Robust Critical Node Selection by Benders Decomposition

Joe Naoum-Sawaya
Ivey Business School, Western University, 1255 Western Road, London, Ontario N6G 0N1, Canada, jnaoumsa@uwaterloo.ca

Christoph Buchheim
Fakultät für Mathematik, Technische Universität Dortmund, Vogelpothsweg 87, 44227 Dortmund, Germany, christoph.buchheim@tu-dortmund.de

The critical node selection problem (CNP) has important applications in telecommunication, supply chain design, and disease propagation prevention. In practice, the weights on the connections are often uncertain or hard to estimate. For this reason, robust optimization approaches have been considered recently for CNP. In this article, we address very general uncertainty sets, only requiring a linear optimization oracle for the set of potential scenarios. In particular, we can deal with discrete scenario based uncertainty, gamma uncertainty, and ellipsoidal uncertainty. For this general class of robust critical node selection problems (RCNP), we propose an exact solution method based on Benders decomposition. The Benders subproblem, which in our approach is a robust optimization problem, is efficiently solved by applying the Floyd-Warshall algorithm. The presented approach is tested on 384 instances based on Forest-Fire, Barabási-Albert, Erdös-Rényi, and Watts-Strogatz graphs with different number of nodes and edges, where running times are compared to CPLEX being directly applied to the robust problem formulation. The computational results show the advantage of the proposed approach in handling the uncertainty thus outperforming CPLEX most notably for the ellipsoidal uncertainty cases.

Key words: Robust Optimization, Critical Nodes, Benders Decomposition.

1. Introduction

Given an undirected graph $G(V, E)$, the objective of the critical node selection problem (CNP) is to find a subset $S \subseteq V$ of at most K nodes such that the residual graph obtained by eliminating the selected nodes has a minimum number of connected node pairs $\{i, j\}$. A pair $\{i, j\}$ is said to be connected if there exists a path linking nodes i and j. In the weighted version of the CNP, we are additionally given a non-negative weight w_{ij} on each pair $\{i, j\}$ and aim at a minimum total weight of connected node pairs.

The CNP has several applications in practice. It is important in assessing the robustness of a network such as a supply chain, where the objective is to measure the number of connections that are still reliable after a "smart" attacker damages the most critical nodes in the network (Arulselvan et al. 2007). Another application arises in health-care particularly in virus immunization where limited immunization resources are to be distributed. The goal is to identify the individuals to be vaccinated in order to reduce the overall transmissibility of a virus and to control the outbreak of an epidemic (Tao et al. 2006). Similarly, the spread of a virus over a telecommunication network can be limited by identifying the critical nodes of the graph and taking them offline (Cohen et al. 2003). Another application arises in wireless network jamming, where the objective is to efficiently prevent communication over a wireless network by finding the critical nodes and jamming them (Commander et al. 2007).

Arulselvan et al. (2007) presented an integer programming (IP) formulation for the CNP and proposed a heuristic that is based on finding an independent set of nodes from the graph and then improving it through a simple 2-exchange local search. Di Summa et al. (2011) showed that the CNP is $\mathcal{NP}$-hard in the general case and then considered graphs with the special tree structure and showed that the resulting CNP is polynomially solvable for the case of unit edge weights and proposed a dynamic programming approach for its solution. Shen and Smith (2012) presented a polynomial-time dynamic programming algorithm for solving the CNP on tree structures and on series-parallel graphs. Addis et al. (2013) also used dynamic programming to solve special cases of the CNP. Based on the IP model presented by Arulselvan et al. (2007), Di Summa et al. (2012) derived valid inequalities based on cliques and cycles in the graph. Di Summa et al. (2012) also proposed an integer programming formulation with a non-polynomial number of constraints and a branch-and-cut framework as a solution methodology. Valid inequalities based on dynamic programming solutions are also studied by Shen et al. (2012). Most recently, Ventresca (2012) proposed a simulated annealing heuristic with incremental learning targeting large graphs of sizes between 1000 and 5000 nodes.

While this paper focuses on CNP, we note that several important variants to the CNP have also been proposed. Arulselvan et al. (2011) consider the problem of minimizing the number of vertices whose deletion results in disconnected components which are constrained by a given cardinality. Shen and Smith (2012) consider the problem of minimizing the number of nodes that belong to the largest maximal connected subgraph. Another variant of CNP is also proposed by Shen and Smith (2012) where the number of disconnected components is maximized. Other closely related problems to CNP are the k-cut problem that partitions a vertex set of a graph into several disjoint subsets so as to minimize the total weight of the edges between the disjoint subsets (Fan and Pardalos 2010a, Ghaddar et al. 2011) and the minimum multi-cut problem that minimizes the weight of edges whose removal disconnects each of the of source-sink node pairs (Garg et al. 1996).

The importance of a connection, modeled by the weight w_{ij} of a pair $\{i, j\}$, is often not known exactly in real-world applications. Such data uncertainties can be addressed by means of either robust or stochastic optimization. The latter approach requires a precise knowledge about probability distributions of the weights. Moreover it usually leads to optimization problems that are far too hard to be solved in practice. On the other hand, the robust optimization approach has already been applied successfully to the CNP by Fan and Pardalos (2010b) and Fan et al. (2012).

In the approach that is proposed in this paper, a set of potential scenarios, the so-called uncertainty set, is defined. The objective of the robust critical node selection problem (RCNP) is then to minimize the maximum total weight of a feasible solution over all scenarios, thus aiming at a worst-case optimal solution. The choice of an appropriate uncertainty set is crucial, as a too large set leads to bad solutions in the average case and a too small set does not sufficiently take the uncertainty into account. Moreover, in most approaches the choice of the uncertainty set has a dramatic impact on the solvability of the resulting model, both from a theoretical and a practical point of view.

In this article, we present an approach for the RCNP that can deal with all common classes of uncertainty sets, including discrete scenario based uncertainty, gamma uncertainty, and ellipsoidal uncertainty which are often used in the literature. This is achieved through a Benders decomposition approach that explicitly handles the uncertainty in the Benders subproblem through an efficient optimization oracle. The structure of the Benders subproblem is also exploited and a specialized polynomial time algorithm that is based on the Floyd-Warshall algorithm (Floyd 1962) is proposed in order to achieve maximum speedup. Problems based on Forest-Fire, Barabási-Albert, Erdös-Rényi, and Watts-Strogatz graphs are solved and a comparison with CPLEX is conducted.

It turns out that our approach is particularly effective for ellipsoidal uncertainties, where it outperforms CPLEX on several instances. Compared to all other classes of uncertainty considered, the ellipsoidal case is usually the computationally most expensive one, but also the most relevant one in practice: under the natural assumption that all weights are jointly normally distributed, one can show that the confidence regions form ellipsoids, defined by the expected values and the covariance matrix of the weights. Any outer polyhedral approximation of the confidence regions, e.g., using interval or gamma uncertainty, therefore necessarily leads to overly conservative solutions: the worst case is (under weak conditions) always attained in one of the vertices, which do not belong to the confidence region and are hence less likely to appear under the assumption of a normal distribution.

Robust optimization is a common approach for network planning problems under uncertainty (Li et al. 2011). Benders decomposition has also been used as an approach to solve robust optimization problems with interval uncertainty (Montemanni 2006, Siddiqui et al. 2011, 2014) and ellipsoidal uncertainty (Saito and Murota 2007). Besides Benders decomposition, other methods such as approximation through piecewise linear functions and Frank–Wolfe iterative algorithms have been applied to approximate robust optimization problems with ellipsoidal uncertainty (Bertsimas and Sim 2004b). Besides the approach of Fan and Pardalos (2010b) which proposes a method for RCNP with gamma uncertainty, to the best of our knowledge, no other problem-specific approaches for the RCNP under uncertainty have been proposed in the literature before.

