# Models and algorithms for the robust resource constrained shortest path problem

Da Lu and Fatma Gzara
Department of Management Sciences, University of Waterloo
200 University Avenue West, Waterloo, ON, Canada

**Abstract**

We study the robust resource constrained shortest path problem (RCSPP) under uncertainty in cost and multiple resource consumption. Contrary to the deterministic RCSPP where the cost and the consumption of resources on an arc are known and fixed, the robust RCSPP models the case where both the cost and the resource consumption are random, and it determines a robust minimum cost path that is feasible with respect to multiple resource constraints. We present a robust optimization model, propose graph reduction techniques tailored for the robust problem, and develop a modified label-setting algorithm that introduces a new dominance rule. We perform extensive numerical testing to compare the modified label-setting algorithm with direct solution of an equivalent deterministic mixed-integer programming model, a sequential algorithm that solves a series of deterministic RCSPP, and a label-setting algorithm proposed by Pessoa et al. [2015]. The label-setting algorithm is comparable to the label-setting algorithm by Pessoa et al. [2015] and outperforms all other approaches significantly.

*Keywords:* robust resource constrained shortest path, robust optimization, graph reduction, label-setting.

## 1. Introduction

Shortest path problems have been widely studied in Operations Research because of their theoretical and practical relevance. While the basic shortest path problem (SPP) is easy to solve and lies at the heart of network flows, extensions with additional restrictions on paths present challenges to solve. Shortest paths appear frequently in practice whenever there is interest to send flow from an origin to a destination in applications as diverse as vehicle routing and scheduling, airline operations planning, transportation, and telecommunications network planning. They also arise often as subproblems when solving optimization problems on networks by decomposition techniques like Lagrangian relaxation, column generation, and Benders decomposition. When additional requirements, such as visiting nodes within a time window, using a limited resource, and avoiding certain paths and cycles are imposed, the problem is referred as constrained shortest path problem (CSPP). We refer interested readers to the surveys by Irnich and Desaulniers [2005]; Irnich [2008]; Pugliese and Guerriero [2013] for classification, modeling issues and solution methodologies for CSPP. In particular, in the deterministic resource constrained shortest path problem (RCSPP) each arc consumes a given amount of a resource with limited availability. The deterministic RCSPP finds a minimum cost path feasible with respect to resource consumption constraints.

In this paper, we study the robust RCSPP with multiple resources by introducing uncertain cost and resource consumption on arcs. More specifically, the uncertain parameters are defined by an interval of uncertainty where the realization of a parameter may take any value within the corresponding interval. The lower limit of the interval of uncertainty is referred to as the nominal value of the parameter and the nominal RCSPP arises when all random parameters are at their nominal values. We present a robust model to find the minimum cost robust path that remains feasible when a subset of random parameters deviate from their nominal values. Next, we review the literature relevant to the deterministic RCSPP, and the relevant literature on robust optimization.

## 1.1 Literature review on the deterministic RCSPP

The solution methodologies for the deterministic RCSPP include Lagrangian relaxation, labeling algorithms, heuristics and enumerative approaches. Lagrangian relaxation is used by Handler and Zang [1980] who relax the resource constraint, resulting in a shortest path problem with Lagrangian length as subproblem. If the dual gap after solving the Lagrangian dual problem is nonzero, a $k$th-shortest path algorithm of Yen [1971] is implemented. The parameter $k$ is initialized at 2 and is increased gradually until the gap closes. Beasley and Christofides [1989] use similar relaxation and close the dual gap through branch-and-bound. The branching scheme starts at the origin and builds a partial path. Each branch corresponds to a deterministic RCSPP from the end node of the current partial path to the destination. The next branch to explore is determined based on the Lagrangian subproblem solution. Beasley and Christofides [1989] also propose graph reduction techniques based on the minimum consumption of each resource and the minimum Lagrangian length between a pair of nodes. Borndörfer et al. [2001] use a similar approach to solve the deterministic RCSPP. However, they add an additional goal consumption for each resource and penalize any deviation from the goal. Santos et al. [2007] improve the algorithm of Handler and Zang [1980] with a refined search direction based on the tightness of the resource limit.

Dynamic programming formulations leading to labeling algorithms form another exact methodology for the deterministic RCSPP. Desrochers [1988] proposes a label correcting algorithm that extends pareto optimal partial paths along the graph from origin to destination. Dumitrescu and Boland [2003] improve and compare a set of algorithms, including Lagrangian relaxation, label setting algorithm, and heuristics with a cost scaling technique. They improve the preprocessing by iteratively using the techniques in Beasley and Christofides [1989] until no more reduction is possible. For the label setting algorithm, they strengthen the feasibility check by considering for each resource the sum of the incurred resource consumption of a partial path and the minimum resource requirement from the end of a partial path to the destination. They also propose an exact weight scaling algorithm that repeatedly performs the label setting algorithm on a graph with resource consumptions on arcs and scaled resource upper bounds. During the process, the minimum cost path provides a lower bound on the original problem even if it is infeasible. As the scaled values ultimately converge to their original values, the algorithm is guaranteed to find an optimal solution. Zhu and Wilhelm [2012] propose a three stage algorithm to solve deterministic RCSPP on an acyclic graph. They reduce the graph using a technique similar to that of Dumitrescu and Boland [2003] in the first stage. Then, they expand it to a graph corresponding to an unconstrained shortest path problem, and solve as SPP using Dijkstra's algorithm. Since the extended graph from one such transformation is reusable for different arc cost, the approach is suitable for column generation where arc costs change in the process. Lozano and Medaglia [2013] use a depth first search strategy to extend labels. As a result, the number of labels stored at each node can be limited at the expense of extending more partial paths that could have been identified as dominated.

Heuristic algorithms for the deterministic RCSPP are based on cost scaling. Hassin [1992] proposes a polynomial approximation scheme for deterministic RCSPP with nonnegative integer costs on arcs and a single resource constraint. The algorithm starts from an upper bound $UB$ and a lower bound $LB$ and determines whether a resource feasible path with cost no more than $\sqrt{UB * LB}$ exists. This is done approximately by scaling arc costs and solving the scaled problem with a label setting algorithm. With approximation factor $\epsilon$, $UB$ or $LB$ is updated as $\sqrt{UB * LB}(1 + \epsilon)$ or $\sqrt{UB * LB}$ depending on whether such path exists. The algorithm is suggested to terminate when $UB/LB \leq 2$. Lorenz and Raz [2001] propose a similar scaling approach and an initialization of $UB$ that guarantees $UB/LB$ is less than or equal to the number of arcs in the graph.

Moreover, Irnich [2008] surveyed the literature in CSPP by focusing on different types of resource extension functions (REF). The aim is to identify properties of REF that may be used for reverse search in the network and may be generalized for partial paths instead of arcs. Such inversion and generalization help accelerate the solution of CSPP. An extensive survey of CSPP is provided by

Pugliese and Guerriero [2013].

## 1.2 Literature review on robust RCSPP and related problems

Robust programming deals with optimization problems where the modeling parameters are uncertain. Uncertainty is described by a set of possible realizations, called the support of the uncertainty. The optimization model finds a minimum cost solution that stays feasible under all realizations in the uncertainty support. The supports studied in the literature are convex support, such as ellipsoidal support, polyhedral support Ben-Tal and Nemirovski [1999] and cardinality constrained support Bertsimas and Sim [2003]. Cardinality constrained support, or budgeted uncertainty, corresponds to the case where each constraint/objective coefficient varies within an interval, and the total number of coefficients changing simultaneously in each constraint/objective function is limited by a constant, referred to as the protection level or budget size. Bertsimas and Sim [2003] provide an equivalent MIP reformulation for robust discrete optimization problem under such support including the robust SPP. Moreover, they prove that for robust combinatorial optimization problems with budgeted uncertainty on objective coefficients, the problem can be solved as a series of deterministic counterparts. Álvarez Miranda et al. [2013] and Goetzmann et al. [2012] generalize the result to combinatorial optimization problems with uncertainty in the constraints. The probabilistic guarantee for constraints with budgeted uncertainty is derived by Bertsimas and Sim [2004] and extended by Poss [2014a] and Poss [2014b] to variable budgeted uncertainty where the budget size is allowed to depend on decision variables. Ben-Tal et al. [2009] and Bertsimas et al. [2011] provide comprehensive reviews on robust optimization.

The literature on robust RCSPP is scant. However, at the same time when we were conducting this research, Pessoa et al. [2015] independently carried out a study of robust RCSPP and suggested a dominance rule which is different from ours. They prove that the problem is NP-hard in the strong sense when the uncertainty set is unbounded. For budgeted uncertainty sets, the robust shortest path problem with a capacity constraint can be solved pseudopolynomially. They proposed a label-setting algorithm for the case with time windows that creates as many dummy resources as the budget size for the existing time resource. The dummy resources are associated with budgeted uncertainties for which the budget sizes vary from zero to the original budget size. A partial path dominates another partial path ending at the same node when it consumes less or equal resources under all the budgeted uncertaintites.The dominance rule we propose is different in that it uses robust resource consumption up to a node $i$ and an upper bound on the maximum variation under the budgeted size from $i$ to the destination. We show in Section 3.1 that each of the dominance rules is able to eliminate dominated partial paths that the other does not. On the other hand, there are several studies on the robust shortest path problem, see for example Catanzaro et al. [2011]; Gabrel et al. [2013]; Yu and Yang [1998]; Resende [2015]; Karasan et al. [2001]. Unlike the problem treated in this paper and that of Pessoa et al. [2015], in the robust SPP there are no additional resource constraints on the paths, i.e., there is no feasibility issue and only optimality of a path is impacted by the uncertainty. In particular the dominace rule suggested by Resende [2015] extends that of the determinitic RCSPP by creating a cost label for each scenario. Similar to Pessoa et al. [2015], a partial path is dominated by another partial path ending at the same node if it has higher or equal cost under all scenarios.

Beside the studies in robust constrained shortest path problem, several studies on robust vehicle routing problems exit. Specifically, Sungur et al. [2008] are the first to study a robust capacitated vehicle routing problem (CVRP) with uncertain demands. They extend the deterministic CVRP formulation to the robust case based on the Miller-Tucker-Zemlin formulation, and focus on three types of uncertainty supports, including convex hull support, hyper-cube support and ellipsoidal support. Under these supports, the authors prove that to solve the robust CVRP it is sufficient to solve a deterministic CVRP corresponding to the worst case scenario. A robust vehicle routing problem with time windows is recently studied by Agra et al. [2013]. They provide two formulations based on resource inequalities and path inequalities, and prove that it is sufficient to consider a subset of the extreme points of the uncertainty support. When the uncertainty support is cardinality constrained, they further reduce the subset of the extreme points of the uncertainty polytope. The resource inequality formulation is solved implicitly by a column and row generation procedure and the path inequality formulation is solved by a cutting plane algorithm. The most recent work by Gounaris et al. [2013] focuses on robust CVRP with uncertain customer demands. They propose a generalized hyper-cube support that is encoded by the intersection of a hyper-rectangle with a halfspace that imposes an upper bound on total customer demand, and show that an equivalent deterministic CVRP can be derived. The robust counterparts of the two-index vehicle flow, the Miller-Tucker-Zemlin, the one-commodity flow, the two-commodity flow and the vehicle assignment formulations are provided and reformulated as MIP problems under the convex polyhedral support. Moreover, when the demand support is a set of disjoint generalized hypercube supports or a convex hull resulting from an affine transformation of a hypercube, the authors derive analytic solutions to evaluate the total demand of a subset of customers in the worst case. These analytic solutions are used to find rounded capacity inequalities and the models are solved by branch-and-cut.

The main contributions in this paper are a new dominance rule for the robust RCSPP, a modified labelling algorithm, and extension of the resource based reduction and Lagrangian cost based reduction of the graph to identify nodes/arcs that cannot be in an optimal robust path. The classical dominance rule for SPP and RCSPP is not valid because when comparing two partial paths, one partial path dominated by another one in a deterministic setting might lead to a better complete path in a robust context. This is because for each resource, the total consumption in a robust context not only depends on the deterministic data but also on a subset of the resource deviations on the complete path, where the prespecified protection level determines the size of the subset. Pessoa et al. [2015] extend the classical dominance rule by creating dummy labels. On the other hand, we propose a new dominance rule to compare two partial paths under the robust framework by developing for each resource and each partial path an upper bound on the sum of resource deviations up to the destination. To calculate these upper bounds, we devise a specialized label correcting algorithm.

We compare the modified label setting algorithm with direct solution of the MIP reformulation using CPLEX 12.4, the label setting algorithm proposed by Pessoa et al. [2015], and the sequential algorithm that solves a series of deterministic RCSPP. The results show the superiority of the modified label-setting algorithm over direct solution of the MIP reformulation and the sequential algorithm. The latter shows poor scalability when the number of resources or the number of distinct deviations increase.

The rest of the paper is organized as follows. Section 2 defines the robust RCSPP and presents the robust formulation, an equivalent MIP reformulation and the sequential algorithm. The graph reduction techniques for the robust RCSPP are presented in Section 2.2. Section 3 develops the new dominance rule and the modified label-setting algorithm. Section 4 reports on the computational testing and Section 5 concludes the paper.

# 2. The robust RCSPP

In this section, we first present the robust RCSPP formulation and an equivalent MIP formulation, and derive the sequential algorithm. In Section 2.2, we generalize the preprocessing techniques of the deterministic RCSPP to reduce the graph for robust RCSPP using resource based reduction and Lagrangian cost based reduction.

Let us define a graph $G = (N, A)$, where $N$ is the set of nodes indexed by $i = 1, ..., |N|$ and $A$ is the set of arcs represented by $(i, j) \in A$. Let $o \in N$ denote the origin, and $d \in N$ denote the destination. Traveling in the network consumes a number of resources. Each resource has a limit on available capacity, representing the maximum amount of the resource that can be used on a feasible path. The cost of an arc is represented by a special resource with infinite capacity. Let $R$ denote the set of resources indexed by $r = 1, ..., |R|$, where resource $r = 1$ is used for the cost. For each arc $(i, j) \in A$, $t_{ijr} \geq 0$ is the minimum consumption of resource $r$ on arc $(i, j)$, also referred to as nominal consumption. $h_{ijr} \geq 0$ is the maximum deviation from the nominal consumption. A realization of the consumption of resource $r$ on $(i, j)$, denoted by $\hat{t}_{ijr}$ is assumed to vary randomly in the interval $[t_{ijr}, t_{ijr} + h_{ijr}]$. When arc $(i, j)$ shows no uncertainty in the consumption of resource $r$, $h_{ijr}$ is set to 0. The limit on resource $r$ is denoted by $B_r$. The decision variable $y_{ij}$ takes value 1 if arc $(i, j)$ is selected in the shortest path, and 0 otherwise. To protect the solution from uncertainty, a prespecified parameter $\Gamma_r$, called protection level, is defined for each resource $r$. It allows at most $\Gamma_r$ arcs to deviate from the nominal values of resource $r$ for any path at any given time. Given a path $p$ involving arcs $S^p \subseteq A$, the worst deviation from the nominal consumption of resource $r$ is determined by $\max\limits_{\{S_r | S_r \subseteq S^p, |S_r| \leq \Gamma_r\}} \sum\limits_{(i,j) \in S_r} h_{ijr}$. The maximum consumption by path $p$ of resource $r$ given $\Gamma_r$ is calculated as $\sum\limits_{(i,j) \in S^p} t_{ijr} + \max\limits_{\{S_r | S_r \subseteq S^p, |S_r| \leq \Gamma_r\}} \sum\limits_{(i,j) \in S_r} h_{ijr}$ and is referred to as the robust consumption of resource $r$. As $\Gamma_r$ increases, more deviations are considered in the robust consumption of resource $r$. The robust consumption of resource $r = 1$ is defined as the robust cost. A path is considered robust if its robust consumption of resource $r$ is no more than $B_r$ for each resource $r \in R \setminus \{1\}$. The robust RCSPP is modeled as follows:

[P]

$$\min \sum_{(i,j) \in A} t_{ij1} y_{ij} + \max_{\{S_1 | S_1 \subseteq A, |S_1| \leq \Gamma_1\}} \sum_{(i,j) \in S_1} h_{ij1} y_{ij} \tag{1}$$

$$\text{s.t.} \sum_{j:(o,j) \in A} y_{o,j} = 1 \tag{2}$$

$$\sum_{i:(i,j) \in A} y_{ij} - \sum_{i:(j,i) \in A} y_{ji} = 0 \quad \forall j \in N \setminus \{o, d\} \tag{3}$$

$$\sum_{i:(i,d) \in A} y_{i,d} = 1 \tag{4}$$

$$\sum_{(i,j) \in A} t_{ijr} y_{ij} + \max_{\{S_r | S_r \subseteq A, |S_r| \leq \Gamma_r\}} \sum_{(i,j) \in S_r} h_{ijr} y_{ij} \leq B_r \quad r = 2, ..., |R| \tag{5}$$

$$y_{ij} \in \{0, 1\} \quad \forall (i, j) \in A \tag{6}$$

The objective function (1) minimizes the total robust cost given protection level $\Gamma_1$. Constraints (2)-(4) are flow balance constraints. Constraint (5) requires that the robust consumption of resource

$r$ on a feasible path must be less than or equal to resource limit $B_r$. A feasible path for robust RCSPP protects against uncertainty with a prespecified level for each resource. A shortest path for robust RCSPP is a feasible path that has the minimum robust cost under protection level $\Gamma_1$.

## 2.1 An equivalent MIP reformulation of robust RCSPP

The robust model [P] is not readily solvable because of the robust terms in the objective function and constraints. Based on Theorem 2 of Bertsimas and Sim [2003], we now present an equivalent linear MIP reformulation that may be solved by any MIP solver. Let $Y$ be the feasible set defined by flow balance constraints (2)-(4) and (6), and $\mathbf{y} \in Y$ be the vector of $y_{ij}$. For ease of exposition, we define $y_r^a$, $t_r^a$ and $h_r^a$ as replicas of $y_{ij}$, $t_{ijr}$ and $h_{ijr}$ when arc $(i, j)$ has the $a$th highest deviation with respect to resource $r$. Without loss of generality, for each resource $r$, we sort the arcs using index $a = 1, ..., |A|$ such that $h_r^1 \geq h_r^2 \geq ... \geq h_r^{|A|-1} \geq h_r^{|A|}$, and $h_r^{|A|+1} = 0$. $\mathbf{y_r}$ and $\mathbf{t_r}$ are vectors of $y_r^a$ and $t_r^a$, respectively. Define $u_r^a$ as a binary variable that takes value 1 if $h_r^a y_r^a$ is selected to maximize the robust term and 0 otherwise. Then, [P] is rewritten as follows:

[P1]

$$
\begin{aligned}
Z = \ & \min \mathbf{t}_1\mathbf{y}_1 + \max_{\substack{0 \leq u_1^a \leq 1 \\ \sum_{a=1}^{|A|} u_1^a \leq \Gamma_1}} \sum_{a=1}^{|A|} h_1^a y_1^a u_1^a \\
& \text{s.t. } \mathbf{y} \in Y \\
& \qquad \mathbf{t_r y_r} + \max_{\substack{0 \leq u_r^a \leq 1 \\ \sum_{a=1}^{|A|} u_r^a \leq \Gamma_r}} \sum_{a=1}^{|A|} h_r^a y_r^a u_r^a \leq B_r \quad r = 2, ..., |R|
\end{aligned}
$$

where $\max_{\substack{0 \leq u_r^a \leq 1 \\ \sum_{a=1}^{|A|} u_r^a \leq \Gamma_r}} \sum_{a=1}^{|A|} h_r^a y_r^a u_r^a, r = 1, ..., |R|$ are the robust terms. Let $v_r$ be the dual variable of constraint $\sum_{a=1}^{|A|} u_r^a \leq \Gamma_r$, and $q_r^a$ be the dual variable of constraint $0 \leq u_r^a \leq 1$. Applying Theorem 2 of Bertsimas and Sim [2003] to [P1], we obtain an equivalent MIP formulation:

[P-MIP]

$$
\begin{aligned}
Z = \ & \min \mathbf{t}_1\mathbf{y}_1 + \Gamma_1 v_1 + \sum_{a=1}^{|A|} q_1^a && && (7)\\
& \text{s.t. } \mathbf{y} \in Y && && \\
& \qquad \mathbf{t}_r\mathbf{y}_r + \Gamma_r v_r + \sum_{a=1}^{|A|} q_r^a \leq B_r && r = 2, ..., |R| && (8)\\
& \qquad v_r + q_r^a \geq h_r^a y_r^a && a = 1, ..., |A|, r = 1, ..., |R| && (9)\\
& \qquad q_r^a, v_r \geq 0 && a = 1, ..., |A|, r = 1, ..., |R| && (10)
\end{aligned}
$$

The robust terms in [P1] are transformed into minimization problems by duality. The objective functions of the resulting dual problems are captured in the objective function of [P-MIP] and constraints (8), respectively. The constraints of the resulting dual minimization problems are added as constraint set (9) to [P-MIP]. As a result, the model has a total of $|N|+|A||R|+|R|-1$ constraints, $|A|$ binary variables and $|A||R|$ continuous variables. Although [P-MIP] may be solved directly using any deterministic MIP solver, this proved to be inefficient as shown in Section 4. An alternative approach is to further transform the problem into a sequence of deterministic RCSPP following the transformation in Poss [2014a] and Poss [2014b].

