	0	51			9.88e-6	2.33e-3	1.92	4.75	100	100	1.44e-3	0.066	0.131
	0.1	3		2.15e-2	2.38e-6	1.31e-4	1.68	4.12	100	85	1.49e-5	0.005	0.320
	0.2	3		1.18e-1		6.34e-5	2.60	7.42	100	85	1.35e-5	0.005	0.303
	0.3	0				1.00e-6	0.00	0.00	100	91	1.13e-5	0.012	0.220
	0.4	0				1.00e-6	0.00	0.00	100	98	1.24e-5	0.024	0.084
	0.5	0				1.00e-6	0.00	0.00	100	99	1.27e-5	0.035	0.084
	0.6	0				1.00e-6	0.00	0.00	100	100	1.04e-5	0.011	0.014
Garr199904	0	0			2.94e-6	1.01e-4	0.41	0.65	100	100	6.52e-5	0.005	0.010
	0.1	0		1.40e-1	1.97e-6	1.68e-5	2.77	5.94	100	100	6.96e-6	0.002	0.005
	0.2	1		4.94e-1		6.71e-6	0.19	5.05	100	100	6.33e-6	0.002	0.005
	0.3	0		3.98e-1		6.60e-6	0.20	4.96	100	99	6.09e-6	0.002	0.006
	0.4	1		3.72e-1	1.04e-6	8.83e-6	1.18	4.46	100	99	5.95e-6	0.002	0.007
	0.5	1		4.04e-1	1.36e-6	1.15e-5	0.46	4.28	100	99	5.93e-6	0.002	0.013
	0.6	1		4.03e-1		6.50e-6	0.25	4.81	100	100	5.63e-6	0.002	0.006
	0.7	1		3.38e-1		8.56e-6	0.72	5.24	100	100	5.62e-6	0.002	0.005
Garr200109	0	0			2.94e-6	1.01e-4	0.41	0.65	100	100	6.52e-5	0.005	0.010
	0.1	0		1.40e-1	1.97e-6	1.68e-5	2.77	5.94	100	100	6.96e-6	0.002	0.005
	0.2	1		4.94e-1		6.71e-6	0.19	5.05	100	100	6.33e-6	0.002	0.005
	0.3	0		3.98e-1		6.60e-6	0.20	4.96	100	99	6.09e-6	0.002	0.006
	0.4	1		3.72e-1	1.04e-6	8.83e-6	1.18	4.46	100	99	5.95e-6	0.002	0.007
	0.5	1		4.04e-1	1.36e-6	1.15e-5	0.46	4.28	100	99	5.93e-6	0.002	0.013
	0.6	1		4.03e-1		6.50e-6	0.25	4.81	100	100	5.63e-6	0.002	0.006
	0.7	1		3.38e-1		8.56e-6	0.72	5.24	100	100	5.62e-6	0.002	0.005
w1-100-04	0	0				1.00e-6	0.00	0.00	100	100	9.61e-4	0.003	0.007
	0.1	79	6.05e-4	1.52e-1	4.83e-2	7.85e-4	2.10	6.76	100	100	1.38e-3	0.257	0.427
	0.2	97	1.27e-3	4.28e-1	4.48e-2	3.02e-4	2.14	7.68	100	100	1.42e-3	0.305	0.403
	0.3	9		4.95e-1		1.04e-4	1.99	2.82	100	0	1.78e-5	0.142	2.055
	0.4	9		5.75e-1		6.83e-5	1.88	2.74	100	1	1.70e-5	0.276	1.950
	0.5	9		6.20e-1		4.66e-5	1.87	2.77	100	0	1.75e-5	0.394	1.961
	0.6	9		6.37e-1		4.44e-5	1.99	2.82	100	1	1.65e-5	0.191	2.156
	0.7	9		6.39e-1		4.95e-5	2.00	2.73	100	52	1.87e-5	0.008	1.020
	0.8	9		5.95e-1		3.84e-5	2.05	3.34	100	74	1.79e-5	0.005	0.571
	0.9	9		4.55e-1		3.08e-5	1.96	2.81	100	100	1.78e-5	0.004	0.007

Table 1. Computational experiments

correspondingly the continuous relaxation fails to detect unfeasibility in a sizable number of cases. As for the Lagrangian heuristic, it also provides optimal solutions in many cases (where the Lagrangian bound is exact), and in most (but not all) cases where the solution is not exact it has a very small gap.

6 Conclusions

Our results clearly show that the Lagrangian approach is extremely competitive in terms of lower bounds and time with the ordinary continuous relaxation solved as a SOCP, and in many cases it provides very good (often exact) solutions to $DCR(\bar{r}_{\min})$. This indicates that it could be a promising starting point for developing approaches to the "full" DCR problem, where $r_{\min}$ is treated as a variable; we envision in particular the possibility of developing Benders' approaches using the Lagrangian bound, a quite unusual arrangement that requires some nontrivial analysis.

References

1. Frangioni, A., Galli, L., Scutellà, M.G.: Delay-Constrained Shortest Paths: Approximation Algorithms and Second-Order Cone Models. Journal of Optimization Theory and Applications, 164, 1051–1077 (2015).
2. Frangioni, A., Galli, L., Stea, G.: Optimal Joint Path Computation and Rate Allocation for Real-time Traffic. The Computer Journal, 58, 1416–1430 (2015).
3. Frangioni, A., Gentile, C.: Perspective Cuts for a Class of Convex 0–1 Mixed Integer Programs. Mathematical Programming, 106, 225–236 (2006).
4. `GARR`, http://www.garr.it.
5. The Internet Topology Zoo, http://www.topology-zoo.org/.
6. 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).
7. Parekh, K., Gallager, R.G.: A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: the Single Node Case. IEEE/ACM Trans. on Networking, 1, 344–357 (1993).
8. Saino, L., Cocora, C., Pavlou, G.: A Toolchain for Simplifying Network Simulation Setup. Proceedings of the 6th International ICST Conference on Simulation Tools and Techniques (2013).
9. `SNDlib`, http://www.sndlib.zib.de.
10. Waxman, B.: Routing of multipoint connections. IEEE Journal on Selected Areas in Communications, 6, 1617–1622 (1988).