Contribution of this paper. The main contributions of this paper are (a) a robust optimization model for the CNP under very general uncertainty sets, including ellipsoidal uncertainty, (b) a novel approach to address this problem through Benders decomposition while explicitly dealing with robustness in the subproblems, and (c) a computationally efficient implementation that is capable of solving problems of reasonable size.

Outline of this paper. Following this introductory section, Section 2 provides the mathematical formulation. Sections 3 presents the Benders decomposition approach. Computational results are presented in Section 4. Finally, a conclusion is given in Section 5.

2. Problem Formulation

We consider an undirected weighted graph $G(V, E)$, with node set V and edge set E. A node pair $\{i, j\}$ is connected if there exists a path that links nodes i and j. To formulate the critical node selection problem, we define the following decision variables

$$x_i = \begin{cases} 1 & \text{if node } i \text{ is deleted from the graph} \\ 0 & \text{otherwise,} \end{cases}$$

$$y_{ij} = \begin{cases} 1 & \text{if node pair } \{i,j\} \text{ is not connected in the residual graph} \\ 0 & \text{otherwise.} \end{cases}$$

We consider an uncertainty set $\mathcal{U}$ that contains all possible realizations of edge weights w. We make the reasonable assumption that $w \geq 0$ for all $w \in \mathcal{U}$. Apart from this natural assumption, we only require that a linear optimization oracle is given for $\mathcal{U}$. The robust critical node selection problem is then formulated as follows:

$$\text{[RCNP]: } \max_{x,y} \ \min_{w \in \mathcal{U}} \ w^\top y \tag{1}$$

$$\text{s.t. } e^\top x \leq K, \tag{2}$$

$$y_{ij} \leq x_i + x_j, \ \forall (i,j) \in E, \tag{3}$$

$$y_{ij} + y_{jk} - y_{ik} \geq 0, \forall i,j,k \in V, \ i < j < k, \tag{4}$$

$$y_{ij} - y_{jk} + y_{ik} \geq 0, \forall i,j,k \in V, \ i < j < k, \tag{5}$$

$$-y_{ij} + y_{jk} + y_{ik} \geq 0, \forall i,j,k \in V, \ i < j < k, \tag{6}$$

$$0 \leq y_{ij} \leq 1, \forall i,j \in V, \ i < j, \tag{7}$$

$$x_i \in \{0,1\}, \forall i \in V. \tag{8}$$

In the non-robust case, i.e. when $\mathcal{U}$ is a singleton, this model is identical to the one presented by Di Summa et al. (2012). The objective function maximizes the worst case weight of disconnected node pairs, i.e., node pairs not connected by paths. Constraint (2) is a cardinality constraint limiting the number of deleted nodes to K; here e is the vector of all ones. Constraints (3) ensure that if a node pair $\{i,j\}$ is disconnected in the residual graph and there exists an edge (i,j) then either node i or node j should be deleted. Constraints (4)–(6) guarantee that for all triplets (i,j,k), if $\{i,j\}$ is connected and $\{j,k\}$ is connected, then $\{i,k\}$ should be connected. For ease of presentation, in the rest of the paper we refer to constraints (3)–(7) as

$$Ay \leq Bx + b.$$

We also note that the binary condition on the y_{ij} variables can be relaxed as discussed in Di Summa et al. (2012) thus permitting the application of Benders decomposition. In the following section, we briefly describe our Benders decomposition approach for problem [RCNP].

3. Benders Decomposition

Benders decomposition (Benders (1962) and Geoffrion (1972)) splits the variables in a mixed integer program into a linear subproblem and a mixed integer master problem that contains all the integer variables. The main motivation is having a subproblem that is significantly easier to solve. The procedure iterates between solving the subproblem and a relaxed version of the mixed integer master problem, until an optimal solution is reached.

In what follows, we describe the application of Benders decomposition to problem [RCNP]. For that, we write [RCNP] as follows:

$$\max_{\substack{e^\top x \le K \\ x \in \{0,1\}^V}} \max_{y} \min_{w \in \mathcal{U}} w^\top y \tag{9}$$

$$\text{s.t. } Ay \le Bx + b. \tag{10}$$

In order to generate optimality cuts for the Benders master problem, we derive the dual of the inner problem

$$\begin{aligned} \text{[PSP]: } f(\overline{x}) = \max_{y} \min_{w \in \mathcal{U}} \ & w^\top y \\ \text{s.t. } & Ay \le B\overline{x} + b. \end{aligned}$$

Note that for any set $\mathcal{U}$, the function $g(y) := \min_{w \in \mathcal{U}} w^\top y$ is concave, as it is defined as the point-wise minimum over linear functions in y. This leads to

THEOREM 1. *The dual of [PSP] is*

$$\begin{aligned} \text{[DSP]: } \min_{\lambda} \ & (B\overline{x} + b)^\top \lambda, \\ s.t. \ & A^\top \lambda \in \mathcal{U}, \\ & \lambda \ge 0, \end{aligned}$$

and strong duality holds.

Proof. The Lagrangian dual of [PSP] is

$$\min_{\lambda \ge 0} \max_{y} \Big(\min_{w \in \mathcal{U}} w^\top y + (B\overline{x} + b - Ay)^\top \lambda \Big).$$

Rearranging the variables we get

$$\min_{\lambda\geq 0}\left((B\overline{x}+b)^\top\lambda+\max_{y}\min_{w\in\mathcal{U}}(w-A^\top\lambda)^\top y\right) = \begin{aligned}[t]\min_{\lambda,w}\ & (B\overline{x}+b)^\top\lambda\\ \text{s.t. } & A^\top\lambda = w\\ & \lambda\geq 0,\ w\in\mathcal{U}\,.\end{aligned}$$

By concavity of g and since all constraints in [PSP] are linear, the Slater condition guarantees strong duality. □

The Benders optimality cut can now be derived from a given optimal solution $\overline{\lambda}$ of [DSP] as

$$\theta-(B^\top\overline{\lambda})^\top x\leq b^\top\overline{\lambda}\,.$$

Hence the Benders master problem is

$$[\text{BMP}]: \max_{x,\theta}\ \theta \tag{11}$$

$$\text{s.t. } e^\top x\leq K, \tag{12}$$

$$\theta-(B^\top\overline{\lambda})^\top x\leq b^\top\overline{\lambda}, \tag{13}$$

$$x_i\in\{0,1\},\ \forall i\in V, \tag{14}$$

where cuts (13) are generated iteratively. Solving the Benders master problem with an additional optimality cut, a new optimal solution $\overline{x}$ is obtained which in turn gives rise to a new optimality cut. This process is repeated iteratively until the master problem and subproblem solutions converge.

3.1. Solving the Primal Benders Subproblem

In this section, we present a specialized algorithm for solving the Benders subproblem [PSP]. We do not require $\overline{x}$ to be a binary vector. Fractional values of $\overline{x}$ appear when using the Analytic Center Cutting Plane Method (ACCPM) that improves over the classical Benders decomposition as discussed in Section 4.1. At the end of this section, we also provide a faster and simpler algorithm for the case of binary $\overline{x}$ vector.

The specialized algorithm for the fractional case is based on finding the shortest path between every pair of nodes in the complete graph $\overline{G}$ defined on the vertices of G. For that, let us define an initial set of weights h_{ij} for every node pair i,j in V such that

$$h_{ij}=\min\{1,\overline{x}_i+\overline{x}_j\},\ \forall(i,j)\in E,$$

$$h_{ij}=1,\ \forall(i,j)\notin E.$$

We then compute all pairwise shortest paths in $\overline{G}$ between every pair of nodes $\{i,j\}$ and finally set

$$\begin{aligned}\overline{y}_{ij} &= \text{shortestPath}(i,j)\\ &:= \min\left\{\textstyle\sum_{(i,j)\in E(P)} h_{ij} \mid P \text{ is any path from } i \text{ to } j \text{ in } \overline{G}\right\},\end{aligned}$$

where $E(P)$ denotes the set of edges of the path P. The pairwise shortest paths can be computed efficiently using the Floyd-Warshall algorithm which finds the shortest paths between all pairs of nodes in a total number of $O(|V|^3)$ comparisons where $|V|$ is the number of nodes in the graph.