This sequential approach is detailed in Algorithm 6 in Appendix A. It creates a sequence of RCSPP and solves the resulting deterministic problems using a label setting algorithm. For a problem with $|R|$ resources and $n_r$ distinct deviations that are less than or equal to $h_r^{\Gamma_r}$, i.e. $n_r = |\{e | h_r^e \neq h_r^{e'}, e \neq e', e = \Gamma_r, ..., |A|+1, e' = \Gamma_r, ..., |A|+1\}|$, in each resource $r \in R$, the total number of problems to solve in the worst case is $\prod_{r=1}^{|R|} n_r$. Algorithm 6 grows fast when $|R|$ or $n_r$ increases.

## 2.2 Reduction of graph $G$

It is known in the literature that reducing the graph on which RCSPP is defined by eliminating arcs and nodes that cannot be in an optimal solution helps solve larger instances. Beasley and Christofides [1989] present two types of graph reduction techniques for RCSPP. One of the techniques determines whether a node/arc should be removed based on the minimum resource consumption required to traverse that node/arc. The other technique relaxes the resource constraints in a Lagrangian fashion. Then, for each node/arc, it finds a lower bound on the cost of paths traversing that node/arc. The lower bound is compared against an upper bound to determine whether the node/arc may be removed. When ignoring uncertainty, the cost and resource consumption happen at the nominal level and the nominal problem is a deterministic RCSPP given by:

[PR1]

$$\min \sum_{(i,j)\in A} t_{ij1} y_{ij} \tag{11}$$
$$\text{s.t. } (2) - (4), (6)$$
$$\sum_{(i,j)\in A} t_{ijr} y_{ij} \leq B_r \quad r = 2, ..., |R| \tag{12}$$

Since constraints (5) are more restricting than constraints (12), and the robust term in (1) is nonegative, the optimal objective value of [PR1] provides a lower bound to [P]. The lower bound may be further improved by deriving bounds on the robust resource consumption. Define the shortest length $l$ as the minimum number of arcs needed to go from $o$ to $d$ in graph $G$ without enforcing resource limits. Let $\hat{\Gamma}_r = \min\{l, \Gamma_r\}$. For any path $p$, the robust consumption of resource $r$ is at least $\sum_{(i,j)\in S^p} t_{ijr} + H_r$, where $H_r = \min_{\{S_r | S_r \subseteq A, |S_r| = \hat{\Gamma}_r\}} \sum_{(i,j)\in S_r} h_{ijr}$ is a constant for resource $r$. Subtracting $H_r$ from the corresponding resource limit $B_r$ leads to the following modified RCSPP:

[PR2]

$$\min \sum_{(i,j)\in A} t_{ij1} y_{ij} + H_1 \tag{13}$$
$$\text{s.t. } (2) - (4), (6)$$
$$\sum_{(i,j)\in A} t_{ijr} y_{ij} \leq B_r - H_r \quad r = 2, ..., |R| \tag{14}$$

Since

$$H_r \leq \max_{\{S_r | S_r \subseteq A, |S_r| \leq \Gamma_r\}} \sum_{(i,j)\in S_r} h_{ijr} y_{ij}, \quad r = 1, ..., |R|$$

any feasible path to [P] is also feasible to [PR2] and its objective value in [P] is no less than its objective value in [PR2]. Using these results, we derive two sets of reduction rules for graph $G$. The first set of rules is based on resource capacity constraints, and the second is based on Lagrangian bounds.

### 2.2.1 Resource based reduction

Removing the resource constraints (14) from [PR2], and the constant term $H_1$ from the objective function, we obtain a deterministic shortest path problem [SPP] as follows:

[SPP]

$$\min \sum_{(i,j)\in A} t_{ij1} y_{ij} \tag{15}$$
$$\text{s.t. (2) - (4), (6).}$$

Define $D^r_{ij}$ as the minimum cost from node $i$ to node $j$ with consumption of resource $r$ as the arc cost. Then, node $i$ satisfying the following inequality may be removed from graph $G$:

$$D^r_{oi} + D^r_{id} > B_r - H_r \quad r = 2, ..., |R|. \tag{16}$$

In inequality (16), if the minimum consumption of resource $r$ from $o$ to $d$ through node $i$ exceeds $B_r - H_r$, any path visiting node $i$ is infeasible to [P]. Moreover, arc $(i, j)$ satisfying the following inequality may be removed from graph $G$:

$$D^r_{oi} + t_{ijr} + D^r_{jd} > B_r - H_r \quad r = 2, ..., |R|. \tag{17}$$

Inequality (17) states that if the minimum consumption of resource $r$ from $o$ to $d$ through arc $(i, j)$ is greater than $B_r - H_r$, any path traversing arc $(i, j)$ is infeasible to [P]. The minimum cost from origin to all nodes in graph $G$ may be calculated using a label setting algorithm. Similarly, the minimum cost from all nodes to destination may be calculated on the reversed graph of $G$. Algorithm 3 in Appendix A summarizes the steps of resource based reduction.

### 2.2.2 Lagrangian cost based reduction

Introducing Lagrangian multipliers $\mu_r \geq 0, r = 2, ..., |R|$ to resource constraints (14), the relaxed problem of [PR2] is:

[PR3]

$$Z^{LR} = \min \sum_{(i,j)\in A} t_{ij1} y_{ij} + H_1 + \sum_{r=2}^{|R|} \mu_r \left( \sum_{(i,j)\in A} t_{ijr} y_{ij} - B_r + H_r \right) \tag{18}$$
$$\text{s.t. (2) - (4), (6)} \qquad .$$

If the solution $\bar{y}_{ij}$ of [PR3] is feasible for [P], it gives an upper bound

$Z^{UB} = \sum_{(i,j)\in A} t_{ij1} \bar{y}_{ij} + \max_{\{S_1|S_1\subseteq A, |S_1|\leq\Gamma_1\}} \sum_{(i,j)\in S_1} h_{ij1} \bar{y}_{ij}$ to the optimal objective value of the original problem [P].

The Lagrangian cost for arc $(i, j) \in A$ is given by $\bar{t}_{ij1} = t_{ij1} + \sum_{r=2}^{|R|} \mu_r t_{ijr}$. The minimum Lagrangian cost over the set of paths $P(i, j)$ from node $i$ to node $j$ is given by $L_{ij}(\mu) = \min_{p\in P(i,j)} \sum_{(i',j')\in S^p} \bar{t}_{i'j'1}$

for a given set of multipliers $\mu_r, r = 2, ..., |R|$. Any node $i$ satisfying the following inequality may be removed from graph $G$:

$$L_{oi}(\mu) + L_{id}(\mu) + H_1 - \sum_{r=2}^{|R|} \mu_r (B_r - H_r) > Z^{UB} \tag{19}$$

The left hand side of inequality (19) calculates the minimum objective value measured by function (18) among all paths from $o$ to $d$ through node $i$. If it is greater than the objective value of the current best solution to [P], any path traversing node $i$ is not optimal to [P]. An arc $(i, j) \in A$ may be removed from $G$ if the following inequality is satisfied:

$$L_{oi}(\mu) + \bar{t}_{ij1} + L_{jd}(\mu) + H_1 - \sum_{r=2}^{|R|} \mu_r (B_r - H_r) > Z^{UB} \tag{20}$$

In inequality (20), if the minimum objective value given by function (18) among all paths from $o$ to $d$ through arc $(i, j)$ is greater than the upper bound on [P], an optimal solution to [P] does not involve arc $(i, j)$. To determine multipliers $\mu_r, r = 2, ..., |R|$, we use Kelley's cutting plane algorithm (Kelley [1960]) to solve the Lagrangian dual problem of [PR3]. The reduction based on Lagrangian cost is summarized in Algorithm 4 in Appendix A.

The overall graph reduction procedure is detailed in Algorithm 5 in Appendix A. It starts with resource based reduction (Algorithm 3). Paths found during resource based reduction are used to construct cuts to initialize the Lagragian master problem. Then, Kelley's cutting plane algorithm is used to solve the Lagrangian master problem of [PR3]. Optimal Lagrangian multipliers $\mu_r, r = 2, ..., |R|$ are used to calculate the Lagrangian cost for each arc, and Lagrangian based reduction (Algorithm 4) is applied. During the procedure, any path feasible to [P] is used to update $Z^{UB}$. If $l$ increases or set $A$ reduces in a reduced graph $G$, $H_r$ may increase to make constraint (14) tighter. The resulting [PR2] may have an increased optimal objective value that in turn may further reduce graph $G$. Hence, resource based reduction and Lagrangian cost based reduction are applied iteratively until the graph can not be reduced any more.

# 3. Label setting algorithm

One important component of label setting algorithm for the deterministic RCSPP is the dominance rule that avoids extending unpromising partial paths. The dominance rule is described as follows. Let $c_{ir}^p = \sum_{(j,k) \in S^p} t_{jkr}$ be the nominal consumption of resource $r$ for path $p$ from origin $o$ to node $i$. A label associated with node $i$ is denoted by vector $C_i^p = [c_{i1}^p, ..., c_{i|R|}^p]$. For two labels at node $i$,

$C_i^1$ dominates $C_i^2$, represented by $C_i^1 < C_i^2$, if there exists resource $\overline{r} \in R$ such that $c_{i\overline{r}}^1 < c_{i\overline{r}}^2$ and $c_{ir}^1 \leq c_{ir}^2$ for $r \in R\backslash\{\bar{r}\}$. The algorithm starts at the origin and proceeds by extending nondominated partial paths. A pseudo-code of the algorithm is given in Algorithm 1.

We now present an example to show that this dominance rule is invalid for the robust RCSPP. Then, we provide a new dominance rule that considers not only the deviations on the arcs of a partial path but also on the possible arcs that may be appended after. Such consideration guarantees the dominance relation as it uses an upper bound on the maximum deviation of a complete path extended from the current partial path. To calculate the upper bounds, we devise a label correcting algorithm that calculates the $\Gamma_r$ maximum deviations relevant to each node. Lastly, we present the complexity of the complete label setting algorithm, discuss the distinction from the dominance rule proposed by Pessoa et al. [2015], and use examples to show that each rule may identify dominated paths that the other fails to.

Define $\lambda_{ir}^p = c_{ir}^p + \max_{\{S:|S|\leq\Gamma_r, S\subseteq S^p\}} \sum_{(j,k)\in S} h_{jkr}$ as the robust consumption of resource $r$ for path $p$ from origin $o$ to node $i$, $\Lambda_i^p = [\lambda_{i1}^p, ..., \lambda_{i|R|}^p]$ as the resource consumption vector of robust RCSPP. Figure 1 shows a network of RCSPP with 2 resources. The origin and destination in Figure 1 are nodes 1 and 4, respectively. On each arc, the first and the second numbers between parentheses represent the consumption of resources 1 and 2 on that arc. The limit on resource 2 is 20. There are two paths 1 and 2 from origin to destination consisting of arcs $\{(1,3),(3,4)\}$ and $\{(1,2),(2,3),(3,4)\}$ with labels $C_4^1 = [7,9]$ and $C_4^2 = [8,9]$, respectively. Path $\{(1,3),(3,4)\}$ is the optimal path. Moreover, as $C_3^1$ dominates $C_3^2$, discarding $C_3^2$ at node 3 does not affect optimality. Figure 2 adds deviations to the graph in Figure 1. A pair of numbers in curly brackets shows the deviations of resources 1 and 2 on each arc. Setting the protection level $\Gamma$ to 2 for both resources, then $\Lambda_3^1 = [12,11]$ and $\Lambda_3^2 = [17,14]$. Based on the dominance rule for RCSPP, $\Lambda_3^2$ is dominated by $\Lambda_3^1$, only one path to node 4 remains with label $\Lambda_4^1 = [22,19]$. However, the optimal path is path 2 with $\Lambda_4^2 = [21,18]$. This shows that the dominance rule for RCSPP is invalid. The reason is that when reaching node 3, path 2 has already encountered the first two biggest deviationss for each resource, whereas path 1 only encountered one of its two deviations for each resource. Therefore, to derive a valid dominance rule, we need not only consider the information given by $\Lambda_i^p$ but also the information from node $i$ to the destination $d$.

**Algorithm 1** Label setting algorithm for deterministic RCSPP

Definition:
$P_i$: the set of non-dominated labels at node $i$
$Unprocessed$: keeps the labels that are created but not processed yet.
$W(i) = \{j|(i,j) \in A\}$: the set of nodes that connects node $i$ with an arc $(i,j) \in A$.
REF($C_i^{\bar{p}}$,$j$): the nondecreasing resource extension function that calculates the resource consumption at node $j$ when $C_i^p$ is extended through arc $(i,j)$.
$Feasible(C_j^p)$: returns true if $c_{i1}^p \leq UB$ and $c_{ir}^p \leq B_r, r = 2, ..., |R|$ and false otherwise.
$Dominance(C_i^p, P_i)$: returns true if $C_i^p$ is not dominated by the labels in $P_i$ and false otherwise
Initialization:
$UB =$ the objective value of best feasible path obtained during preprocessing or from heuristic.
$p = 0$, $C_o^p = [0, ..., 0] \in \mathbb{R}^{|R|}$.
Mark $C_o^p$ as non-dominated.
$P_o \leftarrow \{C_o^p\}, Unprocessed \leftarrow \{C_o^p\}$.
**for all** $i \in N \backslash \{o\}$ **do**
    $P_i \leftarrow \{(\infty, B_2, ..., B_{|R|})\}$.
**end for**
**while** $Unprocessed \neq \varnothing$ **do**
    Extract $C_i^{\bar{p}} \in Unprocessed$
    **if** $C_i^{\bar{p}}$ is not dominated **then**
        **for all** $j \in W(i)$ **do**
            $p = p + 1$.
            $C_j^p$ =REF($C_i^{\bar{p}}$,$j$).
            **if** $Feasible(C_j^p)$ indicates $C_j^p$ is feasible **then**
                **if** $Dominance(C_j^p, P_j)$ indicate $C_j^p$ is not dominated **then**
                    $P_j \leftarrow P_j \bigcup \{C_j^p\}$
                    **if** $j = d$ and $c_{j1} < UB$ **then**
                        Update $UB$ and the current best path.
                    **else**
                        $Unprocessed \leftarrow Unprocessed \bigcup \{C_j^p\}$
                    **end if**
                **end if**
            **end if**
        **end for**
        Mark $C_i^{\bar{p}}$ as processed.
        $Unprocessed \leftarrow Unprocessed \backslash \{C_i^{\bar{p}}\}$
    **end if**
**end while**

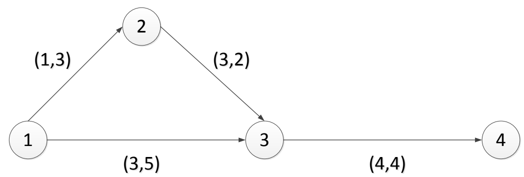


Figure 1: A network for deterministic resource constrained shortest path problem.

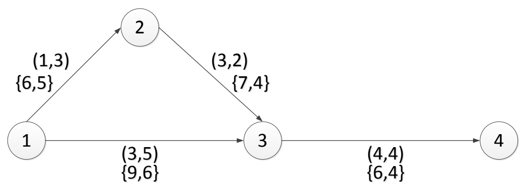


Figure 2: A network for robust resource constrained shortest path problem.

## 3.1 Label setting algorithm for robust RCSPP

Let $G_i = (N_i, A_i)$ denote a subgraph of $G$ rooted at $i$, i.e., $N_i$ is the set of nodes reachable from node $i$, and $A_i$ is the set of arcs $\{(j,k) \in A : j,k \in N_i\}$. $\Delta_{ir}$ is the set of arcs with the highest $\Gamma_r$ deviations across $G_i$ when $\Gamma_r < |A_i|$, or are the $|A_i|$ arcs when $\Gamma_r \geq |A_i|$:

$$\Delta_{ir} = \underset{S_{ir}:S_{ir}\subseteq A_i, |S_{ir}|\leq \Gamma_r}{\arg\max} \sum_{(j,k)\in S_{ir}} h_{jkr}, r \in R, i \in N.$$

To determine $\Delta_{ir}$, we suggest the following modified label correcting algorithm (MLCA) as shown in Algorithm 7. Let $\overline{G} = (N, \overline{A})$ be the reverse graph of $G$ and associate with arc $(j,i) \in \overline{A}$ weight $h_{ijr}$. Note that for every arc $(j,i) \in \overline{A}$ and every resource $r$, the highest $\Gamma_r$ arc weights of $G_j$ cannot all be strictly higher than those of $G_i$, since by the existence of arc $(i,j) \in A$, $A_j \subseteq A_i$. Associate with each node $i$ and resource $r$, a label $V(ir)$ which stores at most $\Gamma_r$ elements. Each element corresponds to an arc $(k,j) \in \bar{A}$ and is formatted as $f = \{h_{jkr}, (k,j)\}$. The set of arcs involved in $V(ir)$ is denoted by $A(V(ir))$. For arc $(j,i) \in \overline{A}$, let $hmin_i = \min\{h_{jkr} : (k,j) \in A(V(ir))\}$ when $|A(V(ir))| = \Gamma_r$, and 0 otherwise; and let $hmax_j = \max\{h_{kk'r} | (k,k') \in A(V(jr)) \backslash A(V(ir))\}$ when $A(V(jr)) \backslash A(V(ir)) \neq \emptyset$, and 0 otherwise. Then, $hmin_i \geq hmax_j, \forall (j,i) \in \overline{A}$ is an optimality condition for $V(ir)$. The label correcting algorithm works on the reverse graph $\overline{G} = (N, \overline{A})$ once for each resource $r \in R$ and starts by initilaizing the label $V(ir) = [\{0, (0,0)\}]$, $i \in N$. For every arc $(j,i) \in \overline{A}$ such that $hmin_i < hmax_j, V(ir)$ does not include the arcs with the highest deviations and is updated as $\left[\{h_{kk'r}, (k',k)\} : (k',k) \in A', A' = \underset{\tilde{A}\subseteq A(V(ir))\bigcup A(V(jr))}{\arg\max} \sum_{(k',k)\in \tilde{A}} h_{kk'r}\right]$. The algorithm stops when there are no arcs that violate the optimality condition. The finiteness of the algorithm follows from that of the label correcting algorithm for SPP. The main differences are in that the arc weights are only used once to initialize the labels, and node labels are of size $\Gamma_r$ for each resource $r$.

For a path $p$ from origin $o$ to node $i$ composed of arcs $S^p$, let us define $S_r^p$ as a subset of $S^p$ for each resource $r \in R$, and

$$\Phi_{ir}^p = \operatorname*{arg\,max}_{S_r^p : S_r^p \subseteq S^p, |S_r^p| \leq \Gamma_r} \sum_{(j,k) \in S_r^p} h_{jkr}$$

as the set of arcs such that the sum of their deviations of resource $r$ is maximized under protection level $\Gamma_r$. Then, for $r \in R$, an upper bound on the total deviation of consumption of resource $r$ under protection level $\Gamma_r$ for a path to destination $d$ extended from $p$ is

$$\xi_{ir}^p = \max_{\{\Upsilon \subseteq \Phi_{ir}^p \cup \Delta_{ir}, |\Upsilon| \leq \Gamma_r\}} \sum_{(j,k) \in \Upsilon} h_{jkr},$$

where $\Phi_{ir}^p \cup \Delta_{ir}$ consists of a subset of arcs from path $p$ and a subset of arcs from $A_i$. $\Xi_i^p = [\xi_{i1}^p, ..., \xi_{i|R|}^p]$ represents a vector that consists of the upper bound on the total deviation of each resource consumption for a path to destination $d$ extended from $p$.