In the following we provide a proof that the resulting solution $\overline{y}$ is an optimal solution to the nominal Benders subproblem for any fixed $w \geq 0$. We first show that $\overline{y}$ is feasible to [PSP] and finally show that it is an optimal solution.

THEOREM 2. *The solution $\overline{y}$ such that $\overline{y}_{ij} = \textit{shortestPath}(i,j)$ is feasible to [PSP].*

Proof: By definition, we have that $0 \leq h_{ij} \leq 1$ and $h_{ij} \leq \overline{x}_i + \overline{x}_j \; \forall (i,j) \in E$. Furthermore, $\overline{y}_{ij} \leq h_{ij}$ since $\overline{y}_{ij} = \text{shortestPath}(i,j)$. Therefore $\overline{y}_{ij} \leq \overline{x}_i + \overline{x}_j$ and $\overline{y}_{ij} \leq 1$ for all $(i,j) \in E$ and hence constraints (3) and (7) are satisfied. Additionally, for every set of nodes (i,j,k) we have that

$$\begin{aligned}\text{shortestPath}(i,k) &\leq \text{shortestPath}(i,j) + \text{shortestPath}(j,k),\\ \text{shortestPath}(j,k) &\leq \text{shortestPath}(i,j) + \text{shortestPath}(i,k),\\ \text{shortestPath}(i,j) &\leq \text{shortestPath}(i,k) + \text{shortestPath}(j,k),\end{aligned}$$

hence constraints (4), (5), (6) are satisfied respectively. Since all the constraints of [PSP] are satisfied, then $\overline{y}_{ij}$ is feasible to [PSP]. □

THEOREM 3. *The solution $\overline{y}$ such that $\overline{y}_{ij} = \textit{shortestPath}(i,j)$ is optimal for [PSP].*

Proof: Let y be any feasible solution for [PSP]. We show that $y \leq \overline{y}$. This implies that $w^\top y \leq w^\top \overline{y}$ for all $w \in \mathcal{U}$ since $w \geq 0$ and therefore

$$\min_{w\in\mathcal{U}} w^\top y \;\leq\; \min_{w\in\mathcal{U}} w^\top \overline{y},$$

showing that $\overline{y}$ is an optimal solution for [PSP].

So fix $i,j \in V$ with $i<j$. If $\overline{y}_{ij}=1$, we obtain $y_{ij} \leq \overline{y}_{ij}$ from (7), so that we may assume $\overline{y}_{ij}<1$. Then, by definition of the weights h_{ij}, the shortest path from i to j contains only edges belonging to E, say, $(i,k_1),\ldots,(k_m,k_{m+1}),\ldots,(k_n,j)$. We then have

$$\begin{aligned}\overline{y}_{ij} &= h_{ik_1}+\cdots+h_{k_m k_{m+1}}+\cdots+h_{k_n j}\\ &= \min\{1,\overline{x}_i+\overline{x}_{k_1}\}+\cdots+\min\{1,\overline{x}_{k_m}+\overline{x}_{k_{m+1}}\}+\cdots+\min\{1,\overline{x}_{k_n}+\overline{x}_j\}.\end{aligned}$$

As y is feasible for [PSP], we have

$$y_{ik_1} \leq \min\{1,\overline{x}_i+\overline{x}_{k_1}\} \tag{15}$$

$$\vdots$$

$$y_{k_m k_{m+1}} \leq \min\{1,\overline{x}_{k_m}+\overline{x}_{k_{m+1}}\} \tag{16}$$

$$\vdots$$

$$y_{k_n j} \leq \min\{1,\overline{x}_{k_n}+\overline{x}_j\} \tag{17}$$

by (3) and (7) and

$$y_{ij} \leq y_{ik_n}+y_{k_n j} \tag{18}$$

$$y_{ik_n} \leq y_{ik_{n-1}}+y_{k_{n-1}k_n} \tag{19}$$

$$\vdots$$

$$y_{ik_2} \leq y_{ik_1}+y_{k_1 k_2} \tag{20}$$

by constraints (4)–(6). Adding constraints (18),(19), ..., (20) we get

$$y_{ij} \leq y_{ik_1}+\cdots+y_{k_m k_{m+1}}+\cdots+y_{k_n j}. \tag{21}$$

Substituting (15)–(17) in (21) we get

$$\begin{aligned}y_{ij} &\leq \min\{1,\overline{x}_i+\overline{x}_{k_1}\}+\cdots+\min\{1,\overline{x}_{k_m}+\overline{x}_{k_{m+1}}\}+\cdots+\min\{1,\overline{x}_{k_n}+\overline{x}_j\}\\ &= \overline{y}_{ij}\end{aligned}$$

as desired. □

COROLLARY 1. *The optimal value of [PSP] is equal to*

$$\min\ \overline{y}^\top w \tag{22}$$

$$s.t.\ w \in \mathcal{U}\,. \tag{23}$$

For solving problem (22)–(23), we can use the linear optimization oracle that we assume is available for $\mathcal{U}$. The running time for computing $\overline{y}$ is $O(|V|^3)$ as mentioned earlier, however the running time for computing the optimal value of [PSP] depends on how quickly we can minimize the linear objective (22) over the set $\mathcal{U}$. For the case of binary $\overline{x}$, the computation of $\overline{y}$ can be accelerated:

COROLLARY 2. *Assume that $\overline{x} \in \{0,1\}^n$. Then a maximizer of [PSP] can be computed in $O(|V|^2)$ time.*

Proof: By Theorem 3, an optimal solution is obtained by setting $\overline{y}_{ij} = \text{shortestPath}(i,j)$. Since $\overline{x} \in \{0,1\}^n$, also the weights h_{ij} are binary. Thus $\overline{y}_{ij} = 0$ if and only if there is a path from i to j consisting of edges $(k,l) \in E$ with $h_{kl} = 0$. From the definition, we have $h_{kl} = 0$ if and only if $\overline{x}_k = \overline{x}_l = 0$. We conclude that $\overline{y}_{ij} = 0$ if and only if there is a path from i to j in G containing only nodes k with $\overline{x}_k = 0$, otherwise $\overline{y}_{ij} = 1$.

It thus suffices to identify all such node pairs $\{i,j\}$ in $O(|V|^2)$ total time, which can be done as follows: we first remove all nodes i from G such that $\overline{x}_i = 1$, along with their incident edges, resulting in a graph $\tilde{G}$. Now $\overline{y}_{ij} = 0$ if and only if i and j belong to the same connected component of $\tilde{G}$. It thus suffices to determine the connected components of $\tilde{G}$, which can be done by a depth-first-search in linear time in the number of vertices and edges of $\tilde{G}$; see, e.g., Hopcroft and Tarjan (1973). □

3.2. Solving the Dual Subproblem

As discussed earlier, to derive the Benders optimality cuts, the dual optimal solution $\overline{\lambda}$ must be obtained.

LEMMA 1. *Let $\overline{w}$ be an optimal solution of problem (22)–(23), then an optimal solution of [DSP] can be computed by solving*

$$\min_{\lambda} \ (B\overline{x} + b)^\top \lambda \tag{24}$$

$$s.t. \ A^\top \lambda = \overline{w} \tag{25}$$

$$\lambda \geq 0. \tag{26}$$

Proof. First note that

$$\begin{array}{llllll} \max_y \min_{w \in \mathcal{U}} \ w^\top y & = & \min_{w \in \mathcal{U}} \ \overline{y}^\top w & = \overline{w}^\top \overline{y} = & \max_y & \overline{w}^\top y \\ \text{s.t.} \quad Ay \leq B\overline{x} + b & & & & \text{s.t.} & Ay \leq B\overline{x} + b\,. \end{array}$$

The first equation follows from Corollary 1, the second one from the definition of $\overline{w}$, and the third one from the optimality of $\overline{y}$ for each $w \geq 0$. Using strong duality, we obtain that the corresponding dual problems have the same optimal values as well, i.e., the optimal value of [DSP] agrees with the optimal value of problem (24)–(26).

Now let λ be any optimal solution for (24)–(26). By (25)–(26) and since $\overline{w} \in \mathcal{U}$, it follows that λ it is also feasible for [DSP]. By the reasoning above, the objective value of λ agrees with the optimal value of [DSP], hence λ is optimal also for [DSP]. □

Lemma 1 shows that solving [DSP] can be reduced to solving a linear program once the primal solution and the corresponding $\overline{w}$ have been computed. In summary, the only step in the algorithm that requires dealing with $\mathcal{U}$ explicitly is the solution of (22)–(23) to compute $\overline{w}$.

In fact, an optimal solution of this linear program can be directly derived from an optimal solution of the primal problem [PSP]. For this, assume that an optimal solution of [PSP] is computed as discussed in Section 3.1. For the following, for a given path P, denote its vertex set by $V(P)$ and its edge set by $E(P)$.

LEMMA 2. *In $O(|V|^3)$ time, one can compute shortest paths P_{ij} for each pair of nodes i, j in $\overline{G}$ with respect to h such that the following conditions are satisfied:*

(A1) No edge $\{i,j\}$ with $h_{ij} = 1$ is contained in any shortest path except for possibly P_{ij}.

(A2) If $|E(P_{ij})| = 1$ and $i, j \in V(P_{kl})$, then $\{i,j\} \in E(P_{kl})$.

Proof: We first compute all shortest paths using the Floyd-Warshall algorithm. Then, by definition of the algorithm, we have $P_{ij} \subseteq P_{kl}$ whenever $i, j \in V(P_{kl})$. In particular, this implies (A2). Now if $(i,j) \in E(P_{kl})$ and $h_{ij} = 1$, we have $h_{kl} \leq 1 \leq \text{shortestPath}(k,l)$, so that we can replace the path $E(P_{kl})$ by the single edge (k,l) wherever it appears as a subpath in any shortest path P_{ij}. Performing all such replacements, we obtain (A1) and preserve (A2). Using the data structures of the Floyd-Warshall algorithm, the replacements can be done in $O(|V|^3)$ total time. □

THEOREM 4. *Given $\bar{w}_{ij}$ for all pairs $\{i,j\}$, a Benders optimality cut can be obtained as*

$$\theta - \sum_{h_{ij}<1} \Big(\sum_{E(P_{kl}) \ni \{i,j\}} \overline{w}_{kl} \Big) (x_i + x_j) \leq \sum_{h_{ij}=1,\ |E(P_{ij})|=1} \overline{w}_{ij} \, . \tag{27}$$

Proof: The Benders primal subproblem reads

$$
\begin{array}{llll}
\max_y \overline{w}^\top y & & & \\
\text{s.t.} & y_{ij} \le \overline{x}_i + \overline{x}_j & \forall (i,j) \in E & [\lambda^1_{ij}] \\
& y_{ij} \le 1 & \forall i,j \in V,\ i<j & [\lambda^2_{ij}] \\
& y_{ij} - y_{jk} - y_{ik} \le 0 & \forall i,j,k \in V,\ i<j<k & [\beta^k_{ij}] \\
& -y_{ij} + y_{jk} - y_{ik} \le 0 & \forall i,j,k \in V,\ i<j<k & [\beta^i_{jk}] \\
& -y_{ij} - y_{jk} + y_{ik} \le 0 & \forall i,j,k \in V,\ i<j<k & [\beta^j_{ik}] \\
& y_{ij} \ge 0 & \forall i,j \in V,\ i<j &
\end{array}
$$

We obtain the Benders dual subproblem

$$
\begin{array}{rl}
\min_{\lambda,\beta} & \sum_{(i,j)\in E} (\overline{x}_i + \overline{x}_j)\lambda^1_{ij} + \sum_{\{i,j\}} \lambda^2_{ij} \\
\text{s.t.} & \lambda^1_{ij} + \lambda^2_{ij} + \sum_{k \ne i,j} (\beta^k_{ij} - \beta^j_{ik} - \beta^i_{jk}) \ge \overline{w}_{ij} \ \forall (i,j) \in E \\
& \lambda^2_{ij} + \sum_{k \ne i,j} (\beta^k_{ij} - \beta^j_{ik} - \beta^i_{jk}) \ge \overline{w}_{ij} \ \forall (i,j) \notin E \\
& \lambda, \beta \ge 0.
\end{array}
$$

Define

$$
\overline{\lambda}^1_{ij} := \begin{cases} 0 & \text{if } h_{ij} = 1 \\ \sum_{E(P_{kl}) \ni \{i,j\}} \overline{w}_{kl} & \text{otherwise} \end{cases}
$$

and

$$
\overline{\lambda}^2_{ij} := \begin{cases} \overline{w}_{ij} & \text{if } h_{ij} = 1 \text{ and } |E(P_{ij})| = 1 \\ 0 & \text{otherwise.} \end{cases}
$$

Moreover, define

$$
\overline{\beta}^k_{ij} := \frac{1}{2}\Big(\sum_{E(P_{il}) \ni (k,j)} \overline{w}_{il} + \sum_{E(P_{jl}) \ni (k,i)} \overline{w}_{jl} \Big)
$$

Here, $E(P_{il}) \ni (k,j)$ means that the path P_{il} contains the vertices k and j one after another, in this order when traversing P_{ij} from i to j. Note that $j = l$ is not excluded.

Since we assume $\overline{w} \ge 0$, we have $\overline{\lambda}, \overline{\beta} \ge 0$. We next show that $\overline{\lambda}, \overline{\beta}$ satisfies all constraints of the dual problem. First note that $\sum_{k \ne i,j} \overline{\beta}^k_{ij}$ is half the sum of all $\overline{w}_{st}$ such that $i, j \in V(P_{st})$ but $\{i,j\} \notin E(P_{st})$ and such that exactly one of the end vertices of the path is i or j, plus $\overline{w}_{ij}$ if $|E(P_{ij})| \ge 2$. Moreover, $\sum_{k \ne i,j} \overline{\beta}^j_{ik}$ is half the sum of all $\overline{w}_{it}$ such that $j \in V(P_{it})$ but $\{i,j\} \notin E(P_{it})$, plus half the sum of all $\overline{w}_{st}$ such that $t \ne j$ and $\{i,j\} \in E(P_{st})$ (analogously for $\sum_{k \ne i,j} \overline{\beta}^j_{ik}$ by symmetry).

<u>*Case* $h_{ij}=1$*:*</u> In this case, no shortest path except for possibly P_{ij} contains edge $\{i,j\}$ by (A1). If $|E(P_{ij})|=1$, then by (A2) all shortest paths containing both i and j must contain edge $\{i,j\}$, so it follows that no shortest path except for P_{ij} contains both i and j. Hence $\overline{\beta}_{ij}^k=\overline{\beta}_{ik}^j=\overline{\beta}_{jk}^i=0$ for all $k\neq i,j$ and $\overline{\lambda}_{ij}^2=\overline{w}_{ij}$. Otherwise, $\sum_{k\neq i,j}(\overline{\beta}_{ij}^k-\overline{\beta}_{ik}^j-\overline{\beta}_{jk}^i)=\overline{w}_{ij}$, as all other terms $\overline{w}_{kl}$ cancel out each other, and $\overline{\lambda}_{ij}^2=0$. If $(i,j)\in E$, we have $\overline{\lambda}_{ij}^1=0$, so in both cases the constraint corresponding to y_{ij} in the dual problem is tight.
<u>*Case* $h_{ij}<1$*:*</u> Here we have $(i,j)\in E$. If $|E(P_{ij})|=1$, then again all shortest paths containing i and j must contain the edge $\{i,j\}$ by (A2), thus $\sum_{k\neq i,j}\overline{\beta}_{ij}^k=0$ and $\overline{\lambda}_{ij}^1\geq\sum_{k\neq i,j}(\overline{\beta}_{ik}^j+\overline{\beta}_{jk}^i)+\overline{w}_{ij}$. Otherwise, $\overline{\lambda}_{ij}^1=0$ by (A1) and $\sum_{k\neq i,j}\overline{\beta}_{ij}^k\geq\sum_{k\neq i,j}(\overline{\beta}_{ik}^j+\overline{\beta}_{jk}^i)+\overline{w}_{ij}$.