**Theorem 1.** *Let $< p_a, p_b >$ denote the concatenation of paths $p_a$ and $p_b$, where the ending node of $p_a$ is the starting node of $p_b$. Given partial paths $p_1$ and $p_2$ both from origin $o$ to node $i$, path $p_3$ from node $i$ to the destination $d$, if $C_i^1 + \Xi_i^1 < \Lambda_i^2$, path $p_4 =< p_1, p_3 >$ dominates path $p_5 =< p_2, p_3 >$. Label $C_i^2$ may be eliminated at node $i$.*

*Proof of Theorem 1.* For path 1 and $\forall r \in R$,

$$\begin{aligned}
\xi_{ir}^1 &= \max_{\{\Upsilon \subseteq \Phi_{ir}^p \cup \Delta_{ir}, |\Upsilon| \leq \Gamma_r\}} \sum_{(j,k) \in \Upsilon} h_{jkr} \\
&= \max_{\{\Upsilon | \Upsilon \subseteq S^1 \cup A_i, |\Upsilon| \leq \Gamma_r\}} \sum_{(j,k) \in \Upsilon} h_{jkr} && (21) \\
&\geq \max_{\{S_r^4 | S_r^4 \subseteq S^1 \cup S^3, |S_r^4| \leq \Gamma_r\}} \sum_{(j,k) \in S_r^4} h_{jkr} && (22) \\
&= \max_{\{S_r^4 | S_r^4 \subseteq S^4, |S_r^4| \leq \Gamma_r\}} \sum_{(j,k) \in S_r^4} h_{jkr} && (23)
\end{aligned}$$

Equality (21) holds due to the definition of $\Phi_{ir}^p$ and $\Delta_{ir}$. Inequality (22) holds because $S^3 \subseteq A_i$. $\xi_{ir}^1 \geq \max_{\{S_r^4 | S_r^4 \subseteq S^4, |S_r^4| \leq \Gamma_r\}} \sum_{(j,k) \in S_r^4} h_{jkr}$ leads to

$$c_{ir}^1 + \max_{\{S_r^4 | S_r^4 \subseteq S^4, |S_r^4| \leq \Gamma_r\}} \sum_{(j,k) \in S_r^4} h_{jkr} \leq c_{ir}^1 + \xi_{ir}^1 \tag{24}$$

Recall that $\lambda_{ir}^p = c_{ir}^p + \max_{\{S_r^p | S_r^p \subseteq S^p, |S_r^p| \leq \Gamma_r\}} \sum_{(i,j)\in S_r^p} h_{jkr}$, then

$$\lambda_{ir}^2 \leq c_{ir}^2 + \max_{\{S_r^5 | S_r^5 \subseteq S^2 \cup S^3 = S^5, |S_r^5| \leq \Gamma_r\}} \sum_{(j,k)\in S_r^5} h_{jkr} \quad r = 1, ..., |R| \tag{25}$$

If $c_{i\bar{r}}^1 + \xi_{i\bar{r}}^1 < \lambda_{i\bar{r}}^2$ for resource $\bar{r} \in R$, $C_i^1 + \Xi_i^1 < \Lambda_i^2$ leads to the following set of inequalities:

$$c_{ir}^1 + \xi_{ir}^1 \leq \lambda_{ir}^2 \leq c_{ir}^2 + \max_{\{S_r^5 | S_r^5 \subseteq S^5, |S_r^5| \leq \Gamma_r\}} \sum_{(j,k)\in S_r^5} h_{jkr} \quad \forall r \in R \backslash \{\bar{r}\} \tag{26}$$

$$c_{i\bar{r}}^1 + \xi_{i\bar{r}}^1 < \lambda_{i\bar{r}}^2 \leq c_{i\bar{r}}^2 + \max_{\{S_{\bar{r}}^5 | S_{\bar{r}}^5 \subseteq S^5, |S_{\bar{r}}^5| \leq \Gamma_{\bar{r}}\}} \sum_{(j,k)\in S_{\bar{r}}^5} h_{jk\bar{r}} \tag{27}$$

Hence,

$$c_{ir}^1 + \max_{\{S_r^4 | S_r^4 \subseteq S^4, |S_r^4| \leq \Gamma_r\}} \sum_{(j,k)\in S_r^4} h_{jkr} \leq c_{ir}^2 + \max_{\{S_r^5 | S_r^5 \subseteq S^5, |S_r^5| \leq \Gamma_r\}} \sum_{(i,j)\in S_r^5} h_{jkr} \quad \forall r \in R \backslash \{\bar{r}\} \tag{28}$$

$$c_{i\bar{r}}^1 + \max_{\left\{S_{\bar{r}}^4 | S_{\bar{r}}^4 \subseteq S^4, |S_{\bar{r}}^4| \leq \Gamma_{\bar{r}}\right\}} \sum_{(j,k)\in S_{\bar{r}}^4} h_{jk\bar{r}} < c_{i\bar{r}}^2 + \max_{\{S_{\bar{r}}^5 | S_{\bar{r}}^5 \subseteq S^5, |S_{\bar{r}}^5| \leq \Gamma_{\bar{r}}\}} \sum_{(i,j)\in S_{\bar{r}}^5} h_{jk\bar{r}} \tag{29}$$

Moreover, as the nominal consumption of resource $r \in R$ on path 3 is $c_{dr}^4 - c_{ir}^1$, which equals $c_{dr}^5 - c_{ir}^2$, adding $c_{dr}^4 - c_{ir}^1$ and $c_{dr}^5 - c_{ir}^2$ to the left and right hand sides of inequalities (28) and (29) results in

$$c_{dr}^4 + \max_{\{S_r^4 | S_r^4 \subseteq S^4, |S_r^4| \leq \Gamma_r\}} \sum_{(j,k)\in S_r^4} h_{jkr} = \lambda_{dr}^4 \leq c_{dr}^5 + \max_{\{S_r^5 | S_r^5 \subseteq S^5, |S_r^5| \leq \Gamma_r\}} \sum_{(j,k)\in S_r^5} h_{jkr} = \lambda_{dr}^5 \quad \forall r \in R \backslash \{\bar{r}\} \tag{30}$$

$$c_{d\bar{r}}^4 + \max_{\left\{S_{\bar{r}}^4 | S_{\bar{r}}^4 \subseteq S^4, |S_{\bar{r}}^4| \leq \Gamma_{\bar{r}}\right\}} \sum_{(j,k)\in S_{\bar{r}}^4} h_{jk\bar{r}} = \lambda_{d\bar{r}}^4 < c_{d\bar{r}}^5 + \max_{\{S_{\bar{r}}^5 | S_{\bar{r}}^5 \subseteq S^5, |S_{\bar{r}}^5| \leq \Gamma_{\bar{r}}\}} \sum_{(j,k)\in S_{\bar{r}}^5} h_{jk\bar{r}} = \lambda_{d\bar{r}}^5 \tag{31}$$

Hence $\Lambda_d^4 < \Lambda_d^5$. As path 3 can be any path from $i$ to $d$, $C_i^1$ dominates $C_i^2$. $\square$

The dominance rule in Theorem 1 says that partial path $p_1$ dominates partial $p_2$ if it consumes less or equally of every resource $r \in R$, and strictly less for at least one resource, where $p_1$'s consumption includes the highest $\Gamma_r$ arc deviations up to the destination, while that of $p_2$ does not. For the instance in Figure 2, partial path $\{(1,3)\}$ does not dominate partial path $\{(1,2),(2,3)\}$ because by considering the cost deviations 4 and 5 on arcs $(1,3)$ and $(3,4)$, the cost of partial path $\{(1,3)\}$ is higher than the nominal cost of the other partial path. Similarly, partial path $\{(1,2),(2,3)\}$ does not dominate $\{(1,3)\}$ for the same reason. The modified label-setting algorithm equipped with the new dominance rule is detailed in Algorithm 2.

The dominance rule by Pessoa et al. [2015] creates $\Gamma_r + 1$ dummy resources for each existing resource and uses a dominance rule as follows. Let $\lambda_{ir}^p(\gamma) = c_{ir}^p + \max_{\{S:|S|\leq\gamma, S\subseteq A^p\}} \sum_{(i,j)\in S} h_{ijr}$ be the

**Algorithm 2** MLS: Modified label setting algorithm for robust RCSPP

Definition:
$V_{ir}^p$: a sorted vector containing $\Gamma_r$ variations of resource $r \in R$ in descending order associated with path $p$ ending at node $i$.
$\mathbf{V}_i^p$: a list consists of $V_{ir}^p$, $r \in R$.
Initialization:
$UB$ = the objective value of the best feasible path obtained during preprocessing or from heuristic.
$p = 0$, $C_o^p = (0, ..., 0) \in \mathbb{R}^{|R|}$.
Mark $C_o^p$ as non-dominated.
Associate $C_i^p$ with a vector $\mathbf{V}_i^p = \varnothing$.
**for all** $r \in R$ **do**
    $\mathbf{V}_i^p \leftarrow \mathbf{V}_i^p \bigcup \{V_{ir}^p\}$, where $V_{ir}^p = [0, ..., 0] \in \mathbb{R}^{\Gamma_r}$.
**end for**
$P_o \leftarrow \{C_o^p\}$, $Unprocessed \leftarrow \{C_o^p\}$.
**for all** $i \in N \backslash \{o\}$ **do**
    $P_i \leftarrow \{(\infty, B_2, ..., B_{|R|})\}$.
**end for**
**while** $Unprocessed \neq \varnothing$ **do**
    Extract $C_i^{\bar{p}} \in Unprocessed$.
    **if** $C_i^{\bar{p}}$ is not dominated **then**
        **for all** $j \in W(i)$ **do**
            $p = p + 1$.
            Create a vector $\mathbf{V}_j^p = \{V_{1r}^{\bar{p}}, ..., V_{i|R|}^{\bar{p}}\}$.
            $C_j^p = Robust\_REF(C_i^{\bar{p}}, j, \mathbf{V}_j^p)$.
            **if** $Robust\_Feasible(C_j^p, \mathbf{V}_j^p)$ indicates $C_j^p$ is feasible, **then**
                **if** $Robust\_Dominance(C_j^p, P_j, \mathbf{V}_j^p)$ indicates $C_j^p$ is not dominated, **then**
                    Mark $C_j^p$ as non-dominated.
                    $P_j \leftarrow P_j \bigcup \{C_j^p\}$.
                    $Unprocessed \leftarrow Unprocessed \bigcup \{C_j^p\}$.
                **end if**
            **end if**
        **end for**
    **end if**
    Mark $C_i^{\bar{p}}$ as processed.
    $Unprocessed \leftarrow Unprocessed \backslash \{C_i^{\bar{p}}\}$.
**end while**

```
function Robust_REF(C_i^p̄, j, V_j^p)
    Create C_j^p.
    for all r ∈ R do
        c_jr^p = c_ir^p̄ + t_ijr.
        for all g ∈ V_jr^p do
            if h_ijr > g then
                V_jr^p ← V_jr^p ⋃{h_ijr}.
                if |V_jr^p| > Γ_r then
                    Find the minimum element g_min in V_jr^p.
                    V_jr^p ← V_jr^p \{g_min}.
                end if
                Break.
            end if
        end for
    end for
    Return C_j^p.
end function
```

```
function Robust_Feasible(C_i^p, V_i^p)
    var = sum of all elements in V_i1^p.
    if c_i1^p + var ≥ UB then
        Return false.
    end if
    for all r ∈ R\{1} do
        var = sum of all elements in V_ir^p
        if c_ir^p + var > B_r then
            Return false.
        end if
    end for
    Return true.
end function
```

**function** *Robust_Dominance*($C_i^p$, $\mathbf{V}_i^p$, $P_i$)
  **for all** $C_i^{\bar{p}} \in P_i$ **do**
    **if** $C_i^{\bar{p}} + \Xi_i^{\bar{p}} < \Lambda_i^p$ **then**
      Mark $C_i^p$ as dominated.
      Return false.
    **else if** $C_i^p + \Xi_i^p < \Lambda_i^{\bar{p}}$ **then**
      Mark $C_j^{\bar{p}}$ as dominated.
      Delete $\mathbf{V_i^{\bar{p}}}$.
      **if** $C_i^{\bar{p}}$ is processed **then**
        $P_i \leftarrow P_i \backslash \{C_i^{\bar{p}}\}$
      **end if**
    **end if**
  **end for**
  Return true.
**end function**

robust resource consumption $r$ for path $p$ from origin $o$ to node $i$ under a budgeted uncertainty with a protection level of $\gamma$, $\tilde{\Lambda}_i^p = [\lambda_{ir}^p(\gamma) : r = 1, ..., |R|, \gamma = 0, ..., \Gamma_r]$ as the resource consumption vector. Given paths $p_1$ and $p_2$ both from origin $o$ to node $i$, if $\lambda_{ir}^1(\gamma) \leq \lambda_{ir}^2(\gamma), r = 1, ..., |R|, \gamma = 0, ..., \Gamma_r$ and at least one inequality is strict, then path $p_2$ is dominated by path $p_1$.

The dominance rules are different and each may identify dominated paths that are missed by the other. To illustrate consider the two graphs in Figures 3 and 4, with $|R| = 2$, $\Gamma_1 = 0$ and $\Gamma_2 = 2$, i.e. the cost resource is not subject to uncertainty. In both graphs, let path $p_1$ be $\{(1, 2), (2, 4)\}$, path $p_2$ be $\{(1, 3), (3, 4)\}$, and path $p_3$ be $\{(4, 5)\}$. In Figure 3, according to the dominance rule by Pessoa et al. [2015], the labels at node 4 are $\tilde{\Lambda}_4^1 = [3, 6, 12.5, 13]$ and $\tilde{\Lambda}_4^2 = [3, 9, 12, 15]$. None of the labels is dominated. However, our dominance rule compares $C_4^1 + \Xi_3^1 = [3, 13.5]$ with $\Lambda_4^2 = [3, 15]$, resulting in path $p_1$ dominating path $p_2$. In Figure 4, the dominance rules by Pessoa et al. [2015] compares $\tilde{\Lambda}_4^1 = [3, 6, 11, 14]$ and $\tilde{\Lambda}_4^2 = [3, 9, 12, 15]$, resulting in $p_1$ dominating $p_2$. On the other hand, our dominance rule compares $C_4^1 + \Xi_4^1 = [3, 16]$ with $\Lambda_4^2 = [3, 15]$ and $C_4^2 + \Xi_4^2 = [3, 17]$ with $\Lambda_4^1 = [3, 14]$, and none of the paths is dominated. The example shows that both dominance rules may successfully identify dominated paths that the other fails to.

The complexity of Algorithm 1 is proved to be $O(|A|B^{|R|})$ for deterministic RCSPP when there is at least one resource with positive consumption for all arcs and appropriate data structure is implemented (Desrochers [1988]). The proposed label setting algorithm, Algorithm 2, implements once the label correcting algorithm for each resource. Compared to the label correcting algorithm for SPP, updating the labels takes time $O(\Gamma_r)$ and the label correcting algorithm uses time

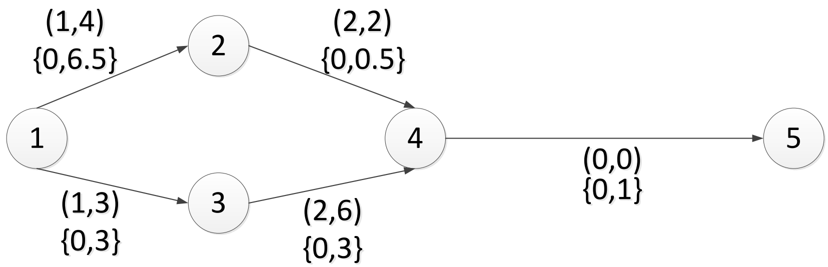


Figure 3: A network where dominance rule in Theorem 1 identifies path $p_1$ dominating path $p_2$.

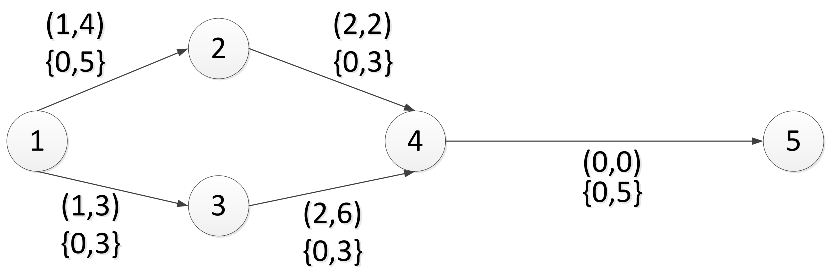


Figure 4: A network where dominance rule of Pessoa et al. [2015] identifies path $p_1$ dominating path $p_2$.

$O(\sum_{r\in R}\Gamma_r(|A||N|))$. Algorithm 2 differs from Algorithm 1 in updating a sorted vector $V^p_{jr}$ in function $Robust\_REF(C^p_j, \mathbf{V}^p_j)$ for each resource $r = 1, ..., |R|$. Updating $V^p_{jr}$ takes time $O(\Gamma_r)$. Therefore, Algorithm 2 runs in time $O((1 + \sum_{r\in R}\Gamma_r)|A|B^{|R|})$. When $R = 1$, Algorithm 2 time is $O(\Gamma|A|B)$ while that of Pessoa et al. [2015] is $O(\Gamma|A|B^{\Gamma+1})$.

# 4. Numerical testing

In this section, we report on extensive numerical testing to compare the modified label-setting algorithm (**MLS**),i.e. Algorithm 2, with the solution of the MIP reformulation [P-MIP] directly as a linear mixed integer program, and the sequential algorithm. Before any of the approaches is used, we first reduce the problem using the graph reduction algorithm of Section 2.1. Then, each of the three approaches is used to solve the reduced instances. We also compare (**MLS**) to the algorithm of Pessoa et al. [2015] denoted by (**PPGP**). All algorithms are coded in C++, and MIP and LP models are solved using CPLEX 12.6.1. Testing is carried out on a workstation with Xeon processor and 8GB RAM.

We perform testing using two datasets. The first set consists of 60 random instances, denoted by **LG**, based on 24 instances from Beasley and Christofides [1989], denoted by **BC** as shown in Table 1. For example, **LG** instances 7, 13, 19 and 25 are based on **BC** instance 5. **BC** instances $1-4, 9-12, 17-20$ have one resource, while instances $5-8, 13-6, 21-24$ have 10 resources, in addition to cost. **BC** instances $1, 2, 5, 6, 9, 10, 13, 14, 17, 18, 21$ are generated and 22 using the scheme of Handler and Zang [1980]. This scheme is designed such that the optimal path has a low ranking when the unconstrained paths are ordered from lowest cost to highest cost. Specifically, the nodes in these instances are randomly generated on a square. The cost on arc $(i, j)$ is an integer based on the Euclidean distance between nodes $i$ and $j$. Resource consumption on an arc is determined by multiplying the reciprocal of arc cost by a uniformly generated random variable. As a result, resource consumption is inversely related to arc cost. In **BC** instances $3, 4, 7, 8, 11, 12, 15, 16, 19, 20, 23$ and 24, both cost and resource consumption are uniformly generated integers in $[0, 5]$. Arc $(i, j)$ is determined by randomly generating $i$ from $[1, |N|]$ and $j$ from $[i + 1, \min(|N|, i + |N|/4)]$.

Table 1: Relationship between **BC** instances and **LG** instances.

| $\lvert R\rvert - 1$ | 1 | 1 | 10 | 2 | 3 | 4 | 10 |
|---|---|---|---|---|---|---|---|
| | **BC** | **LG** | **BC** | **LG** | | | |
| | 1 | 1 | 5 | 7 | 13 | 19 | 25 |
| | 2 | 2 | 6 | 8 | 14 | 20 | 26 |
| | 3 | 31 | 7 | 37 | 43 | 49 | 55 |
| | 4 | 32 | 8 | 38 | 44 | 50 | 56 |
| | 9 | 3 | 13 | 9 | 15 | 21 | 27 |
| | 10 | 4 | 14 | 10 | 16 | 22 | 28 |
| | 11 | 33 | 15 | 39 | 45 | 51 | 57 |
| | 12 | 34 | 16 | 40 | 46 | 52 | 58 |
| | 17 | 5 | 21 | 11 | 17 | 23 | 29 |
| | 18 | 6 | 22 | 12 | 18 | 24 | 30 |
| | 19 | 35 | 23 | 41 | 47 | 53 | 59 |
| | 20 | 36 | 24 | 42 | 48 | 54 | 60 |

In our tests, **LG** instances with $2, 3$ and 4 resources are obtained from **BC** instances with 10 resources where the first $2, 3$ and 4 resources are considered.

The second set of instances is based on the 210 instances from Santos et al. [2007] with a maximum of 40000 nodes, 800000 arcs and 1 resource. The testing on these instances focuses on comparing the solution time of solving the MIP reformulation and the modified label-setting algorithm.

In both datasets, the deviation $h_{ijr}$ in cost and resource consumption is an integer generated in the interval $[0, t_{ijr}]$ following a uniform distribution. The protection level is varied between 2 and 5 resulting in a total of 840 instances. We report on the effect of graph reduction on the size of the instances and on the quality of the initial lower and upper bounds. Then, we compare the three solution approaches of robust RCSPP.

## 4.1 Effects of graph reduction

Table 2 summarizes the results of the graph reduction algorithm. Average results are reported and grouped according to resource consumption. Specifically, instances with resource consumption inversely related to cost are denoted as ***Inverse***, and instances with uniformly generated resource consumption are denoted as ***Uniform***. **Avg_A%** denotes the average percentage of arcs remaining after graph reduction. **Avg_UBG%** refers to the average gap calculated as $100 * \frac{(UB-LB)}{LB}$ where UB and LB are the upper and lower bounds obtained during the graph reduction. **Avg_OPG%** refers to the average gap calculated as $100 * \frac{Opt-LB}{LB}$ where $Opt$ refers to the optimal objective value obtained after solving the instance by the modified label-setting algorithm. The graph reduction runs in less than 0.0005 seconds in all instances, so individual times are not reported. When resource consumption is inversely related to cost, the number of arcs of the reduced graph is 10.85% of the original graph. Reduction is less significant for uniform resource consumption with about 72% of the arcs remaining in the reduced graph.

The quality of the lower and upper bounds show similar trends. The average gap between optimal objective value and LB is 91.87% of LB and the average gap between UB and LB is 107.02% of LB for ***Inverse*** instances. On the other hand, the average gaps for ***Uniform*** instances are 181.88% and 327.60%.

These statistics suggest that ***Uniform*** instances have more dense networks and may be more difficult to solve. Because the performance of Lagrangian based reduction depends on the quality of the Lagrangian multipliers and the UB, we expect that having tight $LB$ and $UB$ tends to result in a smaller network after reduction. This is shown by the positive correlation between **Avg_A%** and **Avg_UBG%**. A single implementation of resource based reduction under protection level $\Gamma$ removes at least as many vertices and arcs as it does under $\Gamma - 1$. This is because $H_r$ increases when $\Gamma$ increases. As a result, the right hand sides of inequalities (16) and (17) become more restricting. However, when $\Gamma$ increases, Lagrangian cost based reduction may be stronger or weaker depending on the upper bound, $H_r$ and $\mu_r$. In Table 3, **Avg_Res%** and **Avg_Lag%** report the average percentage of arcs removed by resource based reduction and Lagrangian cost based reduction, respectively. The additional number of arcs removed from $\Gamma = 2$ to $\Gamma = 3$ is higher under the

Lagrangian based reduction. The percentages are stable for higher values of $\Gamma$. This is because $\hat{\Gamma}_r = \min\{l, \Gamma_r\}$.

Table 2: Summary results on graph reduction procedure.

| Inverse | Avg_UBG% | Avg_OPG% | Avg_A% |
|---|---|---|---|
| $\Gamma = 2$ | 80.93 | 80.75 | 5.26 |
| $\Gamma = 3$ | 114.26 | 91.98 | 12.63 |
| $\Gamma = 4$ | 116.26 | 97.18 | 12.75 |
| $\Gamma = 5$ | 116.63 | 97.55 | 12.76 |
| Average | 107.02 | 91.87 | 10.85 |
| Uniform | | | |
| $\Gamma = 2$ | 230.75 | 137.77 | 62.56 |
| $\Gamma = 3$ | 344.03 | 201.52 | 74.38 |
| $\Gamma = 4$ | 367.80 | 192.79 | 75.38 |
| $\Gamma = 5$ | 367.80 | 195.45 | 75.46 |
| Average | 327.60 | 181.88 | 71.94 |

Table 3: Percentage of arcs removed by resource based reduction and Lagrangian cost based reduction.

| | Inverse | | Uniform | |
|---|---|---|---|---|
| | Avg_Res% | Avg_Lag% | Avg_Res% | Avg_Lag% |
| $\Gamma = 2$ | 52.66 | 42.09 | 10.52 | 26.92 |
| $\Gamma = 3$ | 53.07 | 34.31 | 9.99 | 15.63 |
| $\Gamma = 4$ | 53.08 | 34.17 | 9.96 | 14.67 |
| $\Gamma = 5$ | 53.08 | 34.16 | 9.96 | 14.58 |

## 4.2 Comparison of solution approaches for robust RCSPP

For the first dataset, Tables 5 to 9 report on the size of the original graph given by the number of nodes $|N|$ and number of arcs $|A|$, the number of resources $|R| - 1$, the lower and upper bounds and the relative gap after graph reduction, the clock time used by the sequential algorithm, by CPLEX on the original graph (**OG**) with default optimality tolerance and on the reduced graph (**RG**) under three optimality tolerances $ep1 = 0.05\%, ep2 = 0.01\%$ and $ep3 = 0.001\%$, and the modified label setting algorithm, and the optimal objective value (**Opt**). Tables 6 to 9 report on detailed statistics on all 60 instances with protection levels 2 to 5, respectively, and Table 5 gives average results. The best average solution time under the three optimality tolerances is presented in Table 5 under column **RG**. As the underlying network is relatively small, the computational time used by the label correcting algorithm is negligible and is not reported.

Both the modified label-setting algorithm and CPLEX successfully solved all feasible instances, and determined that the rest are infeasible. The sequential algorithm fails to solve instances 25-30 and 55-60 for $\Gamma = 2, 3$. For instances with up to four resources, it used 64.39 seconds and 343.42 seconds which is 487 and 1431 times slower than the modified label-setting algorithm. Since the sequential algorithm is dominated for $\Gamma = 2, 3$ and instances become more difficult for $\Gamma = 4, 5$, we omitted the comparison with the sequential algorithm for the rest of the testing. Note that the sequential algorithm solves a set of independent problems which could benefit from parallel implementation. The modified label-setting algorithm achieves significantly smaller computational times than CPLEX. Times vary between 0.001 and 0.522 with an average 0.02. The modified label-setting algorithm is on average 160 times faster than the best time achieved by CPLEX under all optimality tolerances. Moreover, a smaller optimality tolerance does not always result in larger time because CPLEX may develop different branching trees under different optimality tolerances. Looking at the average times for increasing $\Gamma$, there is no evidence that instances become harder with higher $\Gamma$ both for modified label-setting algorithm and CPLEX. On the other hand, graph reduction reduces time of CPLEX by about 50%.

While instance 54 is infeasible in the original data, the other infeasible instances are infeasible because of the robust term. For all infeasible instances in Tables 6 to 9, we increased the resource capacities and rerun the testing. In this testing the default optimality tolerance is used by CPLEX. Table 11 reports on these instances and confirms the findings.

The results in Table 5 suggest again that ***Uniform*** instances are more difficult to solve than ***Inverse*** instances. The modified label-setting algorithm solves ***Inverse*** instances in an average of 0.0004 seconds which is about 76 times faster than the average time used to solve ***Uniform*** instances. The average times used by CPLEX on reduced graph are 0.144 seconds and 7.777 seconds for ***Inverse*** and ***Uniform*** instances, respectively. Again, the size of the network after graph reduction is an important factor affecting the difficulty of the instances.

We compare (**MLS**) and (**PPGP**) on the first dataset without applying graph reduction and without cost uncertainty in the objective function. Table 10 shows the resulting computational time for both approaches. While the times for both algorithms are very small, less than 0.02 seconds in most instances, PPGP seems to be faster on these instances.

For the 210 large instances, Table 4 reports on the average times used by modified label correcting algorithm **MLCA**, **MLS**, and CPLEX on the reduced graph with optimality tolerance 0.05%. For CPLEX we report both the time until an optimal solution is first detected (**Incum**) and the total time (**RG_ep1**). As the network becomes larger, the time consumed by the modified label correcting algorithm dominates the entire solution process. It consumes an average of 194.63 seconds before **MLS** starts, while **MLS** only uses an average of 0.20 seconds. On the other hand, CPLEX consumes an average of 1329.96 seconds while the optimal solution is detected in an average of 1052.87 seconds. The last column gives the ratio $\frac{\textbf{MLCA+MLS}}{\textbf{RG\_ep1}}$. On average, CPLEX uses 9.5 times more than the total time used by MLCA and MLS.

Table 4: Average computational time for large instances.

| | MLCA | MLS | CPLEX | | $\frac{\text{RG}}{\text{MLCA+MLS}}$ |
|---|---|---|---|---|---|
| | | | Incum | RG_ep1 | |
| $\Gamma = 2$ | 51.474 | 0.078 | 722.289 | 898.607 | 17.431 |
| $\Gamma = 3$ | 129.440 | 0.166 | 1045.055 | 1260.740 | 9.727 |
| $\Gamma = 4$ | 233.714 | 0.227 | 1235.319 | 1462.152 | 6.250 |
| $\Gamma = 5$ | 363.921 | 0.322 | 1208.822 | 1698.321 | 4.663 |

# 5. Conclusion

In this paper, we address the robust resource constrained shortest path problem where the cost and resource consumption parameters on an arc are defined by intervals and a protection level is prespecified for each random parameter. We extend the resource based and Lagrangian relaxation based graph reduction techniques for the resource constrained shortest path problem to the robust case. The results show that graph reduction helps to reduce the overall solution time significantly. A new dominance rule is developed and used within a label setting algorithm to solve the robust problem. The new dominance rule is theoretically different from that of Pessoa et al. [2015] as each rule is able to identify dominated partial paths that the other fails to. Numerical testing shows that the modified label setting algorithm based on the new dominance rule dominates the direct solution of an equivalent MIP reformulation and the sequential algorithm, but slightly inferior to the label setting algorithm based on the dominance rule proposed by Pessoa et al. [2015].

## References


Agra, A., Christiansen, M., Figueiredo, R., Hvattum, L. M., Poss, M., and Requejo, C. The robust vehicle routing problem with time windows. *Computers & Operations Research*, 40(3):856–866, 2013.

Álvarez Miranda, E., Ljubić, I., and Toth, P. A note on the bertsimas & sim algorithm for robust combinatorial optimization problems. *4OR*, 11(4):349–360, 2013. ISSN 1619-4500.

Ball, M. *Network routing*. Handbooks in operations research and management science. Elsevier, 1995.

Beasley, J. E. and Christofides, N. An algorithm for the resource constrained shortest path problem. *Networks*, 19(4):379–394, 1989.

Ben-Tal, A. and Nemirovski, A. Robust convex optimization. *Mathematics of Operations Research*, 23(4):769–805, 1998.

Ben-Tal, A. and Nemirovski, A. Robust solutions of uncertain linear programs. *Operations Research Letters*, 25(1):1–13, 1999.

Ben-Tal, A. and Nemirovski, A. Robust solutions of linear programming problems contaminated with uncertain data. *Mathematical Programming*, 88(3):411–424, 2000.

Ben-Tal, A., Ghaoui, L. E., and Nemirovski, A. *Robust Optimization*. Princeton Series in Applied Mathematics. Princeton University Press, 2009.

Bertsimas, D. and Sim, M. Robust discrete optimization and network flows. *Mathematical Programming*, 98(1-3):49–71, 2003.

Bertsimas, D. and Sim, M. The price of robustness. *Operations Research*, 52(1):35–53, 2004.

Bertsimas, D., Brown, D., and Caramanis, C. Theory and applications of robust optimization. *SIAM Review*, 53(3):464–501, 2011.

Borndörfer, R., Grötschel, M., and Löbel, A. Scheduling duties by adaptive column generation, 2001.

Catanzaro, D., Labbé, M., and Salazar-Neumann, M. Reduction approaches for robust shortest path problems. *Comput. Oper. Res.*, 38(11):1610–1619, 2011.

Table 5: Summary of computational time.

| | | $\lvert R\rvert - 1$ | MLS | Cplex OG | Cplex RG | Algorithm 6 |
|---|---|---|---|---|---|---|
| $\Gamma = 2$ | Inverse | 1 | 0.000 | 0.412 | 0.028 | 0.475 |
| | | 2 | 0.000 | 1.776 | 0.419 | 3.216 |
| | | 3 | 0.000 | 2.508 | 0.052 | 52.585 |
| | | 4 | 0.000 | 6.580 | 0.036 | 456.052 |
| | | 10 | 0.000 | 15.520 | 0.017 | |
| | | Average | 0.000 | 5.359 | 0.110 | 128.082 |
| | Uniform | 1 | 0.015 | 0.512 | 0.697 | 0.014 |
| | | 2 | 0.002 | 1.147 | 0.243 | 0.022 |
| | | 3 | 0.014 | 2.298 | 1.321 | 0.406 |
| | | 4 | 0.034 | 3.349 | 1.987 | 2.339 |
| | | 10 | 0.094 | 34.812 | 30.908 | |
| | | Average | 0.032 | 8.424 | 7.031 | 0.695 |
| | Average | | 0.016 | 6.891 | 3.571 | 64.389 |
| $\Gamma = 3$ | Inverse | 1 | 0.000 | 0.603 | 0.120 | 0.957 |
| | | 2 | 0.001 | 1.349 | 0.182 | 6.231 |
| | | 3 | 0.001 | 2.510 | 0.115 | 122.837 |
| | | 4 | 0.001 | 7.153 | 0.170 | 2614.169 |
| | | 10 | 0.001 | 16.633 | 0.078 | |
| | | Average | 0.001 | 5.650 | 0.133 | 686.048 |
| | Uniform | 1 | 0.026 | 1.025 | 0.772 | 0.015 |
| | | 2 | 0.009 | 1.296 | 0.554 | 0.021 |
| | | 3 | 0.029 | 5.204 | 2.668 | 0.863 |
| | | 4 | 0.053 | 9.624 | 5.801 | 2.238 |
| | | 10 | 0.024 | 41.386 | 29.076 | |
| | | Average | 0.028 | 11.707 | 7.774 | 0.784 |
| | Average | | 0.014 | 8.678 | 3.953 | 343.416 |
| $\Gamma = 4$ | Inverse | 1 | 0.001 | 0.586 | 0.115 | |
| | | 2 | 0.000 | 1.605 | 0.364 | |
| | | 3 | 0.001 | 2.768 | 0.185 | |
| | | 4 | 0.001 | 8.018 | 0.211 | |
| | | 10 | 0.001 | 18.764 | 0.076 | |
| | | Average | 0.001 | 6.348 | 0.190 | |
| | Uniform | 1 | 0.028 | 0.815 | 0.684 | |
| | | 2 | 0.010 | 1.589 | 0.580 | |
| | | 3 | 0.053 | 9.023 | 3.991 | |
| | | 4 | 0.064 | 15.729 | 10.291 | |
| | | 10 | 0.017 | 40.021 | 26.928 | |
| | | Average | 0.034 | 13.435 | 8.495 | |
| | Average | | 0.017 | 9.892 | 4.342 | |
| $\Gamma = 5$ | Inverse | 1 | 0.000 | 0.823 | 0.089 | |
| | | 2 | 0.001 | 3.377 | 0.226 | |
| | | 3 | 0.000 | 3.242 | 0.135 | |
| | | 4 | 0.001 | 7.536 | 0.185 | |
| | | 10 | 0.000 | 16.758 | 0.071 | |
| | | Average | 0.000 | 6.347 | 0.141 | |
| | Uniform | 1 | 0.028 | 1.387 | 0.821 | |
| | | 2 | 0.016 | 1.985 | 0.546 | |
| | | 3 | 0.053 | 7.739 | 5.442 | |
| | | 4 | 0.062 | 14.260 | 7.925 | |
| | | 10 | 0.019 | 40.139 | 24.308 | |
| | | Average | 0.036 | 13.102 | 7.808 | |
| | Average | | 0.018 | 9.724 | 3.975 | |

Table 6: Detailed results with protection level $\Gamma = 2$.