It remains to show that the objective function values of $\overline{y}$ in [PSP] and of $\overline{\lambda},\overline{\beta}$ in the dual problem agree, then $\overline{\lambda},\overline{\beta}$ is an optimal dual solution. We have

$$\begin{aligned}
\sum_{(i,j)\in E}(\overline{x}_i+\overline{x}_j)\overline{\lambda}_{ij}^1+\sum_{\{i,j\}}\overline{\lambda}_{ij}^2 &= \sum_{h_{ij}<1}(\overline{x}_i+\overline{x}_j)\sum_{E(P_{kl})\ni\{i,j\}}\overline{w}_{kl}+\sum_{\substack{h_{ij}=1\\|E(P_{ij})|=1}}\overline{w}_{ij}\\
&= \sum_{h_{ij}<1}h_{ij}\sum_{E(P_{kl})\ni\{i,j\}}\overline{w}_{kl}+\sum_{\substack{h_{ij}=1\\|E(P_{ij})|=1}}h_{ij}\overline{w}_{ij}\\
&= \sum_{|E(P_{ij})|=1}h_{ij}\overline{w}_{ij}+\sum_{|E(P_{kl})|\neq1}\sum_{(i,j)\in P_{kl}}h_{ij}\overline{w}_{kl}\\
&= \sum_{|E(P_{ij})|=1}\overline{y}_{ij}\overline{w}_{ij}+\sum_{|E(P_{kl})|\neq1}\overline{y}_{kl}\overline{w}_{kl}\\
&= \overline{w}^\top\overline{y}.
\end{aligned}$$

□

Hence at each iteration, the Benders subproblem is solved as discussed in Section 3.1 and a constraint of the form (27) is generated and added to the Benders master problem.

3.3. Complete Algorithm to Compute the Benders Cut

In summary, the following steps are performed to compute a Benders cut for a given solution $\overline{x}$ of the Benders master problem [BMP]:

1. Use the algorithm of Section 3.1 to compute an optimal solution $\overline{y}$ of [PSP] as well as the corresponding pair-wise shortest paths. This step can be done in $O(|V|^3)$ time.

2. Given $\overline{y}$, compute the corresponding worst case scenario $\overline{w}$. This step highly depends on the uncertainty set $\mathcal{U}$. In Section 4, we provide closed form formulas for all standard uncertainty sets, i.e. scenario based, gamma, and ellipsoidal uncertainty.

3. Given $\overline{w}$ and the shortest paths of Step 1, compute the Benders cut (27).

Note that in this approach, the computation of an optimal solution $\overline{y}$ of [PSP] is the main algorithmic step needed to compute the Benders cut.

4. Computational Testing

4.1. Application of a Modified ACCPM

The application of classical Benders decomposition often leads to poor performance due to slow convergence. A number of attempts have been made to alleviate this problem. The concept of pareto-optimal cuts, introduced by Magnanti and Wong (1981), looks for a non-dominated cut to add to the master problem when the subproblem has multiple optimal solutions. Recently, Naoum-Sawaya and Elhedhli (2013) proposed integrating the Benders framework within branch-and-bound and using the ACCPM to generate pareto-optimal cuts thus achieving very good speed-ups compared to classical Benders decomposition. The computational results conducted by Naoum-Sawaya and Elhedhli (2013) showed that the majority of the Benders cuts are generated at the root node of the branch-and-cut tree. We hence use a modified version of the approach presented in Naoum-Sawaya and Elhedhli (2013). Rather than generating the Benders cuts at all the nodes of the branch-and-cut tree, the Benders cuts are generated only at the root node and when a new incumbent solution is found. The modified algorithm can also be seen as warm starting the Benders decomposition algorithm with a set of cuts that are generated from the LP relaxation using the ACCPM.

4.2. Implementation

The proposed Benders decomposition algorithm was coded in C using CPLEX 12.5 callable libraries. The computations are carried out on a Lenovo C30 workstation with 2GHz CPU and 8GB of memory. We conduct the computational testing on four classes of commonly used complex network models: Forest-Fire, Barabási-Albert, Erdös-Rényi, and Watts-Strogatz. The characteristics of each of these networks can be found in Ventresca and Aleman (2014). A total of 32 networks of varying sizes with varying uncertainty sets were considered. The CPU time is limited to 10,000 seconds, after which the algorithm stops and returns the best solution found. All running times are compared to solving the problem [RCNP] directly by CPLEX, with default settings.

4.3. Results

We consider different uncertainty sets centered at w_0 where $(w_0)_{ij} = 1$ for all node pairs $\{i, j\}$. Parameter K in constraint (2) is set to 5. Computational results for scenario based uncertainty, gamma uncertainty, and ellipsoidal uncertainty are presented next. In all our test instances, we have $\mathcal{U} \subseteq \mathbb{R}^+$, i.e., all weights are non-negative in all scenarios.

The interval uncertainty case, i.e. $\mathcal{U} = \{w\colon w_0 - \hat{w} \le w \le w_0 + \hat{w}\}$, is not included in our computational testing since solving the problem with interval uncertainty is equivalent to solving the nominal problem

$$\begin{aligned} \max_{x,y} \; & (w_0 - \hat{w})^\top y \\ \text{s.t. } & (2) - (8). \end{aligned} \tag{28}$$

The following details are reported in the results tables 1–4:

$Vertices$: Number of vertices in the graph.
$Edges$: Number of edges in the graph.
$CUTS0$: Number of Benders cuts that are generated by ACCPM at the root node.
$CPU0$: Computational time spent on generating Benders cuts at the root node.
CPU_{FW} : Computational time spent on calculating shortest paths.
CPU_{WCW} : Computational time spent on calculating the worst case scenarios.
CPU_{BC} : Computational time spent on calculating Benders cuts.
CPU : Total computational time.
Opt : Objective function value.

All CPU times are given in seconds. The optimal objective function values are displayed for the cases that were solved within the time limit while the best lower and upper bounds are displayed for the cases where the time limit was exceeded. The instances that are marked by bold indicate the approach that outperforms the others in terms of computational time if the problem is solved within the time limit or in terms of the optimality gap if the time limit is reached.