| | | Original graph | | | Reduced graph | | | | | | | Time | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | CPLEX | | | | | |
| | instance | $\|N\|$ | $\|A\|$ | $\|R\|-1$ | $\|N\|$ | $\|A\|$ | Res% | Lag% | LB | UB | Gap% | MLS | OG | RG_ep1 | RG_ep2 | RG_ep3 | Algorithm 6 | Opt |
| | 1 | 100 | 955 | 1 | 13 | 19 | 6.91 | 91.10 | 133 | 181 | 36.09 | 0 | 0.225 | 0.015 | 0.002 | 0.002 | 0.009 | 181 |
| | 2 | 100 | 955 | 1 | 7 | 7 | 7.33 | 91.94 | 149 | 165 | 10.74 | 0 | 0.218 | 0.001 | 0.015 | 0.001 | 0.007 | 165 |
| | 3 | 200 | 2040 | 1 | 7 | 7 | 99.66 | 0.00 | 420 | - | - | 0 | 0.204 | 0.000 | 0.000 | 0.000 | 0.005 | - |
| | 4 | 200 | 2040 | 1 | 7 | 7 | 99.66 | 0.00 | 420 | - | - | 0 | 0.478 | 0.000 | 0.000 | 0.000 | 0.004 | - |
| | 5 | 500 | 4858 | 1 | 184 | 396 | 0.60 | 91.25 | 488.571 | 1061 | 117.16 | 0.001 | 1.287 | 0.234 | 0.280 | 0.234 | 1.839 | 937 |
| | 6 | 500 | 4858 | 1 | 55 | 81 | 0.89 | 97.45 | 654 | 977 | 49.39 | 0 | 1.157 | 0.016 | 0.015 | 0.016 | 0.988 | 977 |
| | 7 | 100 | 990 | 2 | 6 | 6 | 1.31 | 98.08 | 91 | 104 | 14.29 | 0 | 0.304 | 0.001 | 0.016 | 0.001 | 0.009 | 104 |
| | 8 | 100 | 990 | 2 | 62 | 184 | 4.24 | 77.17 | 89 | 182 | 104.49 | 0 | 0.538 | 0.125 | 0.140 | 0.125 | 6.538 | 159 |
| | 9 | 200 | 2080 | 2 | 10 | 13 | 25.43 | 73.94 | 271 | 385 | 42.07 | 0 | 0.609 | 0.016 | 0.001 | 0.015 | 0.016 | 385 |
| | 10 | 200 | 2080 | 2 | 195 | 1269 | 38.99 | 0.00 | 267.971 | - | - | 0.002 | 1.188 | 1.185 | 1.373 | 1.217 | 9.19 | 566 |
| | 11 | 500 | 4847 | 2 | 55 | 79 | 87.79 | 10.58 | 863 | 1878 | 117.61 | 0 | 3.247 | 0.031 | 0.015 | 0.032 | 1.514 | 1513 |
| | 12 | 500 | 4847 | 2 | 46 | 63 | 95.52 | 3.18 | 867 | 2194 | 153.06 | 0 | 2.810 | 0.016 | 0.016 | 0.015 | 2.03 | 2194 |
| | 13 | 100 | 990 | 3 | 6 | 6 | 1.52 | 97.88 | 91 | 104 | 14.29 | 0 | 0.250 | 0.015 | 0.002 | 0.002 | 0.014 | 104 |
| | 14 | 100 | 990 | 3 | 62 | 182 | 4.34 | 77.27 | 89 | 182 | 104.49 | 0 | 0.520 | 0.141 | 0.140 | 0.140 | 225.431 | 159 |
| Inverse | 15 | 200 | 2080 | 3 | 86 | 198 | 73.46 | 17.02 | 268 | 844 | 214.93 | 0.001 | 2.431 | 0.172 | 0.172 | 0.171 | 31.488 | 666 |
| | 16 | 200 | 2080 | 3 | 41 | 67 | 82.21 | 14.57 | 510 | 758 | 48.63 | 0 | 2.091 | 0.015 | 0.016 | 0.015 | 17.043 | 758 |
| | 17 | 500 | 4847 | 3 | 54 | 77 | 87.93 | 10.48 | 863 | 1878 | 117.61 | 0 | 3.518 | 0.016 | 0.016 | 0.031 | 14.308 | 1513 |
| | 18 | 500 | 4847 | 3 | 46 | 63 | 95.75 | 2.95 | 867 | 2194 | 153.06 | 0 | 6.622 | 0.015 | 0.015 | 0.016 | 27.226 | 2194 |
| | 19 | 100 | 990 | 4 | 9 | 11 | 2.32 | 96.57 | 103 | 127 | 23.30 | 0 | 0.376 | 0.016 | 0.003 | 0.015 | 3.298 | 127 |
| | 20 | 100 | 990 | 4 | 21 | 39 | 4.55 | 91.52 | 96.6444 | 159 | 64.52 | 0 | 0.629 | 0.016 | 0.015 | 0.032 | 407.645 | 159 |
| | 21 | 200 | 2080 | 4 | 68 | 138 | 72.31 | 21.06 | 450 | 844 | 87.56 | 0 | 3.532 | 0.078 | 0.062 | 0.078 | 456.505 | 740 |
| | 22 | 200 | 2080 | 4 | 41 | 68 | 89.86 | 6.88 | 509 | 928 | 82.32 | 0 | 3.563 | 0.031 | 0.031 | 0.031 | 256.119 | 815 |
| | 23 | 500 | 4847 | 4 | 47 | 63 | 90.06 | 8.64 | 863 | 1878 | 117.61 | 0 | 11.010 | 0.031 | 0.016 | 0.031 | 268.447 | 1878 |
| | 24 | 500 | 4847 | 4 | 102 | 161 | 96.68 | 0.00 | 858 | - | - | 0.001 | 22.994 | 0.078 | 0.078 | 0.078 | 1344.3 | 3599 |
| | 25 | 100 | 990 | 10 | 9 | 11 | 7.68 | 91.21 | 103 | 127 | 23.30 | 0 | 0.696 | 0.016 | 0.016 | 0.015 | - | 127 |
| | 26 | 100 | 990 | 10 | 9 | 11 | 14.04 | 84.85 | 108 | 159 | 47.22 | 0 | 0.773 | 0.017 | 0.031 | 0.032 | - | 159 |
| | 27 | 200 | 2080 | 10 | 16 | 19 | 99.09 | 0.00 | 393.98 | - | - | 0 | 4.050 | 0.001 | 0.001 | 0.001 | - | - |
| | 28 | 200 | 2080 | 10 | 0 | 0 | 100.00 | 0.00 | -10000 | - | - | 0 | 2.652 | 0 | 0 | 0 | - | - |
| | 29 | 500 | 4847 | 10 | 40 | 53 | 91.79 | 7.12 | 863 | 1878 | 117.61 | 0 | 24.355 | 0.046 | 0.032 | 0.031 | - | 1878 |
| | 30 | 500 | 4847 | 10 | 71 | 109 | 97.75 | 0.00 | 858 | - | - | 0.001 | 36.110 | 0.094 | 0.078 | 0.078 | - | - |
| | 31 | 100 | 959 | 1 | 97 | 845 | 2.92 | 8.97 | 1.5 | 12 | 700.00 | 0.013 | 0.253 | 0.187 | 0.328 | 0.203 | 0.01 | 5 |
| | 32 | 100 | 959 | 1 | 88 | 541 | 3.65 | 39.94 | 2 | 10 | 400.00 | 0.016 | 0.906 | 0.203 | 0.172 | 0.187 | 0.008 | 7 |
| | 33 | 200 | 1971 | 1 | 98 | 216 | 2.94 | 86.10 | 6 | 8 | 33.33 | 0.001 | 0.317 | 0.031 | 0.031 | 0.047 | 0.008 | 8 |
| | 34 | 200 | 1971 | 1 | 194 | 1915 | 2.84 | 0.00 | 6 | 39 | 550.00 | 0.021 | 0.621 | 0.359 | 0.312 | 0.358 | 0.011 | 8 |
| | 35 | 500 | 4978 | 1 | 467 | 4461 | 5.99 | 4.40 | 6 | 16 | 166.67 | 0.025 | 1.209 | 0.889 | 0.827 | 0.842 | 0.027 | 11 |
| | 36 | 500 | 4978 | 1 | 443 | 2799 | 6.41 | 37.36 | 6 | 14 | 133.33 | 0.013 | 1.181 | 0.796 | 0.811 | 0.812 | 0.022 | 9 |
| | 37 | 100 | 999 | 2 | 89 | 834 | 10.41 | 6.11 | 3.375 | 12 | 255.56 | 0.005 | 0.467 | 0.296 | 0.312 | 0.312 | 0.031 | 9 |
| | 38 | 100 | 999 | 2 | 89 | 601 | 10.41 | 29.43 | 3.625 | 10 | 175.86 | 0.002 | 0.418 | 0.406 | 0.328 | 0.327 | 0.019 | 10 |
| | 39 | 200 | 1960 | 2 | 94 | 183 | 11.63 | 79.03 | 5.25 | 9 | 71.43 | 0 | 0.677 | 0.046 | 0.032 | 0.031 | 0.015 | 9 |
| | 40 | 200 | 1960 | 2 | 184 | 1767 | 9.85 | 0.00 | 5.75 | - | - | 0.002 | 1.030 | 0.982 | 0.983 | 0.905 | 0.037 | 12 |
| | 41 | 500 | 4868 | 2 | 59 | 96 | 8.38 | 89.65 | 4 | 6 | 50.00 | 0 | 3.178 | 0.016 | 0.031 | 0.016 | 0.016 | 6 |
| | 42 | 500 | 4868 | 2 | 49 | 83 | 11.65 | 86.65 | 4 | 6 | 50.00 | 0 | 1.793 | 0.016 | 0.015 | 0.016 | 0.016 | 6 |
| | 43 | 100 | 999 | 3 | 90 | 889 | 11.01 | 0.00 | 3.8202 | 35 | 816.18 | 0.027 | 1.336 | 1.185 | 1.170 | 1.202 | 0.074 | 10 |
| | 44 | 100 | 999 | 3 | 90 | 878 | 12.11 | 0.00 | 4.52778 | - | - | 0.015 | 1.433 | 1.029 | 1.030 | 1.030 | 0.974 | 16 |
| | 45 | 200 | 1960 | 3 | 184 | 1774 | 9.49 | 0.00 | 5.25 | - | - | 0.037 | 2.768 | 1.810 | 1.950 | 1.825 | 1.116 | 14 |
| Uniform | 46 | 200 | 1960 | 3 | 181 | 1734 | 11.53 | 0.00 | 5.75 | - | - | 0.007 | 2.426 | 1.747 | 1.747 | 1.669 | 0.197 | 16 |
| | 47 | 500 | 4868 | 3 | 103 | 178 | 11.40 | 84.94 | 4 | 7 | 75.00 | 0 | 16.661 | 0.062 | 0.047 | 0.063 | 0.037 | 7 |
| | 48 | 500 | 4868 | 3 | 36 | 49 | 12.86 | 86.13 | 4 | 7 | 75.00 | 0 | 3.162 | 0.031 | 0.016 | 0.015 | 0.036 | 7 |
| | 49 | 100 | 999 | 4 | 90 | 889 | 11.01 | 0.00 | 3.86654 | - | - | 0.037 | 2.602 | 1.903 | 1.950 | 1.872 | 2.531 | 14 |
| | 50 | 100 | 999 | 4 | 90 | 878 | 12.11 | 0.00 | 4.57734 | - | - | 0.005 | 1.552 | 1.311 | 1.310 | 1.326 | 1.841 | 16 |
| | 51 | 200 | 1960 | 4 | 182 | 1754 | 10.51 | 0.00 | 5.79245 | - | - | 0.154 | 8.471 | 4.774 | 4.789 | 4.852 | 5.408 | 18 |
| | 52 | 200 | 1960 | 4 | 181 | 1718 | 12.35 | 0.00 | 6.70245 | - | - | 0.007 | 5.460 | 1.326 | 1.404 | 1.358 | 2.205 | 22 |
| | 53 | 500 | 4868 | 4 | 97 | 160 | 12.78 | 83.94 | 4 | 7 | 75.00 | 0 | 8.345 | 0.062 | 0.047 | 0.047 | 0.977 | 7 |
| | 54 | 500 | 4868 | 4 | 36 | 49 | 13.99 | 85.00 | 4.25 | 7 | 64.71 | 0.001 | 6.069 | 0.016 | 0.015 | 0.016 | 1.069 | 7 |
| | 55 | 100 | 999 | 10 | 90 | 839 | 16.02 | 0.00 | 4.15904 | - | - | 0.009 | 2.371 | 1.997 | 2.777 | 2.277 | - | - |
| | 56 | 100 | 999 | 10 | 89 | 730 | 26.93 | 0.00 | 5.38219 | - | - | 0.002 | 1.509 | 1.077 | 1.030 | 1.045 | - | - |
| | 57 | 200 | 1960 | 10 | 180 | 1705 | 13.01 | 0.00 | 6.85392 | - | - | 0.006 | 11.912 | 9.282 | 9.984 | 9.453 | - | - |
| | 58 | 200 | 1960 | 10 | 177 | 1618 | 17.45 | 0.00 | 8.99812 | - | - | 0 | 8.161 | 6.505 | 6.162 | 6.848 | - | - |
| | 59 | 500 | 4868 | 10 | 469 | 4559 | 6.35 | 0.00 | 3.49231 | - | - | 0.522 | 158.220 | 115.597 | 111.977 | 112.851 | - | - |
| | 60 | 500 | 4868 | 10 | 469 | 4495 | 7.66 | 0.00 | 4.26087 | - | - | 0.026 | 67.346 | 44.461 | 43.867 | 44.882 | - | - |

Table 7: Detailed results with protection level $\Gamma = 3$.

| | | Original graph | | | Reduced graph | | | | | | | Time | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | MLS | CPLEX | | | | Algorithm 6 | Opt |
| | instance | $\|N\|$ | $\|A\|$ | $\|R\|-1$ | $\|N\|$ | $\|A\|$ | Res% | Lag% | LB | UB | Gap% | | OG | RG_ep1 | RG_ep2 | RG_ep3 | | |
| Inverse | 1 | 100 | 955 | 1 | 14 | 20 | 7.120419 | 90.78534 | 136 | 186 | 36.76471 | 0 | 0.244 | 0.016 | 0.015 | 0.016 | 0.009 | 186 |
| | 2 | 100 | 955 | 1 | 100 | 892 | 6.596859 | 0 | 131 | 434 | 231.2977 | 0 | 0.326 | 0.328 | 0.343 | 0.327 | 0.99 | 203 |
| | 3 | 200 | 2040 | 1 | 7 | 7 | 99.65686 | 0 | 420 | - | - | 0 | 0.190 | 0.000 | 0.000 | 0.000 | 0.005 | - |
| | 4 | 200 | 2040 | 1 | 7 | 7 | 99.65686 | 0 | 420 | - | - | 0 | 0.196 | 0.015 | 0.000 | 0.000 | 0.006 | - |
| | 5 | 500 | 4858 | 1 | 244 | 614 | 0.679292 | 86.68176 | 489.571 | 1140 | 132.8569 | 0.001 | 1.265 | 0.343 | 0.375 | 0.359 | 3.654 | 958 |
| | 6 | 500 | 4858 | 1 | 69 | 111 | 0.864553 | 96.85056 | 656 | 1024 | 56.09756 | 0 | 1.397 | 0.031 | 0.016 | 0.016 | 1.076 | 1024 |
| | 7 | 100 | 990 | 2 | 95 | 486 | 1.010101 | 49.89899 | 89 | 219 | 146.0674 | 0.001 | 0.258 | 0.171 | 0.156 | 0.171 | 15.592 | 128 |
| | 8 | 100 | 990 | 2 | 62 | 184 | 4.242424 | 77.17172 | 89 | 182 | 104.4944 | 0 | 0.257 | 0.110 | 0.187 | 0.109 | 5.723 | 160 |
| | 9 | 200 | 2080 | 2 | 10 | 13 | 25.57692 | 73.79808 | 272 | 390 | 43.38235 | 0 | 0.483 | 0.015 | 0.001 | 0.016 | 0.017 | 390 |
| | 10 | 200 | 2080 | 2 | 195 | 1269 | 38.99038 | 0 | 267.971 | - | - | 0.002 | 1.103 | 0.796 | 0.780 | 0.765 | 11.482 | 628 |
| | 11 | 500 | 4847 | 2 | 55 | 79 | 87.82752 | 10.5426 | 876 | 1923 | 119.5205 | 0 | 2.857 | 0.031 | 0.016 | 0.015 | 1.612 | 1552 |
| | 12 | 500 | 4847 | 2 | 68 | 103 | 95.39922 | 2.475758 | 872 | 2447 | 180.6193 | 0 | 3.136 | 0.016 | 0.046 | 0.016 | 2.962 | 2447 |
| | 13 | 100 | 990 | 3 | 95 | 480 | 1.212121 | 50.30303 | 89 | 219 | 146.0674 | 0.001 | 0.355 | 0.187 | 0.234 | 0.280 | 413.982 | 128 |
| | 14 | 100 | 990 | 3 | 62 | 182 | 4.343434 | 77.27273 | 89 | 182 | 104.4944 | 0 | 0.348 | 0.156 | 0.172 | 0.218 | 191.239 | 160 |
| | 15 | 200 | 2080 | 3 | 102 | 267 | 75.91346 | 11.25 | 269 | 925 | 243.8662 | 0.001 | 2.587 | 0.219 | 0.218 | 0.234 | 36.072 | 722 |
| | 16 | 200 | 2080 | 3 | 57 | 107 | 85.72115 | 9.134615 | 512 | 951 | 85.74219 | 0.001 | 4.870 | 0.078 | 0.078 | 0.078 | 31.004 | 951 |
| | 17 | 500 | 4847 | 3 | 54 | 77 | 87.97194 | 10.43945 | 881 | 1923 | 118.2747 | 0 | 3.073 | 0.016 | 0.016 | 0.031 | 16.992 | 1552 |
| | 18 | 500 | 4847 | 3 | 67 | 100 | 95.66742 | 2.269445 | 872 | 2447 | 180.6193 | 0 | 3.829 | 0.031 | 0.016 | 0.031 | 47.734 | 2447 |
| | 19 | 100 | 990 | 4 | 9 | 11 | 2.323232 | 96.56566 | 103 | 128 | 24.27184 | 0 | 0.368 | 0.015 | 0.003 | 0.003 | 3.065 | 128 |
| | 20 | 100 | 990 | 4 | 40 | 86 | 5.858586 | 85.45455 | 94.6444 | 182 | 92.29875 | 0 | 0.595 | 0.047 | 0.031 | 0.047 | 1938.42 | 182 |
| | 21 | 200 | 2080 | 4 | 83 | 185 | 75.48077 | 15.625 | 451 | 925 | 105.0998 | 0.001 | 3.518 | 0.202 | 0.188 | 0.187 | 738.652 | 925 |
| | 22 | 200 | 2080 | 4 | 46 | 76 | 90.24038 | 6.105769 | 512 | 951 | 85.74219 | 0 | 3.242 | 0.031 | 0.031 | 0.031 | 305.115 | 951 |
| | 23 | 500 | 4847 | 4 | 246 | 529 | 89.08603 | 0 | 858 | - | - | 0.003 | 19.172 | 0.686 | 0.702 | 0.702 | 11254.1 | 2462 |
| | 24 | 500 | 4847 | 4 | 102 | 161 | 96.67836 | 0 | 858 | - | - | 0.001 | 16.023 | 0.062 | 0.062 | 0.063 | 1445.66 | - |
| | 25 | 100 | 990 | 10 | 9 | 11 | 7.676768 | 91.21212 | 103 | 128 | 24.27184 | 0 | 0.696 | 0.015 | 0.015 | 0.016 | - | 128 |
| | 26 | 100 | 990 | 10 | 11 | 16 | 16.9697 | 81.41414 | 108 | 182 | 68.51852 | 0 | 1.107 | 0.016 | 0.031 | 0.016 | - | 182 |
| | 27 | 200 | 2080 | 10 | 16 | 19 | 99.08654 | 0 | 393.98 | - | - | 0 | 4.398 | 0.001 | 0.001 | 0.001 | - | - |
| | 28 | 200 | 2080 | 10 | 0 | 0 | 100 | 0 | -10000 | - | - | 0 | 2.640 | 0 | 0 | 0 | - | - |
| | 29 | 500 | 4847 | 10 | 107 | 180 | 92.36641 | 3.91995 | 869 | 2462 | 183.3142 | 0.002 | 54.473 | 0.375 | 0.359 | 0.359 | - | 2462 |
| | 30 | 500 | 4847 | 10 | 71 | 109 | 97.75119 | 0 | 858 | - | - | 0.001 | 36.481 | 0.062 | 0.062 | 0.078 | - | - |
| Uniform | 31 | 100 | 959 | 1 | 98 | 877 | 2.815433 | 5.735141 | 1.5 | 13 | 766.6667 | 0.036 | 0.910 | 0.359 | 0.343 | 0.343 | 0.009 | 7 |
| | 32 | 100 | 959 | 1 | 92 | 647 | 3.441084 | 29.09281 | 2 | 11 | 450 | 0.021 | 0.293 | 0.156 | 0.172 | 0.156 | 0.008 | 8 |
| | 33 | 200 | 1971 | 1 | 194 | 1915 | 2.841197 | 0 | 6 | 42 | 600 | 0.04 | 0.648 | 0.655 | 0.655 | 0.671 | 0.011 | 10 |
| | 34 | 200 | 1971 | 1 | 194 | 1915 | 2.841197 | 0 | 6 | 42 | 600 | 0.022 | 0.398 | 0.343 | 0.312 | 0.344 | 0.01 | 8 |
| | 35 | 500 | 4978 | 1 | 469 | 4539 | 5.966252 | 2.852551 | 6 | 17 | 183.3333 | 0.026 | 2.124 | 2.278 | 2.277 | 2.293 | 0.033 | 12 |
| | 36 | 500 | 4978 | 1 | 454 | 3370 | 6.347931 | 25.9542 | 6 | 15 | 150 | 0.013 | 1.775 | 0.889 | 0.874 | 0.873 | 0.02 | 9 |
| | 37 | 100 | 999 | 2 | 90 | 870 | 10.41041 | 2.502503 | 3.375 | 13 | 285.1852 | 0.013 | 0.524 | 1.155 | 1.123 | 1.155 | 0.024 | 10 |
| | 38 | 100 | 999 | 2 | 90 | 895 | 10.41041 | 0 | 3.625 | - | - | 0.037 | 1.452 | 1.092 | 1.108 | 1.061 | 0.026 | 13 |
| | 39 | 200 | 1960 | 2 | 94 | 183 | 11.63265 | 79.03061 | 5.25 | 9 | 71.42857 | 0 | 0.601 | 0.046 | 0.032 | 0.031 | 0.014 | 9 |
| | 40 | 200 | 1960 | 2 | 184 | 1767 | 9.846939 | 0 | 5.75 | - | - | 0.002 | 0.979 | 1.279 | 0.936 | 1.092 | 0.021 | 12 |
| | 41 | 500 | 4868 | 2 | 142 | 279 | 9.161873 | 85.10682 | 3.28571 | 7 | 113.0438 | 0 | 2.632 | 0.109 | 0.109 | 0.110 | 0.022 | 7 |
| | 42 | 500 | 4868 | 2 | 49 | 83 | 11.64749 | 86.64749 | 4 | 6 | 50 | 0.001 | 1.585 | 0.016 | 0.015 | 0.265 | 0.016 | 6 |
| | 43 | 100 | 999 | 3 | 90 | 889 | 11.01101 | 0 | 3.8202 | 39 | 920.889 | 0.027 | 1.317 | 0.952 | 0.921 | 0.920 | 0.978 | 14 |
| | 44 | 100 | 999 | 3 | 90 | 878 | 12.11211 | 0 | 4.52778 | - | - | 0.027 | 1.669 | 1.295 | 1.326 | 1.326 | 0.977 | 21 |
| | 45 | 200 | 1960 | 3 | 184 | 1774 | 9.489796 | 0 | 5.25 | - | - | 0.03 | 3.125 | 2.231 | 2.138 | 2.169 | 1.07 | 15 |
| | 46 | 200 | 1960 | 3 | 181 | 1734 | 11.53061 | 0 | 5.75 | - | - | 0.018 | 2.548 | 2.403 | 2.434 | 2.355 | 0.973 | 18 |
| | 47 | 500 | 4868 | 3 | 252 | 658 | 9.737058 | 76.7461 | 3.3125 | 8 | 141.5094 | 0.001 | 7.613 | 0.421 | 0.406 | 0.437 | 0.045 | 8 |
| | 48 | 500 | 4868 | 3 | 469 | 4553 | 6.47083 | 0 | 3.65625 | - | - | 0.068 | 14.951 | 8.892 | 8.783 | 8.814 | 1.133 | 17 |
| | 49 | 100 | 999 | 4 | 90 | 889 | 11.01101 | 0 | 3.86654 | - | - | 0.02 | 2.149 | 2.184 | 2.199 | 2.169 | 1.37 | 15 |
| | 50 | 100 | 999 | 4 | 90 | 878 | 12.11211 | 0 | 4.57734 | - | - | 0.003 | 1.388 | 1.341 | 1.341 | 1.310 | 1.974 | - |
| | 51 | 200 | 1960 | 4 | 182 | 1754 | 10.5102 | 0 | 5.79245 | - | - | 0.103 | 6.473 | 6.521 | 6.474 | 6.833 | 2.567 | 21 |
| | 52 | 200 | 1960 | 4 | 181 | 1718 | 12.34694 | 0 | 6.70245 | - | - | 0.007 | 6.381 | 4.181 | 4.149 | 4.196 | 1.996 | - |
| | 53 | 500 | 4868 | 4 | 238 | 611 | 12.18159 | 75.26705 | 3.32819 | 8 | 140.3709 | 0.001 | 9.316 | 0.578 | 0.577 | 0.593 | 1.011 | 8 |
| | 54 | 500 | 4868 | 4 | 469 | 4553 | 6.47083 | 0 | 3.71429 | - | - | 0.183 | 32.037 | 19.999 | 20.280 | 19.812 | 4.51 | 22 |
| | 55 | 100 | 999 | 10 | 90 | 839 | 16.01602 | 0 | 4.15904 | - | - | 0.004 | 2.046 | 1.528 | 1.545 | 1.451 | - | - |
| | 56 | 100 | 999 | 10 | 89 | 730 | 26.92693 | 0 | 5.38219 | - | - | 0.001 | 2.046 | 0.858 | 0.858 | 0.842 | - | - |
| | 57 | 200 | 1960 | 10 | 180 | 1705 | 13.0102 | 0 | 6.85392 | - | - | 0.003 | 12.564 | 10.249 | 10.062 | 10.187 | - | - |
| | 58 | 200 | 1960 | 10 | 177 | 1618 | 17.44898 | 0 | 8.99812 | - | - | 0 | 11.214 | 6.645 | 7.207 | 6.662 | - | - |
| | 59 | 500 | 4868 | 10 | 469 | 4559 | 6.347576 | 0 | 3.49231 | - | - | 0.126 | 141.920 | 91.495 | 91.276 | 90.215 | - | - |
| | 60 | 500 | 4868 | 10 | 469 | 4495 | 7.662284 | 0 | 4.26087 | - | - | 0.01 | 78.528 | 65.115 | 65.333 | 65.099 | - | - |

Table 8: Detailed results with protection level $\Gamma = 4$.

| | | Original graph | | Reduced graph | | | | | | | | Time | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | MLS | CPLEX | | | | Opt |
| | instance | $\lvert N\rvert$ | $\lvert A\rvert$ | $\lvert R\rvert - 1$ | $\lvert N\rvert$ | $\lvert A\rvert$ | Res% | Lag% | LB | UB | Gap% | | OG | RG_ep1 | RG_ep2 | RG_ep3 | |
| | 1 | 100 | 955 | 1 | 14 | 20 | 7.120419 | 90.78534 | 136 | 186 | 36.76471 | 0 | 0.215 | 0.002 | 0.015 | 0.002 | 186 |
| | 2 | 100 | 955 | 1 | 100 | 892 | 6.596859 | 0 | 131 | 450 | 243.5115 | 0.001 | 0.395 | 0.312 | 0.437 | 0.749 | 203 |
| | 3 | 200 | 2040 | 1 | 7 | 7 | 99.65686 | 0 | 420 | - | - | 0 | 0.201 | 0.000 | 0.000 | 0.000 | - |
| | 4 | 200 | 2040 | 1 | 7 | 7 | 99.65686 | 0 | 420 | - | - | 0 | 0.186 | 0.000 | 0.015 | 0.000 | - |
| | 5 | 500 | 4858 | 1 | 260 | 685 | 0.741046 | 85.1585 | 490.571 | 1160 | 136.4591 | 0.001 | 1.296 | 0.343 | 0.343 | 0.374 | 959 |
| | 6 | 500 | 4858 | 1 | 90 | 149 | 0.988061 | 95.94483 | 657 | 1063 | 61.79604 | 0.001 | 1.221 | 0.031 | 0.031 | 0.031 | 1063 |
| | 7 | 100 | 990 | 2 | 95 | 486 | 1.010101 | 49.89899 | 89 | 219 | 146.0674 | 0 | 0.255 | 0.187 | 0.218 | 0.187 | 128 |
| | 8 | 100 | 990 | 2 | 62 | 184 | 4.242424 | 77.17172 | 89 | 182 | 104.4944 | 0 | 0.319 | 0.094 | 0.094 | 0.093 | 160 |
| | 9 | 200 | 2080 | 2 | 10 | 13 | 25.57692 | 73.79808 | 272 | 390 | 43.38235 | 0 | 0.456 | 0.001 | 0.016 | 0.001 | 390 |
| | 10 | 200 | 2080 | 2 | 195 | 1269 | 38.99038 | 0 | 267.971 | - | - | 0.002 | 1.062 | 1.872 | 1.903 | 1.872 | 636 |
| | 11 | 500 | 4847 | 2 | 55 | 79 | 87.86878 | 10.50134 | 876 | 1935 | 120.8904 | 0 | 5.119 | 0.016 | 0.015 | 0.016 | 1935 |
| | 12 | 500 | 4847 | 2 | 68 | 103 | 95.39922 | 2.475758 | 872 | 2454 | 181.422 | 0 | 2.418 | 0.031 | 0.016 | 0.016 | 2454 |
| | 13 | 100 | 990 | 3 | 95 | 480 | 1.212121 | 50.30303 | 89 | 219 | 146.0674 | 0.001 | 0.415 | 0.702 | 0.655 | 0.655 | 128 |
| | 14 | 100 | 990 | 3 | 62 | 182 | 4.343434 | 77.27273 | 89 | 182 | 104.4944 | 0 | 0.528 | 0.156 | 0.156 | 0.156 | 160 |
| Inverse | 15 | 200 | 2080 | 3 | 104 | 283 | 76.10577 | 10.28846 | 269 | 937 | 248.3271 | 0.001 | 1.658 | 0.219 | 0.218 | 0.218 | 727 |
| | 16 | 200 | 2080 | 3 | 57 | 107 | 85.72115 | 9.134615 | 515 | 954 | 85.24272 | 0.001 | 6.536 | 0.047 | 0.047 | 0.047 | 954 |
| | 17 | 500 | 4847 | 3 | 54 | 77 | 88.0132 | 10.39818 | 881 | 1935 | 119.6368 | 0.001 | 3.795 | 0.046 | 0.031 | 0.015 | 1935 |
| | 18 | 500 | 4847 | 3 | 67 | 100 | 95.66742 | 2.269445 | 872 | 2454 | 181.422 | 0.001 | 3.674 | 0.031 | 0.031 | 0.016 | 2454 |
| | 19 | 100 | 990 | 4 | 9 | 11 | 2.323232 | 96.56566 | 103 | 128 | 24.27184 | 0 | 0.345 | 0.016 | 0.016 | 0.015 | 128 |
| | 20 | 100 | 990 | 4 | 40 | 86 | 5.858586 | 85.45455 | 94.6444 | 182 | 92.29875 | 0 | 0.545 | 0.047 | 0.047 | 0.031 | 182 |
| | 21 | 200 | 2080 | 4 | 84 | 190 | 75.67308 | 15.19231 | 453 | 937 | 106.8433 | 0.001 | 5.742 | 0.312 | 0.297 | 0.296 | 937 |
| | 22 | 200 | 2080 | 4 | 46 | 76 | 90.24038 | 6.105769 | 515 | 954 | 85.24272 | 0 | 3.101 | 0.031 | 0.031 | 0.031 | 954 |
| | 23 | 500 | 4847 | 4 | 246 | 529 | 89.08603 | 0 | 858 | - | - | 0.003 | 17.982 | 0.796 | 0.812 | 0.827 | 2565 |
| | 24 | 500 | 4847 | 4 | 102 | 161 | 96.67836 | 0 | 858 | - | - | 0 | 20.391 | 0.062 | 0.063 | 0.078 | - |
| | 25 | 100 | 990 | 10 | 9 | 11 | 7.676768 | 91.21212 | 103 | 128 | 24.27184 | 0 | 0.678 | 0.015 | 0.016 | 0.016 | 128 |
| | 26 | 100 | 990 | 10 | 11 | 16 | 16.9697 | 81.41414 | 108 | 182 | 68.51852 | 0 | 1.122 | 0.032 | 0.031 | 0.031 | 182 |
| | 27 | 200 | 2080 | 10 | 16 | 19 | 99.08654 | 0 | 393.98 | - | - | 0 | 2.892 | 0.001 | 0.015 | 0.001 | - |
| | 28 | 200 | 2080 | 10 | 0 | 0 | 100 | 0 | - | - | - | 0 | 2.678 | 0 | 0 | 0 | - |
| | 29 | 500 | 4847 | 10 | 117 | 201 | 92.24262 | 3.610481 | 866 | 2565 | 196.1894 | 0.002 | 63.124 | 0.343 | 0.343 | 0.359 | 2565 |
| | 30 | 500 | 4847 | 10 | 71 | 109 | 97.75119 | 0 | 858 | - | - | 0.001 | 42.088 | 0.063 | 0.063 | 0.062 | - |
| | 31 | 100 | 959 | 1 | 98 | 899 | 2.919708 | 3.336809 | 1.5 | 14 | 833.3333 | 0.034 | 0.242 | 0.358 | 0.484 | 0.328 | 7 |
| | 32 | 100 | 959 | 1 | 92 | 713 | 3.336809 | 22.31491 | 2 | 12 | 500 | 0.026 | 0.311 | 0.344 | 0.296 | 0.343 | 9 |
| | 33 | 200 | 1971 | 1 | 194 | 1915 | 2.841197 | 0 | 6 | 44 | 633.3333 | 0.037 | 0.535 | 0.453 | 0.453 | 0.468 | 10 |
| | 34 | 200 | 1971 | 1 | 194 | 1915 | 2.841197 | 0 | 6 | 44 | 633.3333 | 0.028 | 0.394 | 0.358 | 0.374 | 0.374 | 8 |
| | 35 | 500 | 4978 | 1 | 469 | 4539 | 5.966252 | 2.852551 | 6 | 17 | 183.3333 | 0.027 | 2.133 | 1.747 | 1.731 | 1.731 | 12 |
| | 36 | 500 | 4978 | 1 | 460 | 3846 | 6.368019 | 16.37204 | 6 | 16 | 166.6667 | 0.013 | 1.277 | 0.889 | 0.889 | 0.858 | 9 |
| | 37 | 100 | 999 | 2 | 90 | 870 | 10.41041 | 2.502503 | 3.375 | 13 | 285.1852 | 0.014 | 0.743 | 1.107 | 1.107 | 1.092 | 11 |
| | 38 | 100 | 999 | 2 | 90 | 895 | 10.41041 | 0 | 3.625 | - | - | 0.033 | 1.529 | 0.718 | 0.671 | 0.608 | 13 |
| | 39 | 200 | 1960 | 2 | 94 | 183 | 11.63265 | 79.03061 | 5.25 | 9 | 71.42857 | 0.001 | 0.585 | 0.047 | 0.031 | 0.031 | 9 |
| | 40 | 200 | 1960 | 2 | 184 | 1767 | 9.846939 | 0 | 5.75 | - | - | 0.007 | 1.247 | 1.155 | 1.170 | 1.185 | 12 |
| | 41 | 500 | 4868 | 2 | 280 | 827 | 8.052588 | 74.95892 | 3.28571 | 8 | 143.4786 | 0.002 | 3.606 | 0.515 | 0.499 | 0.546 | 8 |
| | 42 | 500 | 4868 | 2 | 49 | 83 | 11.64749 | 86.64749 | 4 | 6 | 50 | 0 | 1.823 | 0.016 | 0.016 | 0.015 | 6 |
| | 43 | 100 | 999 | 3 | 90 | 889 | 11.01101 | 0 | 3.8202 | 42 | 999.4189 | 0.024 | 2.027 | 0.952 | 0.936 | 0.951 | 14 |
| | 44 | 100 | 999 | 3 | 90 | 878 | 12.11211 | 0 | 4.52778 | - | - | 0.029 | 1.925 | 1.685 | 1.685 | 1.669 | - |
| Uniform | 45 | 200 | 1960 | 3 | 184 | 1774 | 9.489796 | 0 | 5.25 | - | - | 0.027 | 3.654 | 2.449 | 2.730 | 2.465 | 15 |
| | 46 | 200 | 1960 | 3 | 181 | 1734 | 11.53061 | 0 | 5.75 | - | - | 0.017 | 3.783 | 3.042 | 3.073 | 3.183 | 19 |
| | 47 | 500 | 4868 | 3 | 252 | 658 | 9.737058 | 76.7461 | 3.3125 | 8 | 141.5094 | 0.001 | 5.109 | 0.468 | 0.640 | 0.499 | 8 |
| | 48 | 500 | 4868 | 3 | 469 | 4553 | 6.47083 | 0 | 3.65625 | - | - | 0.222 | 37.642 | 15.350 | 15.444 | 15.351 | 20 |
| | 49 | 100 | 999 | 4 | 90 | 889 | 11.01101 | 0 | 3.86654 | - | - | 0.017 | 2.449 | 1.966 | 2.200 | 1.794 | 15 |
| | 50 | 100 | 999 | 4 | 90 | 878 | 12.11211 | 0 | 4.57734 | - | - | 0.003 | 1.603 | 1.497 | 1.498 | 1.498 | - |
| | 51 | 200 | 1960 | 4 | 182 | 1754 | 10.5102 | 0 | 5.79245 | - | - | 0.103 | 10.609 | 5.179 | 5.211 | 5.164 | 25 |
| | 52 | 200 | 1960 | 4 | 181 | 1718 | 12.34694 | 0 | 6.70245 | - | - | 0.006 | 5.743 | 4.275 | 4.212 | 4.306 | - |
| | 53 | 500 | 4868 | 4 | 238 | 611 | 12.18159 | 75.26705 | 3.32819 | 8 | 140.3709 | 0.001 | 6.289 | 0.578 | 0.577 | 0.578 | 8 |
| | 54 | 500 | 4868 | 4 | 469 | 4553 | 6.47083 | 0 | 3.71429 | - | - | 0.253 | 67.678 | 49.062 | 48.860 | 48.408 | - |
| | 55 | 100 | 999 | 10 | 90 | 839 | 16.01602 | 0 | 4.15904 | - | - | 0.005 | 3.250 | 1.638 | 1.622 | 1.653 | - |
| | 56 | 100 | 999 | 10 | 89 | 730 | 26.92693 | 0 | 5.38219 | - | - | 0.001 | 0.843 | 0.593 | 0.546 | 0.546 | - |
| | 57 | 200 | 1960 | 10 | 180 | 1705 | 13.0102 | 0 | 6.85392 | - | - | 0.004 | 13.348 | 8.144 | 7.800 | 7.597 | - |
| | 58 | 200 | 1960 | 10 | 177 | 1618 | 17.44898 | 0 | 8.99812 | - | - | 0 | 3.837 | 2.465 | 2.449 | 2.324 | - |
| | 59 | 500 | 4868 | 10 | 469 | 4559 | 6.347576 | 0 | 3.49231 | - | - | 0.085 | 145.019 | 91.339 | 92.805 | 92.103 | - |
| | 60 | 500 | 4868 | 10 | 469 | 4495 | 7.662284 | 0 | 4.26087 | - | - | 0.009 | 73.827 | 58.812 | 58.750 | 57.346 | - |

Table 9: Detailed results with protection level $\Gamma = 5$.