4.3.1. Scenario based Uncertainty In this section, we consider scenario based robustness where the uncertainty set is

$$\mathcal{U} = \{w^1, w^2, \ldots, w^n\}.$$

Instance		Benders							CPLEX	
Vertices	Edges	CUTS0	CPU0	CPU_{FW}	CPU_{WCW}	CPU_{BC}	CPU	Opt	CPU	Opt
Forest-Fire										
40	186	192	0.9	0.0	0.2	0.0	**3.0**	629.29	21.4	629.29
	194	271	1.4	0.0	0.1	0.0	**3.4**	624.18	90.3	624.18
50	276	455	3.9	0.2	1.0	0.1	**15.5**	770.56	802.8	770.56
	312	403	3.6	0.1	0.7	0.0	**15.2**	708.31	905.2	708.31
60	226	140	1.4	0.1	0.7	0.0	**3.8**	1542.02	95.1	1542.02
	326	555	7.6	0.1	0.6	0.0	**25.3**	1111.12	1597.4	1111.12
70	292	397	6.3	0.2	0.9	0.1	**16.0**	1953.25	236.9	1953.25
	436	877	18.6	0.5	2.6	0.2	**95.2**	1059.80	3465.4	1059.80
	Average	411.3	5.5	0.2	0.9	0.1	22.2	Gap: 0%	901.8	Gap: 0%
Barabási-Albert										
40	76	203	0.3	0.1	0.8	0.0	**5.7**	419.49	292.0	419.49
	256	569	2.0	0.0	0.1	0.0	**29.8**	249.82	68.2	249.82
50	96	330	1.1	0.1	0.6	0.0	**5.8**	646.51	501.4	646.51
	336	1085	9.4	5.9	39.3	2.2	>10,000	(233.99;322.57)	>10,000	**(233.96;240.36)**
60	116	805	4.5	0.1	0.5	0.0	**156.2**	590.99	3,770.8	590.99
	416	940	12.7	1.4	8.1	0.6	**1,416.2**	337.5	1,989.8	337.50
70	136	449	2.8	0.5	2.8	0.2	**17.5**	1075.93	3,360.9	1075.93
	496	2563	112.2	1.8	9.3	0.6	>10,000	(398.03;453.15)	>10,000	**(398.03;422.25)**
	Average	868.0	18.1	1.2	7.7	0.5	>2703.9	Gap: 5%	>3747.9	Gap: 1%
Erdös-Rényi										
40	174	593	3.3	0.7	5.8	0.3	**182.2**	343.17	811.3	343.17
	208	860	7.1	1.3	10.9	0.7	1061.5	282.35	**779.3**	282.35
50	186	1014	10.9	0.8	5.6	0.4	**422.0**	625.50	1202.6	625.50
	370	1760	44.3	3.6	24.0	1.5	>10,000	(320.61;341.52)	**2367.7**	320.61
60	178	807	9.5	1.3	7.7	0.7	**470.7**	971.40	2833.2	971.40
	408	2271	96.4	2.1	12.5	0.9	>10,000	(390.57;552.72)	**6368.5**	390.57
70	190	999	17.1	2.1	10.8	0.9	**453.6**	1588.09	4083.6	1588.09
	648	2729	252.9	1.1	5.7	0.5	>10,000	(397.97;598.90)	>10,000	**(397.88;504.94)**
	Average	1379.1	55.2	1.6	10.4	0.7	>4073.8	Gap: 9%	>3555.8	Gap: 3%
Watts-Strogatz										
40	160	769	5.6	0.6	5.2	0.4	738.8	362.08	**612.6**	362.08
	240	1059	11.6	1.8	14.9	0.9	3475.4	279.28	**1515.8**	279.31
50	200	1174	17.8	0.4	3.0	0.3	**953.3**	629.06	1257.9	629.06
	300	1252	25.8	0.7	4.5	0.3	**3691.3**	433.21	8038.2	433.21
60	240	1017	16.1	1.3	7.5	0.7	**2424.5**	828.61	4323.1	828.61
	360	2026	67.2	4.7	26.8	2.1	>10,000	(390.75;610.11)	>10,000	**(390.61;440.86)**
70	280	1559	41.3	7.4	37.6	3.9	>10,000	(692.32;1139.23)	>10,000	**(577.53;877.89)**
	420	3968	473.7	15.9	81.5	7.7	>10,000	**(334.11;886.44)**	>10,000	(0;708.16)
	Average	1603.0	82.4	4.1	22.6	2.0	>5160.4	Gap: 17%	>5718.5	Gap: 18%

Table 1 Computational Results with Scenario based Uncertainty.

In this case the subproblem (22)–(23) is solved by evaluating each of the scenarios and choosing the scenario i that minimizes $(w^i)^\top \overline{y}$. For each instance in the computational experiments, we generate vectors $w^1, \ldots, w^{1000}$ each chosen uniformly at random from the ball around w_0 with radius 1. For comparison with CPLEX, we reformulate problem [RCNP] by replacing $\min\{(w^1)^\top y, (w^2)^\top y, \ldots, (w^n)^\top y\}$ with a new variable z and adding the constraint $z \leq (w^i)^\top y$ for every scenario i thus turning the problem into a MIP.

The 32 network instances are solved and the results are reported in Table 1. The proposed Benders approach outperforms CPLEX for 22 of the 32 tested instances. We notice that the Forest-Fire instances are the easiest to solve requiring an average of 22.2 seconds of computational time for the proposed Benders approach and 901.8 seconds for CPLEX. For the remaining networks, the time limit is reached for 8 instances with the Benders approach and for 5 instances with CPLEX. We also notice that CPLEX tends to outperform the proposed approach on the instances that have a larger number of edges as is the case for the Erdös-Rényi graphs where the proposed approach outperformed CPLEX for the instances with lower number of edges while CPLEX achieved better results for the cases with higher number of edges.

4.3.2. Gamma Uncertainty In this section, we consider the gamma robustness as defined by Bertsimas and Sim (2004a) and considered by Fan and Pardalos (2010b) in the context of the critical node selection problem. In this case, at most $\lfloor \Gamma \rfloor$ entries of w are allowed to deviate from the nominal scenario w_0, in addition to one entry (i,j) that is allowed to deviate by $(\Gamma - \lfloor \Gamma \rfloor)\hat{w}_{ij}$ for a given $\Gamma \geq 0$. The subproblem (22)–(23) can then be rewritten as

$$\begin{aligned} \overline{y}^\top w_0 - \max_{\tilde{w}} \ & \tilde{y}^\top v \\ \text{s.t. } & 0 \leq \tilde{w} \leq 1 \\ & e^\top v \leq \Gamma \end{aligned}$$

where $\tilde{y}_{ij} = \overline{y}_{ij}\hat{w}_{ij}$. An optimal solution $\overline{w}$ is easily obtained by setting $w_{ij} = (w_0)_{ij} - \hat{w}_{ij}$ for the $\lfloor \Gamma \rfloor$ largest entries $\tilde{y}_{ij}$ and setting $w_{ij} = (w_0)_{ij} - (\Gamma - \lfloor \Gamma \rfloor)\hat{w}_{ij}$ for the next largest entry $\tilde{y}_{ij}$. For comparison, we use the approach of Bertsimas and Sim (2004a) to reformulate problem [RCNP] with gamma uncertainty as a MIP that we solve using CPLEX.

For our computational testing, we assume $\Gamma = \frac{|V|}{2}$ and the results are reported in Table 2. The proposed Benders approach outperformed CPLEX for 17 out of the 32 tested instances. Similar to the scenario based uncertainty case, the proposed Benders approach significantly outperformed CPLEX for the Forest-Fire networks while mixed results were obtained for the remaining networks. The Watts-Strogatz instances are the most challenging to solve, with CPLEX achieving better results for 7 of the 8 tested instances. Furthermore, CPLEX outperforms the proposed approach for the Erdös-Rényi instances that have a larger number of edges while the proposed approach achieves better results for the instances that have a lower number of edges.