| | Original graph | | | Reduced graph | | | | | | | | Time | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | MLS | CPLEX | | | | Opt |
| | instance | $\lvert N\rvert$ | $\lvert A\rvert$ | $\lvert R\rvert - 1$ | $\lvert N\rvert$ | $\lvert A\rvert$ | Res% | Lag% | LB | UB | Gap% | | OG | RG_ep1 | RG_ep2 | RG_ep3 | |
| Inverse | 1 | 100 | 955 | 1 | 14 | 20 | 7.120419 | 90.78534 | 136 | 186 | 36.76471 | 0 | 0.399 | 0.002 | 0.015 | 0.002 | 186 |
| | 2 | 100 | 955 | 1 | 100 | 892 | 6.596859 | 0 | 131 | 455 | 247.3282 | 0 | 0.353 | 0.218 | 0.234 | 0.343 | 203 |
| | 3 | 200 | 2040 | 1 | 7 | 7 | 99.65686 | 0 | 420 | - | - | 0 | 0.263 | 0.000 | 0.015 | 0.000 | - |
| | 4 | 200 | 2040 | 1 | 7 | 7 | 99.65686 | 0 | 420 | - | - | 0 | 0.272 | 0.000 | 0.000 | 0.000 | - |
| | 5 | 500 | 4858 | 1 | 260 | 685 | 0.741046 | 85.1585 | 490.571 | 1160 | 136.4591 | 0.001 | 1.775 | 0.280 | 0.281 | 0.297 | 959 |
| | 6 | 500 | 4858 | 1 | 90 | 149 | 0.988061 | 95.94483 | 657 | 1063 | 61.79604 | 0 | 1.876 | 0.031 | 0.031 | 0.016 | 1063 |
| | 7 | 100 | 990 | 2 | 95 | 486 | 1.010101 | 49.89899 | 89 | 219 | 146.0674 | 0 | 0.393 | 0.188 | 0.187 | 0.187 | 128 |
| | 8 | 100 | 990 | 2 | 62 | 184 | 4.242424 | 77.17172 | 89 | 182 | 104.4944 | 0 | 0.383 | 0.109 | 0.093 | 0.110 | 160 |
| | 9 | 200 | 2080 | 2 | 10 | 13 | 25.57692 | 73.79808 | 272 | 390 | 43.38235 | 0 | 0.673 | 0.001 | 0.016 | 0.001 | 390 |
| | 10 | 200 | 2080 | 2 | 195 | 1269 | 38.99038 | 0 | 267.971 | - | - | 0.002 | 1.725 | 1.014 | 1.108 | 1.358 | 636 |
| | 11 | 500 | 4847 | 2 | 55 | 79 | 87.86878 | 10.50134 | 876 | 1935 | 120.8904 | 0 | 6.220 | 0.031 | 0.015 | 0.016 | 1935 |
| | 12 | 500 | 4847 | 2 | 68 | 103 | 95.39922 | 2.475758 | 872 | 2454 | 181.422 | 0.001 | 10.866 | 0.015 | 0.031 | 0.032 | 2454 |
| | 13 | 100 | 990 | 3 | 95 | 480 | 1.212121 | 50.30303 | 89 | 219 | 146.0674 | 0.001 | 0.545 | 0.234 | 0.203 | 0.250 | 128 |
| | 14 | 100 | 990 | 3 | 62 | 182 | 4.343434 | 77.27273 | 89 | 182 | 104.4944 | 0 | 0.405 | 0.140 | 0.156 | 0.172 | 160 |
| | 15 | 200 | 2080 | 3 | 104 | 283 | 76.10577 | 10.28846 | 269 | 937 | 248.3271 | 0.001 | 3.068 | 0.359 | 0.359 | 0.359 | 727 |
| | 16 | 200 | 2080 | 3 | 57 | 107 | 85.72115 | 9.134615 | 515 | 954 | 85.24272 | 0 | 5.886 | 0.046 | 0.047 | 0.047 | 954 |
| | 17 | 500 | 4847 | 3 | 54 | 77 | 88.0132 | 10.39818 | 881 | 1935 | 119.6368 | 0 | 4.724 | 0.016 | 0.016 | 0.031 | 1935 |
| | 18 | 500 | 4847 | 3 | 67 | 100 | 95.66742 | 2.269445 | 872 | 2454 | 181.422 | 0 | 4.826 | 0.016 | 0.031 | 0.031 | 2454 |
| | 19 | 100 | 990 | 4 | 9 | 11 | 2.323232 | 96.56566 | 103 | 128 | 24.27184 | 0 | 0.544 | 0.015 | 0.016 | 0.003 | 128 |
| | 20 | 100 | 990 | 4 | 40 | 86 | 5.858586 | 85.45455 | 94.6444 | 182 | 92.29875 | 0 | 0.511 | 0.063 | 0.062 | 0.031 | 182 |
| | 21 | 200 | 2080 | 4 | 84 | 190 | 75.67308 | 15.19231 | 453 | 937 | 106.8433 | 0.001 | 5.892 | 0.234 | 0.234 | 0.234 | 937 |
| | 22 | 200 | 2080 | 4 | 46 | 76 | 90.24038 | 6.105769 | 515 | 954 | 85.24272 | 0 | 3.566 | 0.047 | 0.031 | 0.032 | 954 |
| | 23 | 500 | 4847 | 4 | 246 | 529 | 89.08603 | 0 | 858 | - | - | 0.003 | 17.404 | 0.780 | 0.795 | 0.749 | 2603 |
| | 24 | 500 | 4847 | 4 | 102 | 161 | 96.67836 | 0 | 858 | - | - | 0.001 | 17.300 | 0.062 | 0.063 | 0.062 | - |
| | 25 | 100 | 990 | 10 | 9 | 11 | 7.676768 | 91.21212 | 103 | 128 | 24.27184 | 0 | 0.694 | 0.006 | 0.006 | 0.006 | 128 |
| | 26 | 100 | 990 | 10 | 11 | 16 | 16.9697 | 81.41414 | 108 | 182 | 68.51852 | 0 | 1.363 | 0.031 | 0.031 | 0.032 | 182 |
| | 27 | 200 | 2080 | 10 | 16 | 19 | 99.08654 | 0 | 393.98 | - | - | 0 | 3.458 | 0.001 | 0.001 | 0.001 | - |
| | 28 | 200 | 2080 | 10 | 0 | 0 | 100 | 0 | -10000 | - | - | 0 | 3.006 | 0 | 0 | 0 | - |
| | 29 | 500 | 4847 | 10 | 122 | 208 | 92.24262 | 3.466061 | 866 | 2603 | 200.5774 | 0.002 | 62.468 | 0.344 | 0.343 | 0.344 | 2603 |
| | 30 | 500 | 4847 | 10 | 71 | 109 | 97.75119 | 0 | 858 | - | - | 0 | 29.558 | 0.047 | 0.124 | 0.063 | - |
| Uniform | 31 | 100 | 959 | 1 | 98 | 899 | 2.919708 | 3.336809 | 1.5 | 14 | 833.3333 | 0.032 | 0.502 | 0.374 | 0.312 | 0.312 | 7 |
| | 32 | 100 | 959 | 1 | 92 | 713 | 3.336809 | 22.31491 | 2 | 12 | 500 | 0.029 | 0.530 | 0.234 | 0.281 | 0.234 | 9 |
| | 33 | 200 | 1971 | 1 | 194 | 1915 | 2.841197 | 0 | 6 | 45 | 650 | 0.037 | 0.637 | 0.390 | 0.390 | 0.390 | 10 |
| | 34 | 200 | 1971 | 1 | 194 | 1915 | 2.841197 | 0 | 6 | 46 | 666.6667 | 0.029 | 0.538 | 0.374 | 0.390 | 0.374 | 8 |
| | 35 | 500 | 4978 | 1 | 469 | 4539 | 5.966252 | 2.852551 | 6 | 17 | 183.3333 | 0.026 | 3.304 | 1.919 | 1.918 | 1.918 | 12 |
| | 36 | 500 | 4978 | 1 | 460 | 3846 | 6.368019 | 16.37204 | 6 | 16 | 166.6667 | 0.017 | 2.809 | 1.684 | 1.669 | 1.700 | 10 |
| | 37 | 100 | 999 | 2 | 90 | 895 | 10.41041 | 0 | 3.375 | 43 | 1174.074 | 0.022 | 1.385 | 0.670 | 0.702 | 0.686 | 11 |
| | 38 | 100 | 999 | 2 | 90 | 895 | 10.41041 | 0 | 3.625 | - | - | 0.066 | 1.853 | 1.263 | 1.264 | 1.248 | 13 |
| | 39 | 200 | 1960 | 2 | 94 | 183 | 11.63265 | 79.03061 | 5.25 | 9 | 71.42857 | 0 | 0.930 | 0.032 | 0.031 | 0.032 | 9 |
| | 40 | 200 | 1960 | 2 | 184 | 1767 | 9.846939 | 0 | 5.75 | - | - | 0.007 | 0.992 | 0.733 | 0.733 | 0.733 | 12 |
| | 41 | 500 | 4868 | 2 | 280 | 827 | 8.052588 | 74.95892 | 3.28571 | 8 | 143.4786 | 0.001 | 4.493 | 0.577 | 0.562 | 0.562 | 8 |
| | 42 | 500 | 4868 | 2 | 49 | 83 | 11.64749 | 86.64749 | 4 | 6 | 50 | 0 | 2.256 | 0.016 | 0.016 | 0.015 | 6 |
| | 43 | 100 | 999 | 3 | 90 | 889 | 11.01101 | 0 | 3.8202 | 43 | 1025.596 | 0.027 | 2.067 | 1.544 | 1.482 | 1.467 | 14 |
| | 44 | 100 | 999 | 3 | 90 | 878 | 12.11211 | 0 | 4.52778 | - | - | 0.03 | 3.232 | 1.825 | 1.747 | 1.747 | - |
| | 45 | 200 | 1960 | 3 | 184 | 1774 | 9.489796 | 0 | 5.25 | - | - | 0.03 | 6.420 | 2.745 | 2.761 | 2.762 | 16 |
| | 46 | 200 | 1960 | 3 | 181 | 1734 | 11.53061 | 0 | 5.75 | - | - | 0.019 | 3.613 | 3.495 | 3.542 | 3.400 | 20 |
| | 47 | 500 | 4868 | 3 | 252 | 658 | 9.737058 | 76.7461 | 3.3125 | 8 | 141.5094 | 0.001 | 5.570 | 0.499 | 0.515 | 0.546 | 8 |
| | 48 | 500 | 4868 | 3 | 469 | 4553 | 6.47083 | 0 | 3.65625 | - | - | 0.209 | 25.529 | 22.667 | 22.604 | 23.244 | 20 |
| | 49 | 100 | 999 | 4 | 90 | 889 | 11.01101 | 0 | 3.86654 | - | - | 0.017 | 2.714 | 1.888 | 2.044 | 1.919 | 15 |
| | 50 | 100 | 999 | 4 | 90 | 878 | 12.11211 | 0 | 4.57734 | - | - | 0.003 | 2.523 | 0.531 | 0.562 | 0.561 | - |
| | 51 | 200 | 1960 | 4 | 182 | 1754 | 10.5102 | 0 | 5.79245 | - | - | 0.103 | 6.801 | 5.694 | 5.772 | 5.913 | 25 |
| | 52 | 200 | 1960 | 4 | 181 | 1718 | 12.34694 | 0 | 6.70245 | - | - | 0.006 | 7.468 | 4.165 | 4.087 | 4.088 | - |
| | 53 | 500 | 4868 | 4 | 238 | 611 | 12.18159 | 75.26705 | 3.32819 | 8 | 140.3709 | 0.001 | 6.598 | 0.608 | 0.515 | 0.499 | 8 |
| | 54 | 500 | 4868 | 4 | 469 | 4553 | 6.47083 | 0 | 3.71429 | - | - | 0.241 | 59.453 | 34.663 | 35.256 | 34.617 | - |
| | 55 | 100 | 999 | 10 | 90 | 839 | 16.01602 | 0 | 4.15904 | - | - | 0.005 | 2.139 | 1.716 | 1.888 | 1.716 | - |
| | 56 | 100 | 999 | 10 | 89 | 730 | 26.92693 | 0 | 5.38219 | - | - | 0.001 | 0.991 | 0.530 | 0.530 | 0.530 | - |
| | 57 | 200 | 1960 | 10 | 180 | 1705 | 13.0102 | 0 | 6.85392 | - | - | 0.004 | 15.757 | 9.547 | 9.734 | 9.813 | - |
| | 58 | 200 | 1960 | 10 | 177 | 1618 | 17.44898 | 0 | 8.99812 | - | - | 0.001 | 3.897 | 2.511 | 2.262 | 2.511 | - |
| | 59 | 500 | 4868 | 10 | 469 | 4559 | 6.347576 | 0 | 3.49231 | - | - | 0.094 | 151.075 | 86.394 | 86.268 | 86.939 | - |
| | 60 | 500 | 4868 | 10 | 469 | 4495 | 7.662284 | 0 | 4.26087 | - | - | 0.01 | 66.974 | 45.256 | 45.163 | 44.928 | - |

Table 10: Comparison to label setting algorithm of Pessoa et al. [2015].

| | | Original graph | | | $\Gamma = 2$ | | | $\Gamma = 3$ | | | $\Gamma = 4$ | | | $\Gamma = 5$ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Instance | $\|N\|$ | $\|A\|$ | $\|R\|$ | Opt | MLS | PPGP | Opt | MLS | PPGP | Opt | MLS | PPGP | Opt | MLS | PPGP |
| | 1 | 100 | 955 | 1 | 131 | 0.001 | 0 | 131 | 0.001 | 0 | 131 | 0.001 | 0 | 131 | 0.002 | 0 |
| | 2 | 100 | 955 | 1 | 131 | 0.001 | 0 | 142 | 0.001 | 0 | 142 | 0.001 | 0 | 142 | 0.002 | 0 |
| | 3 | 100 | 959 | 1 | 4 | 0.024 | 0.001 | 5 | 0.019 | 0.002 | 5 | 0.024 | 0.002 | 5 | 0.027 | 0.001 |
| | 4 | 100 | 959 | 1 | 5 | 0.04 | 0.001 | 6 | 0.033 | 0.002 | 6 | 0.021 | 0.002 | 6 | 0.025 | 0.001 |
| | 5 | 200 | 2040 | 1 | 100000 | 0.002 | 0 | 100000 | 0.003 | 0 | 100000 | 0.004 | 0 | 100000 | 0.003 | 0 |
| | 6 | 200 | 2040 | 1 | 100000 | 0.002 | 0 | 100000 | 0.002 | 0 | 100000 | 0.004 | 0 | 100000 | 0.004 | 0.001 |
| | 7 | 200 | 1971 | 1 | 6 | 0.005 | 0.003 | 7 | 0.013 | 0.004 | 7 | 0.017 | 0.003 | 7 | 0.023 | 0.004 |
| | 8 | 200 | 1971 | 1 | 7 | 0.014 | 0.003 | 7 | 0.013 | 0.004 | 7 | 0.075 | 0.003 | 7 | 0.044 | 0.004 |
| | 9 | 500 | 4858 | 1 | 652 | 0.02 | 0.003 | 690 | 0.026 | 0.003 | 690 | 0.04 | 0.002 | 690 | 0.041 | 0.003 |
| | 10 | 500 | 4858 | 1 | 690 | 0.017 | 0.003 | 690 | 0.027 | 0.002 | 690 | 0.052 | 0.002 | 690 | 0.04 | 0.003 |
| | 11 | 500 | 4978 | 1 | 8 | 0.007 | 0.006 | 8 | 0.008 | 0.006 | 8 | 0.029 | 0.007 | 8 | 0.017 | 0.008 |
| | 12 | 500 | 4978 | 1 | 8 | 0.012 | 0.007 | 8 | 0.012 | 0.011 | 8 | 0.029 | 0.008 | 9 | 0.017 | 0.008 |
| | 13 | 100 | 990 | 2 | 89 | 0.001 | 0 | 100 | 0.001 | 0 | 100 | 0.002 | 0 | 100 | 0.003 | 0 |
| | 14 | 100 | 990 | 2 | 100 | 0.001 | 0 | 100 | 0.003 | 0 | 100 | 0.002 | 0.001 | 100 | 0.004 | 0 |
| Uniform | 15 | 100 | 999 | 2 | 6 | 0.003 | 0.005 | 7 | 0.008 | 0.005 | 7 | 0.009 | 0.006 | 7 | 0.008 | 0.006 |
| | 16 | 100 | 999 | 2 | 7 | 0.004 | 0.004 | 8 | 0.004 | 0.004 | 8 | 0.005 | 0.004 | 10 | 0.008 | 0.004 |
| | 17 | 200 | 2080 | 2 | 339 | 0.004 | 0.001 | 339 | 0.006 | 0.001 | 339 | 0.009 | 0.001 | 339 | 0.009 | 0.001 |
| | 18 | 200 | 2080 | 2 | 426 | 0.007 | 0.001 | 426 | 0.01 | 0.001 | 426 | 0.008 | 0.001 | 426 | 0.01 | 0.001 |
| | 19 | 200 | 1960 | 2 | 7 | 0.004 | 0.004 | 7 | 0.008 | 0.004 | 7 | 0.013 | 0.005 | 7 | 0.015 | 0.005 |
| | 20 | 200 | 1960 | 2 | 8 | 0.004 | 0.003 | 8 | 0.006 | 0.003 | 8 | 0.007 | 0.003 | 8 | 0.009 | 0.003 |
| | 21 | 500 | 4847 | 2 | 1335 | 0.024 | 0 | 1335 | 0.034 | 0.001 | 1477 | 0.038 | 0.001 | 1477 | 0.065 | 0 |
| | 22 | 500 | 4847 | 2 | 1477 | 0.01 | 0 | 1477 | 0.034 | 0 | 1477 | 0.065 | 0.001 | 1477 | 0.074 | 0.001 |
| | 23 | 500 | 4868 | 2 | 4 | 0.009 | 0.019 | 4 | 0.008 | 0.017 | 5 | 0.044 | 0.017 | 5 | 0.031 | 0.019 |
| | 24 | 500 | 4868 | 2 | 5 | 0.004 | 0.012 | 5 | 0.009 | 0.011 | 5 | 0.014 | 0.01 | 5 | 0.06 | 0.011 |
| | 25 | 100 | 990 | 3 | 89 | 0.002 | 0 | 100 | 0.003 | 0 | 100 | 0.004 | 0 | 100 | 0.005 | 0.001 |
| | 26 | 100 | 990 | 3 | 100 | 0.002 | 0 | 100 | 0.004 | 0.001 | 100 | 0.004 | 0 | 100 | 0.005 | 0 |
| | 27 | 100 | 999 | 3 | 7 | 0.006 | 0.005 | 10 | 0.011 | 0.003 | 10 | 0.012 | 0.004 | 10 | 0.013 | 0.004 |
| | 28 | 100 | 999 | 3 | 11 | 0.006 | 0.002 | 16 | 0.004 | 0.001 | 100000 | 0.006 | 0.001 | 100000 | 0.006 | 0.001 |
| | 29 | 200 | 2080 | 3 | 552 | 0.006 | 0.001 | 552 | 0.013 | 0.001 | 552 | 0.013 | 0 | 552 | 0.012 | 0 |
| | 30 | 200 | 2080 | 3 | 552 | 0.008 | 0.001 | 651 | 0.014 | 0 | 651 | 0.011 | 0 | 651 | 0.014 | 0 |
| | 31 | 200 | 1960 | 3 | 10 | 0.01 | 0.006 | 10 | 0.012 | 0.006 | 11 | 0.013 | 0.005 | 12 | 0.021 | 0.005 |
| | 32 | 200 | 1960 | 3 | 12 | 0.006 | 0.002 | 13 | 0.006 | 0.002 | 13 | 0.007 | 0.001 | 13 | 0.008 | 0.001 |
| | 33 | 500 | 4847 | 3 | 1335 | 0.029 | 0.001 | 1335 | 0.054 | 0 | 1477 | 0.062 | 0 | 1477 | 0.106 | 0.001 |
| | 34 | 500 | 4847 | 3 | 1477 | 0.016 | 0 | 1477 | 0.037 | 0.001 | 1477 | 0.092 | 0 | 1477 | 0.11 | 0 |
| | 35 | 500 | 4868 | 3 | 5 | 0.063 | 0.032 | 5 | 0.055 | 0.025 | 5 | 0.03 | 0.025 | 5 | 0.093 | 0.025 |
| | 36 | 500 | 4868 | 3 | 5 | 0.014 | 0.013 | 11 | 0.013 | 0.01 | 17 | 0.043 | 0.01 | 17 | 0.048 | 0.01 |
| | 37 | 100 | 990 | 4 | 100 | 0.002 | 0 | 100 | 0.003 | 0.001 | 100 | 0.005 | 0 | 100 | 0.007 | 0 |
| | 38 | 100 | 990 | 4 | 100 | 0.003 | 0 | 119 | 0.004 | 0 | 119 | 0.006 | 0 | 119 | 0.007 | 0 |
| | 39 | 100 | 999 | 4 | 11 | 0.011 | 0.002 | 11 | 0.006 | 0.002 | 11 | 0.008 | 0.002 | 11 | 0.009 | 0.002 |
| | 40 | 100 | 999 | 4 | 11 | 0.002 | 0.001 | 100000 | 0.002 | 0.001 | 100000 | 0.004 | 0 | 100000 | 0.005 | 0.001 |
| | 41 | 200 | 2080 | 4 | 642 | 0.007 | 0.001 | 651 | 0.018 | 0.001 | 651 | 0.015 | 0 | 651 | 0.015 | 0 |
| | 42 | 200 | 2080 | 4 | 569 | 0.011 | 0 | 651 | 0.017 | 0 | 651 | 0.013 | 0 | 651 | 0.019 | 0 |
| | 43 | 200 | 1960 | 4 | 14 | 0.018 | 0.006 | 14 | 0.018 | 0.004 | 15 | 0.021 | 0.004 | 15 | 0.025 | 0.004 |
| | 44 | 200 | 1960 | 4 | 15 | 0.004 | 0.001 | 100000 | 0.005 | 0.001 | 100000 | 0.006 | 0.001 | 100000 | 0.009 | 0.001 |
| Uniform | 45 | 500 | 4847 | 4 | 1477 | 0.039 | 0 | 1797 | 0.081 | 0.001 | 1797 | 0.084 | 0 | 1797 | 0.124 | 0 |
| | 46 | 500 | 4847 | 4 | 2936 | 0.023 | 0 | 100000 | 0.062 | 0 | 100000 | 0.127 | 0 | 100000 | 0.143 | 0 |
| | 47 | 500 | 4868 | 4 | 5 | 0.016 | 0.04 | 5 | 0.035 | 0.027 | 5 | 0.023 | 0.022 | 5 | 0.077 | 0.025 |
| | 48 | 500 | 4868 | 4 | 5 | 0.021 | 0.013 | 15 | 0.027 | 0.009 | 100000 | 0.047 | 0.008 | 100000 | 0.053 | 0.008 |
| | 49 | 100 | 990 | 10 | 100 | 0.006 | 0 | 100 | 0.008 | 0.001 | 100 | 0.014 | 0 | 100 | 0.018 | 0 |
| | 50 | 100 | 990 | 10 | 100 | 0.006 | 0 | 119 | 0.011 | 0 | 119 | 0.014 | 0 | 119 | 0.018 | 0.001 |
| | 51 | 100 | 999 | 10 | 100000 | 0.003 | 0.001 | 100000 | 0.005 | 0.001 | 100000 | 0.008 | 0 | 100000 | 0.01 | 0 |
| | 52 | 100 | 999 | 10 | 100000 | 0.003 | 0 | 100000 | 0.005 | 0 | 100000 | 0.007 | 0 | 100000 | 0.011 | 0.001 |
| | 53 | 200 | 2080 | 10 | 100000 | 0.022 | 0 | 100000 | 0.035 | 0.001 | 100000 | 0.04 | 0 | 100000 | 0.044 | 0 |
| | 54 | 200 | 2080 | 10 | 100000 | 0.024 | 0 | 100000 | 0.034 | 0 | 100000 | 0.035 | 0 | 100000 | 0.044 | 0 |
| | 55 | 200 | 1960 | 10 | 100000 | 0.007 | 0.001 | 100000 | 0.01 | 0 | 100000 | 0.015 | 0.001 | 100000 | 0.021 | 0.001 |
| | 56 | 200 | 1960 | 10 | 100000 | 0.006 | 0 | 100000 | 0.009 | 0 | 100000 | 0.014 | 0 | 100000 | 0.02 | 0 |
| | 57 | 500 | 4847 | 10 | 1477 | 0.101 | 0.001 | 1797 | 0.189 | 0.001 | 1797 | 0.21 | 0.001 | 1797 | 0.274 | 0 |
| | 58 | 500 | 4847 | 10 | 100000 | 0.079 | 0 | 100000 | 0.18 | 0 | 100000 | 0.283 | 0 | 100000 | 0.329 | 0 |
| | 59 | 500 | 4868 | 10 | 100000 | 0.046 | 0.01 | 100000 | 0.037 | 0.006 | 100000 | 0.047 | 0.005 | 100000 | 0.064 | 0.006 |
| | 60 | 500 | 4868 | 10 | 100000 | 0.019 | 0.002 | 100000 | 0.029 | 0.003 | 100000 | 0.041 | 0.001 | 100000 | 0.057 | 0.001 |
| | | | | | Avg | 0.014 | 0.004 | Avg | 0.022 | 0.003 | Avg | 0.032 | 0.003 | Avg | 0.040 | 0.003 |