To further evaluate the proposed Benders approach, we compare the computational results with the approach of Fan and Pardalos (2010b). Fan and Pardalos (2010b) presented an approach for the critical node selection problem with gamma robustness based on solving

Instance		Benders							CPLEX	
Vertices	Edges	CUTS0	CPU0	CPU_{FW}	CPU_{WCW}	CPU_{BC}	CPU	Opt	CPU	Opt
Forest-Fire										
40	186	235	0.01	0.0	0.0	0.0	**1.9**	612.35	7.2	612.35
	194	271	0.01	0.0	0.0	0.0	**1.7**	607.36	20.8	607.36
50	276	486	0.09	0.0	0.1	0.0	**8.0**	748.53	197.2	748.53
	312	426	0.05	0.0	0.0	0.0	**9.6**	686.53	291.0	686.53
60	226	147	0.10	0.0	0.1	0.0	**2.0**	1515.4	34.9	1515.40
	326	571	0.17	0.1	0.2	0.1	**15.5**	1084.5	1299.6	1084.50
70	292	378	0.14	0.1	0.1	0.0	**9.3**	1922.44	148.6	1922.44
	436	990	0.65	0.3	0.6	0.2	**79.3**	1027.71	2175.3	1027.71
	Average	438.0	0.2	0.1	0.2	0.1	15.9	Gap: 0%	521.8	Gap: 0%
Barabási-Albert										
40	76	197	0.2	0.0	0.0	0.0	**3.2**	402.52	117.6	402.52
	256	865	3.0	0.1	0.0	0.0	69.1	232.70	**38.3**	232.70
50	96	347	0.4	0.1	0.0	0.1	**4.9**	624.50	149.0	624.50
	336	1379	11.3	6.2	2.1	3.1	>10,000	(212.58;285.58)	>10,000	**(212.71;213.26)**
60	116	856	2.6	1.4	0.4	0.8	**140.3**	563.96	1,393.0	563.96
	416	1325	15.6	1.5	0.4	0.7	4,591.5	310.44	**832.9**	310.44
70	136	484	1.2	0.5	0.1	0.3	**12.8**	1043.68	1,632.6	1043.68
	496	3570	180.0	1.8	9.3	0.6	>10,000	(366.39;473.46)	>10,000	**(366.41;382.09)**
	Average	1127.9	26.8	1.5	1.5	0.7	>3102.7	Gap: 6%	>3020.4	Gap: 1%
Erdös-Rényi										
40	174	570	1.7	0.7	0.3	0.4	**178.4**	325.82	260.2	325.82
	208	793	4.4	1.1	0.5	0.7	1299.6	264.96	**278.0**	264.96
50	186	1036	7.4	0.9	0.3	0.5	**358.2**	603.54	785.9	603.54
	370	1954	45.9	5.1	1.8	2.7	>10,000	(298.09;330.57)	**1563.2**	298.09
60	178	817	5.5	1.4	0.5	0.9	**504.1**	944.60	2164.3	944.60
	408	2446	99.7	2.3	0.7	1.2	>10,000	(363.31;548.61)	**3908.1**	363.31
70	190	1029	10.3	2.1	0.7	1.2	**494.4**	1556.62	5641.4	1556.62
	648	4298	509.0	0.4	0.1	0.2	>10,000	(366.32;609.91)	>10,000	**(365.96;451.31)**
	Average	1617.9	85.5	1.8	0.6	1.0	>4104.3	Gap: 10%	>3075.1	Gap: 2%
Watts-Strogatz										
40	160	790	4.2	0.6	0.3	0.4	653.3	344.88	**326.1**	344.88
	240	1078	9.6	4.0	1.7	2.6	6763.5	261.62	**754.6**	261.62
50	200	1190	14.8	0.4	0.1	0.3	1265.4	606.53	**571.6**	606.53
	300	1256	21.2	0.9	0.3	0.6	4551.3	410.85	**4436.6**	410.85
60	240	1080	12.7	1.1	0.3	0.7	**2220.3**	801.61	2843.2	801.61
	360	2281	71.7	5.5	1.6	3.1	>10,000	(363.28;566.05)	**8775.2**	363.30
70	280	1575	30.1	5.6	1.4	3.6	>10,000	(668.20;1103.12)	>10,000	**(604.93;828.57)**
	420	3731	405.7	13.0	3.3	7.9	>10,000	(365.39;870.18)	>10,000	**(302.34;627.08)**
	Average	1622.6	71.3	3.9	1.1	2.4	>5681.7	Gap: 17%	>4713.4	Gap: 10%

Table 2 Computational Results with Gamma Uncertainty.

a series of nominal critical node selection problems where the one that yields the best objective function also yields an optimal solution for the original problem under gamma uncertainty. Although solving each of the nominal problems is typically computationally less challenging than solving the problem with gamma uncertainty, the burden of solving a large number of nominal problems is computationally expensive. As shown in Table 3, the approach that is discussed in Fan and Pardalos (2010b) requires significantly more computational time than the proposed Benders approach. Only 4 of the tested instances were solved to optimality while the remaining 28 instances were stopped at the time limit thus yielding only an upper bound on the optimal solution.

Instance				Instance			
Vertices	Edges	CPU	Opt	Vertices	Edges	CPU	Opt
Forest-Fire				Barabási-Albert			
40	186	593.5	609.37	40	76	4941.5	402.52
	194	1829.3	604.56		256	>10,000	487.83†
50	276	>10,000	890.80†	50	96	>10,000	856.14†
	312	>10,000	805.10†		336	>10,000	356.86†
60	226	3111.6	1508.28	60	116	>10,000	890.56†
	326	>10,000	1617.40†		416	>10,000	494.25†
70	292	>10,000	2006.22†	70	136	>10,000	1611.81†
	436	>10,000	1558.95†		496	>10,000	690.38†
Erdös-Rényi				Watts-Strogatz			
40	174	>10,000	407.80†	40	160	>10,000	416.95†
	208	>10,000	347.44†		240	>10,000	361.90†
50	186	>10,000	870.45†	50	200	>10,000	854.21†
	370	>10,000	458.43†		300	>10,000	641.51†
60	178	>10,000	1426.45†	60	240	>10,000	1232.92†
	408	>10,000	588.45†		360	>10,000	596.11†
70	190	>10,000	2379.63†	70	280	>10,000	1269.59†
	648	>10,000	736.63†		420	>10,000	984.93†

†: Best found upper bound.

Table 3 Computational Results with Gamma Uncertainty Using Fan and Pardalos (2010b) Approach.

4.3.3. Ellipsoidal Uncertainty In this section, we consider ellipsoidal robustness (Ben-Tal et al. 2009) where the uncertainty set is

$$\mathcal{U} = \{w\colon (w - w_0)W^{-1}(w - w_0) \leq 1\}$$

for a positive definite matrix W chosen such that $w \geq 0$ for all $w \in \mathcal{U}$. In this case the subproblem (22)–(23) is minimizing a linear objective function over an ellipsoid and hence admits an optimal solution

$$\overline{w} = w_0 - \frac{1}{\sqrt{\overline{y}^\top W \overline{y}}} W \overline{y}.$$

For each instance in the computational experiments, we choose the matrix W as $CC^\top$ for a random matrix C. The largest eigenvalue of W is then normalized to 1.

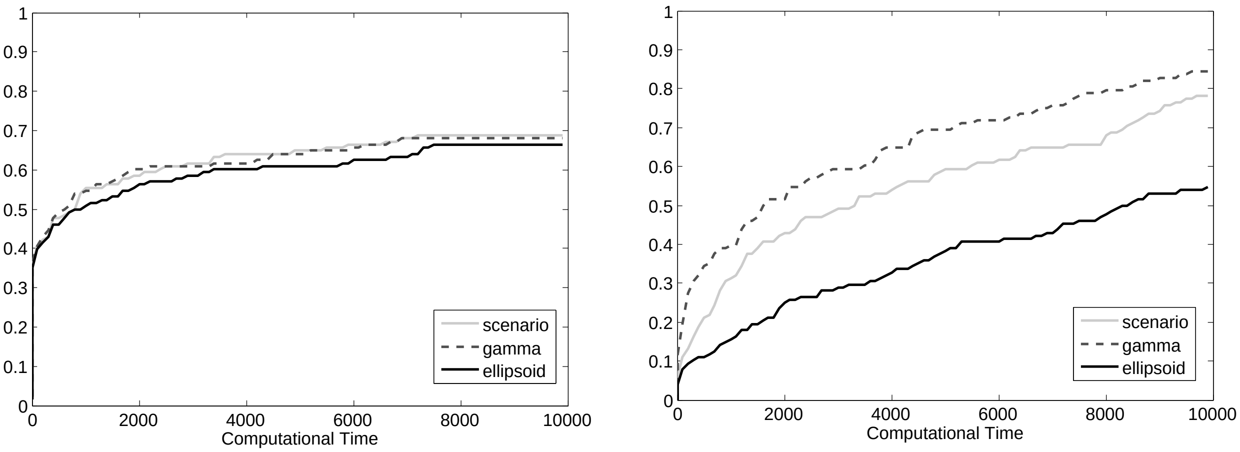
Since $\overline{w}^\top y = w_0^\top y - \sqrt{y^\top W y}$, we can model [RCNP] with ellipsoidal uncertainty as a second-order cone program that we solve using CPLEX. The results that are displayed in Table 4 show that the proposed Benders approach is particularly successful for the ellipsoidal uncertainty cases. The proposed Benders approach outperformed CPLEX in 29 of the 32 tested instances. Futhermore, CPLEX failed to solve 13 instances withing the time limit while the proposed Benders approach reached the time limit in only 7 instances. We note that in 12 of the 13 instances that CPLEX failed to solve, CPLEX could not even find a lower bound for the problem within the time limit.