Table 11: Results on infeasible instances in Tables 6 to 9 with increased resource limits.

| | instance | | Original graph $\vert N\vert$ | Original graph $\vert A\vert$ | Original graph $\vert R\vert-1$ | Reduced graph $\vert N\vert$ | Reduced graph $\vert A\vert$ | LB | UB | Gap% | Time MLS | Time CPLEX OG | Time CPLEX RG | Opt |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\Gamma=2$ | Inverse | 3 | 200 | 2040 | 1 | 4 | 3 | 292 | 316 | 8.22 | 0 | 0.099 | 0.004 | 316 |
| | | 4 | 200 | 2040 | 1 | 4 | 3 | 282 | 329 | 16.67 | 0 | 0.099 | 0.005 | 329 |
| | | 27 | 200 | 2080 | 10 | 11 | 14 | 234 | 303 | 29.49 | 0 | 0.797 | 0.013 | 303 |
| | | 28 | 200 | 2080 | 10 | 68 | 166 | 202 | 536 | 165.35 | 0.001 | 14.663 | 0.289 | 536 |
| | | 30 | 500 | 4847 | 10 | 476 | 2590 | 611 | 1895 | 210.15 | 0.009 | 18.45 | 10.013 | 1266 |
| | Uniform | 55 | 100 | 999 | 10 | 70 | 188 | 3 | 6 | 100.00 | 0.001 | 0.341 | 0.042 | 6 |
| | | 56 | 100 | 999 | 10 | 45 | 95 | 3 | 5 | 66.67 | 0 | 0.237 | 0.027 | 5 |
| | | 57 | 200 | 1960 | 10 | 183 | 1711 | 5 | 15 | 200.00 | 0.003 | 1.134 | 0.976 | 6 |
| | | 58 | 200 | 1960 | 10 | 184 | 1795 | 5 | - | - | 0.002 | 1.631 | 1.368 | 8 |
| | | 59 | 500 | 4868 | 10 | 69 | 93 | 3 | 4 | 33.33 | 0 | 2.013 | 0.036 | 4 |
| | | 60 | 500 | 4868 | 10 | 58 | 79 | 3 | 4 | 33.33 | 0.001 | 2.325 | 0.027 | 4 |
| $\Gamma=3$ | Inverse | 3 | 200 | 2040 | 1 | 4 | 3 | 336 | 336 | 0.00 | 0 | 0.118 | 0.004 | 336 |
| | | 4 | 200 | 2040 | 1 | 4 | 3 | 343 | 343 | 0.00 | 0 | 0.098 | 0.001 | 343 |
| | | 24 | 500 | 4847 | 4 | 499 | 4040 | 611 | 2447 | 300.49 | 0.008 | 8.355 | 6.056 | 1322 |
| | | 27 | 200 | 2080 | 10 | 26 | 43 | 202 | 390 | 93.07 | 0 | 6.905 | 0.022 | 390 |
| | | 28 | 200 | 2080 | 10 | 68 | 166 | 203 | 537 | 164.53 | 0.001 | 10.866 | 0.165 | 537 |
| | | 30 | 500 | 4847 | 10 | 499 | 4036 | 611 | 2447 | 300.49 | 0.014 | 82.068 | 15.519 | 1322 |
| | Uniform | 50 | 100 | 999 | 4 | 45 | 95 | 3 | 5 | 66.67 | 0 | 0.118 | 0.012 | 5 |
| | | 52 | 200 | 1960 | 4 | 184 | 1795 | 5 | 35 | 600.00 | 0.001 | 0.494 | 0.586 | 8 |
| | | 55 | 100 | 999 | 10 | 70 | 188 | 3 | 6 | 100.00 | 0 | 0.231 | 0.041 | 6 |
| | | 56 | 100 | 999 | 10 | 45 | 95 | 3 | 5 | 66.67 | 0.001 | 0.246 | 0.027 | 5 |
| | | 57 | 200 | 1960 | 10 | 184 | 1751 | 5 | 16 | 220.00 | 0.003 | 0.992 | 1.015 | 6 |
| | | 58 | 200 | 1960 | 10 | 184 | 1795 | 5 | - | - | 0.002 | 1.598 | 1.598 | 8 |
| | | 59 | 500 | 4868 | 10 | 69 | 93 | 3 | 4 | 33.33 | 0 | 1.294 | 0.035 | 4 |
| | | 60 | 500 | 4868 | 10 | 138 | 243 | 3 | 5 | 66.67 | 0.003 | 2.055 | 0.157 | 4 |
| $\Gamma=4$ | Inverse | 3 | 200 | 2040 | 1 | 4 | 3 | 336 | 336 | 0.00 | 0 | 0.171 | 0.001 | 336 |
| | | 4 | 200 | 2040 | 1 | 4 | 3 | 343 | 343 | 0.00 | 0 | 0.103 | 0.002 | 343 |
| | | 24 | 500 | 4847 | 4 | 499 | 4042 | 611 | 2454 | 301.64 | 0.008 | 10.901 | 8.207 | 1371 |
| | | 27 | 200 | 2080 | 10 | 26 | 43 | 202 | 390 | 93.07 | 0 | 8.454 | 0.022 | 390 |
| | | 28 | 200 | 2080 | 10 | 68 | 166 | 203 | 537 | 164.53 | 0.001 | 16.482 | 0.19 | 537 |
| | | 30 | 500 | 4847 | 10 | 499 | 4038 | 611 | 2454 | 301.64 | 0.014 | 21.344 | 16.645 | 1411 |
| | Uniform | 44 | 100 | 999 | 3 | 45 | 95 | 3 | 5 | 66.67 | 0 | 0.098 | 0.011 | 5 |
| | | 50 | 100 | 999 | 4 | 45 | 95 | 3 | 5 | 66.67 | 0 | 0.126 | 0.014 | 5 |
| | | 52 | 200 | 1960 | 4 | 184 | 1795 | 5 | - | - | 0.001 | 1.013 | 0.713 | 8 |
| | | 54 | 500 | 4868 | 4 | 147 | 260 | 3 | 5 | 66.67 | 0.001 | 0.601 | 0.095 | 4 |
| | | 55 | 100 | 999 | 10 | 70 | 188 | 3 | 6 | 100.00 | 0.001 | 0.291 | 0.041 | 6 |
| | | 56 | 100 | 999 | 10 | 90 | 895 | 3 | 27 | 800.00 | 0.009 | 3.083 | 1.422 | 7 |
| | | 57 | 200 | 1960 | 10 | 184 | 1795 | 5 | 26 | 420.00 | 0.004 | 0.877 | 0.818 | 6 |
| | | 58 | 200 | 1960 | 10 | 184 | 1795 | 5 | - | - | 0.04 | 8.243 | 11.98 | 12 |
| | | 59 | 500 | 4868 | 10 | 69 | 93 | 3 | 4 | 33.33 | 0.001 | 1.653 | 0.036 | 4 |
| | | 60 | 500 | 4868 | 10 | 469 | 4572 | 3 | 28 | 833.33 | 0.022 | 7.416 | 3.957 | 4 |
| $\Gamma=5$ | Inverse | 3 | 200 | 2040 | 1 | 4 | 3 | 336 | 336 | 0.00 | 0 | 0.12 | 0.001 | 336 |
| | | 4 | 200 | 2040 | 1 | 4 | 3 | 343 | 343 | 0.00 | 0 | 0.1 | 0.001 | 343 |
| | | 24 | 500 | 4847 | 4 | 499 | 4042 | 611 | 2454 | 301.64 | 0.008 | 6.477 | 10.712 | 1411 |
| | | 27 | 200 | 2080 | 10 | 26 | 43 | 202 | 390 | 93.07 | 0 | 8.56 | 0.024 | 390 |
| | | 28 | 200 | 2080 | 10 | 68 | 166 | 203 | 537 | 164.53 | 0.001 | 16.379 | 0.17 | 537 |
| | | 30 | 500 | 4847 | 10 | 499 | 4038 | 611 | 2454 | 301.64 | 0.014 | 20.297 | 16.473 | 1411 |
| | Uniform | 44 | 100 | 999 | 3 | 45 | 95 | 3 | 5 | 66.67 | 0 | 0.113 | 0.01 | 5 |
| | | 50 | 100 | 999 | 4 | 45 | 95 | 3 | 5 | 66.67 | 0.001 | 0.125 | 0.012 | 5 |
| | | 52 | 200 | 1960 | 4 | 184 | 1795 | 5 | - | - | 0.002 | 1.276 | 1.223 | 8 |
| | | 54 | 500 | 4868 | 4 | 147 | 260 | 3 | 5 | 66.67 | 0.001 | 0.78 | 0.106 | 4 |
| | | 55 | 100 | 999 | 10 | 70 | 188 | 3 | 6 | 100.00 | 0 | 0.251 | 0.044 | 6 |
| | | 56 | 100 | 999 | 10 | 90 | 895 | 3 | 27 | 800.00 | 0.009 | 2.768 | 1.363 | 7 |
| | | 57 | 200 | 1960 | 10 | 184 | 1795 | 5 | 26 | 420.00 | 0.004 | 0.931 | 0.876 | 6 |
| | | 58 | 200 | 1960 | 10 | 184 | 1795 | 5 | - | - | 0.04 | 8.218 | 10.999 | 12 |
| | | 59 | 500 | 4868 | 10 | 69 | 93 | 3 | 4 | 33.33 | 0.001 | 1.429 | 0.037 | 4 |
| | | 60 | 500 | 4868 | 10 | 469 | 4572 | 3 | 28 | 833.33 | 0.022 | 7.435 | 3.859 | 4 |

Desrochers, M. An algorithm for the shortest path problem with resource constraints. Technical Report G-88-27, GEARD, Ecole des H.E.C., Montreal, Canada, 1988.

Desrosiers, J., Dumas, Y., Solomon, M. M., and Soumis, F. Chapter 2 time constrained routing and scheduling. In M.O. Ball, C. M., T.L. Magnanti and Nemhauser, G., editors, *Network Routing*, volume 8 of *Handbooks in Operations Research and Management Science*, pages 35–139. Elsevier, 1995.

Dumitrescu, I. and Boland, N. Improved preprocessing, labeling and scaling algorithms for the weight-constrained shortest path problem. *Networks*, 42(3):135–153, 2003.

Eggenberg, N., Salani, M., and Bierlaire, M. Robust optimization with recovery: application to shortest paths and airline scheduling. In *7th Swiss Transport Research Conference*, Sep 2007.

Gabrel, V., Murat, C., and Wu, L. New models for the robust shortest path problem: complexity, resolution and generalization. *Annals of Operations Research*, 207(1):97–120, 2013.

Goetzmann, K.-S., Stiller, S., and Telha, C. Optimization over integers with robustness in cost and few constraints. In Solis-Oba, R. and Persiano, G., editors, *Approximation and Online Algorithms*, volume 7164 of *Lecture Notes in Computer Science*, pages 89–101. 2012.

Gounaris, C. E., Wiesemann, W., and Floudas, C. A. The robust capacitated vehicle routing problem under demand uncertainty. *Operations Research*, 61(3):677–693, 2013.

Handler, G. and Zang, I. A dual algorithm for the constrained shortest path problem. *Networks*, 10(4):293–309, 1980.

Hassin, R. Approximation schemes for the restricted shortest path problem. *Mathematics of Operations Research*, 17(1):36–42, 1992.

Irnich, S. and Desaulniers, G. Shortest path problems with resource constraints. In Desaulniers, G., Desrosiers, J., and Solomon, M. M., editors, *Column Generation*, pages 33–65. Springer US, 2005.

Irnich, S. Resource extension functions: properties, inversion, and generalization to segments. *OR Spectrum*, 30(1):113–148, 2008.

Karasan, O., Pinar, M., and H., Y. The robust shortest path problem with interval data. *Technical Report*, 2001.

Kelley, J., J. E. The cutting-plane method for solving convex programs. *Journal of the Society for Industrial and Applied Mathematics*, 8(4):703–712, 1960.

Lorenz, D. H. and Raz, D. A simple efficient approximation scheme for the restricted shortest path problem. *Operations Research Letters*, 28(5):213–219, 2001.

Lozano, L. and Medaglia, A. L. On an exact method for the constrained shortest path problem. *Computers & Operations Research*, 40(1):378–384, 2013.

Lu, D. and Gzara, F. The robust crew pairing problem: model and solution methodology. *Journal of Global Optimization*, 62(1):29–54, 2015.

Mehlhorn, K. and Ziegelmann, M. Resource constrained shortest paths. In Paterson, M., editor, *Algorithms - ESA 2000*, volume 1879 of *Lecture Notes in Computer Science*, pages 326–337. Springer Berlin Heidelberg, 2000. ISBN 978-3-540-41004-1.

Montemanni, R. and Gambardella, L. An exact algorithm for the robust shortest path problem with interval data. *Computers & Operations Research*, 31(10):1667 – 1680, 2004.

Pessoa, A. A., Di Puglia Pugliese, L., Guerriero, F., and Poss, M. Robust constrained shortest path problems under budgeted uncertainty. *Networks*, 66(2):98–111, 2015.

Poss, M. Robust combinatorial optimization with variable cost uncertainty. *European Journal of Operational Research*, 237(3):836–845, 2014a.

Poss, M. Robust combinatorial optimization with variable cost uncertainty. *European Journal of Operational Research*, 237(3):836 – 845, 2014b.

Pugliese, L. D. P. and Guerriero, F. A survey of resource constrained shortest path problems: Exact solution approaches. *Networks*, 62(3):183–200, 2013.

Resende, M. The robust shortest path problem with discrete data. *PhD dissertation*, 2015. URL `https://estudogeral.sib.uc.pt/handle/10316/28273`.

Ribeiro, C. C. and Minoux, M. A heuristic approach to hard constrained shortest path problems. *Discrete Applied Mathematics*, 10(2):125 – 137, 1985.

Santos, L., Coutinho-Rodrigues, J., and Current, J. R. An improved solution algorithm for the constrained shortest path problem. *Transportation Research Part B: Methodological*, 41(7):756 – 771, 2007.

Soyster, A. L. Convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations Research*, 21(5):1154–1157, 1973.

Sungur, I., Ordóñez, F., and Dessouky, M. A robust optimization approach for the capacitated vehicle routing problem with demand uncertainty. *IIE Transactions*, 40(5):509–523, 2008.

Yen, J. Y. Finding the k shortest loopless paths in a network. *Management Science*, 17(11):712–716, 1971.

Yu, G. and Yang, J. On the robust shortest path problem. *Comput. Oper. Res.*, 25(6):457–468, 1998.

Zhu, X. and Wilhelm, W. E. A three-stage approach for the resource-constrained shortest path as a sub-problem in column generation. *Computers & Operations Research*, 39(2):164–178, 2012.

**Appendix**

# A. Algorithms

The content in this appendix includes all the algorithms used in the paper. Moreover, for the robust RCSPP, functions $Robust\_REF$, $Robust\_Feasible$ and $Robust\_Dominance$ are modifications of $REF$, $Feasible$ and $Dominance$, respectively.

**Algorithm 3** Resource based reduction

**for** $r = 2, ..., |R|$ **do**
Set $t_{ij1} = t_{ijr}, \forall (i, j) \in A$
Solve the resulting shortest path problem [SPP] using Dijkstra's algorithm
**for all** $i \in N \backslash \{o, d\}$ **do**
**if** $D^r_{oi} + D^r_{id} > B_r - H_r$ **then**
Remove node $i$, and its out-going and in-coming arcs.
**end if**
Update $N$ and $A$
**end for**
**for all** $(i, j) \in A$, **do**
**if** $D^r_{oi} + t_{ijr} + D^r_{jd} > B_r - H_r$ **then**
Remove arc $(i, j)$
**end if**
**if** node $i$ has no out-going arcs **then**
Remove node $i$, and its in-coming arcs
**end if**
**if** node $j$ has no in-coming arcs **then**
Remove node $j$, and its out-going arcs.
**end if**
Update $N$ and $A$
**end for**
**end for**

**Algorithm 4** Lagrangian based reduction

**for all** $i \in N \backslash \{o, d\}$ **do**
  **if** $L_{oi}(\mu) + L_{id}(\mu) + H_1 - \sum_{r=2}^{|R|} \mu_r(B_r - H_r) > Z^{UB}$ **then**
    Remove vertex $i$, and its out-going and in-coming arcs.
  **end if**
  Update $N$ and $A$
**end for**
**for all** $(i, j) \in A$ **do**
  **if** $L_{oi}(\mu) + \bar{t}_{ij1} + L_{jd}(\mu) + H_1 - \sum_{r=2}^{|R|} \mu_r(B_r - H_r) > Z^{UB}$ **then**
    remove arc $(i, j)$
  **end if**
  **if** node $i$ has no out-going arcs **then**
    Remove node $i$ and its in-coming arcs
  **end if**
  **if** node $j$ has no in-coming arcs **then**
    Remove node $j$ and its out-going arcs
  **end if**
  Update $N$ and $A$.
**end for**

**Algorithm 5** Graph reduction

**while** $G$ is reduced **do**
  Implement Algorithm 3:
    any path found in the process is kept for the initiation of Kelley's cutting plane algorithm.
    any path feasible to problem [P] is used to update $Z^{UB}$;
  Implement Kelley's cutting plane algorithm to solve the Lagrangian dual problem of [PR3].
  Use the $\mu_r$ at the end of Kelley's cutting plane algorithm to calculate the Lagrangian cost for each arc.
  Implement Algorithm 4.
**end while**

**Algorithm 6** The sequential algorithm

**for** $h_1^e = h_1^{\Gamma_1}, ..., h_1^{|A|+1}$ and $h_1^e \neq h_1^{e+1}, e = \Gamma_1, ..., |A|$ **do**
    **if** current best solution has an arc with a cost greater than $h_1^e$ **then**
        Find arc $(i, j) \in A$ with arc cost deviation greater than $h_1^e$
        Set $t_{ij1} = t_{ij1} + h_{ij1} - h_1^e$
        **for** $r = 2, ..., |R|$ **do**
            **for** $h_r^e = h_r^{\Gamma_r}, ..., h_r^{|A|+1}$ and $h_r^e \neq h_r^{e+1}, e = \Gamma_r, ..., |A|$ **do**
                Find arc $(i, j) \in A$ with $t_{ijr} > h_r^e$
                Set $t_{ijr} = t_{ijr} + t_{ijr} - h_r^e$
                **if** $r = |R|$ **then**
                    Implement Algorithm 1 and find the optimal path $\overline{\mathbf{y}}$.
                    **if** the cost of $\overline{\mathbf{y}}$ is less than current best solution **then**
                        Update current best solution.
                    **end if**
                **end if**
            **end for**
        **end for**
    **end if**
**end for**

**Algorithm 7** Modified label correcting for $\bar{G}$ with $h_{ijr}, (j,i) \in \bar{A}$ as arc costs

Definition:
$V(ir)$: a vector of size $\Gamma_r$
$f$: an element in $V(ir)$, where the first value records the variation, the second and third values in $(,)$ record the corresponding arc.
$arc(f)$: returns the arc recorded in $f$.
$var(f)$: returns the variation recorded in $f$.
$M$: a priority queue of nodes with top node $i$ having the maximum $hmin_i$ .
Initialization:
set $h_{ijr}$ as the arc cost for each $(j,i) \in \bar{A}$
**for all** $i \in N$ **do**
    create a vector $V(ir)$.
    set all elements in $V(i)$ to $f = \{0, (0,0)\}$
**end for**
create a queue $M$ with no duplicated elements
enqueue $d$ onto $M$
**while** $M$ is not empty **do**
    $j \leftarrow M$.top()
    **for all** $(j,i) \in \bar{A}$ **do**
        **for all** $f' \in V(jr) \bigcup \{h_{ijr}, (j,i)\}$ **do**
            **for all** $f \in V(ir)$ **do**
                **if** $arc(f')$ is $arc(f)$ **then**
                    break
                **else if** $var(f') > var(f)$ **then**
                    insert $f'$ before $f$
                    pop out the end element of $V(ir)$
                    enqueue $i$ onto $M$
                    break
                **end if**
            **end for**
        **end for**
    **end for**
**end while**
Return $\Delta_{ir} = \{(j,i) | h_{ijr} > 0, \{h_{ijr}, (j,i)\} \in V(ir)\}, \forall i \in N$.