Instance		Benders							CPLEX	
Vertices	Edges	CUTS0	CPU0	CPU_{FW}	CPU_{WCW}	CPU_{BC}	CPU	Opt	CPU	Opt
Forest-Fire										
40	186	224	0.6	0.0	0.1	0.0	**2.0**	609.37	32.0	609.37
	194	276	0.9	0.0	0.1	0.0	**2.5**	604.56	238.8	604.56
50	276	480	3.2	0.1	0.6	0.0	**10.3**	750.91	6751.9	750.91
	312	426	3.0	0.0	0.3	0.0	**10.9**	690.70	5017.6	690.70
60	226	161	1.6	0.1	0.7	0.0	**3.8**	1508.28	174.5	1508.28
	326	557	6.9	0.1	0.6	0.0	**22.2**	1087.52	3959.3	1087.52
70	292	397	7.3	0.1	1.1	0.0	**16.1**	1917.18	1047.1	1917.18
	436	925	21.3	0.3	3.2	0.1	**107.9**	1040.39	>10,000	(0;1115.93)
	Average	430.8	5.6	0.1	0.8	0.0	22.0	Gap: 0%	>3402.7	Gap: 13%
Barabási-Albert										
40	76	214	0.3	0.0	0.0	0.0	**4.0**	406.88	1,964.6	406.88
	256	648	2.2	0.1	0.5	0.0	**29.4**	242.99	344.7	242.99
50	96	344	1.2	0.1	0.7	0.0	**6.4**	630.46	4,547.4	630.46
	336	1172	12.3	6.1	40.3	2.2	>10,000	(228.32;309.26)	>10,000	**(228.32;231.21)**
60	116	793	6.1	0.1	1.0	0.0	**157.7**	578.91	>10,000	(0;881.99)
	416	976	15.5	1.2	9.9	0.5	**1,549.0**	330.94	5,370.2	330.94
70	136	465	5.2	0.4	3.7	0.2	**18.6**	1056.06	>10,000	(0;1317.51)
	496	2900	119.5	2.0	18.7	0.7	>10,000	**(390.89;480.99)**	>10,000	(0;509.86)
	Average	939.0	20.3	1.3	9.4	0.5	>2720.6	Gap: 6%	>6528.4	Gap: 38%
Erdös-Rényi										
40	174	586	2.6	0.7	3.7	0.3	**174.0**	332.65	2324.4	332.65
	208	690	4.1	1.1	6.1	0.6	**791.7**	273.84	1675.1	273.84
50	186	1005	10.4	0.8	5.6	0.4	**397.0**	610.07	4698.3	610.07
	370	1841	40.5	0.6	4.2	0.3	7125.3	312.80	**4899.8**	312.80
60	178	815	11.4	1.4	10.8	0.7	**718.3**	950.85	>10,000	(0;1091.54)
	408	2393	100.1	1.7	13.5	0.7	>10,000	**(382.69;536.76)**	>10,000	(0;525.04)
70	190	1010	23.2	1.7	15.5	0.7	**356.0**	1558.63	>10,000	(0;1668.70)
	648	3062	326.8	0.6	5.3	0.2	>10,000	**(390.89;589.03)**	>10,000	(0;729.84)
	Average	1425.3	64.9	1.1	8.1	0.5	>3695.3	Gap: 8%	>6699.7	Gap: 50%
Watts-Strogatz										
40	160	734	4.4	0.5	3.0	0.3	**664.4**	350.95	2182.4	350.95
	240	1031	9.7	2.4	13.6	1.2	4341.6	270.94	**2790.8**	270.94
50	200	1190	16.7	0.4	3.0	0.3	**1033.7**	612.99	4980.9	612.99
	300	1249	23.7	0.6	3.7	0.3	**3485.1**	422.58	7982.8	422.58
60	240	1078	19.9	1.0	8.2	0.6	**2056.9**	811.25	>10,000	(0;901.63)
	360	2107	71.8	5.2	40.9	2.3	>10,000	**(382.69;583.04)**	>10,000	(0;554.54)
70	280	1523	48.9	7.7	71.0	4.0	>10,000	**(687.72;1121.77)**	>10,000	(0;926.69)
	420	3773	447.9	16.0	145.4	7.6	>10,000	**(328.20;873.98)**	>10,000	(0;709.62)
	Average	1585.6	80.4	4.2	36.1	2.1	>5197.7	Gap: 17%	>7242.1	Gap: 50%

Table 4 Computational Results with Ellipsoidal Uncertainty.

4.3.4. Varying Budget In the computational results that are presented earlier, the parameter K in constraint (2) is set to 5. In this section, we consider $K = 3, 7, 10$ and evaluate the impact on the computational performance. The results are displayed in Tables 1–3 in the online supplement.

The results show that the proposed approach performs significantly better than CPLEX for $K = 3$ where the proposed Benders approach achieved better results for all types of graphs and all uncertainty sets. Particularly, for ellipsoidal uncertainty, CPLEX failed in solving 17 of the 32 tested instances within the time limit while the proposed approach failed in solving only 1. As K is increased to 7, CPLEX starts to achieve better results. In total, CPLEX achieved better results for 31 of the 96 tested instances while the proposed approach performed better on the remaining 65 instances. With $K = 10$, CPLEX achieved



(a) Benders Decomposition

(b) CPLEX

Figure 1 Performance profiles for the 4 different uncertainty types with respect to the running times for the proposed Benders decomposition algorithm (a) and for CPLEX (b).

better results particularly for the Erdös-Rényi and Watts-Strogatz graphs while the proposed approach achieved better results for the Forest-Fire graphs. With scenario based uncertainty, the proposed Benders approach achieved better results for 12 instances while CPLEX achieved better results in the remaining 20 instances. For gamma uncertainty, the proposed approach achieved better results in 10 of the 32 tested instances while for the ellipsoidal uncertainty the proposed approach achieved better results in 21 of the 32 tested instances. We note that for several instances, CPLEX fails to even find a feasible solution within the time limit particularly for the ellipsoidal uncertainty where CPLEX failed in finding a feasible solution for 27 instances.

4.4. Results Summary

This section provides a summary of the effect of uncertainty on the performance of the proposed Benders decomposition algorithm and on the performance of CPLEX. The performance profiles (Dolan and Moré 2002) which are shown in Figure 1 were constructed to show for each uncertainty set, the cumulative ratio of instances that were solved in less than the computational time that is given by the x-axis of the plots. These include all the instances with $K = 3, 5, 7, 10$. Figure 1 (a) shows that the type of the uncertainty set has little effect on the performance of the proposed Benders decomposition algorithm. All the tested uncertainty sets have similar performance profiles, thus revealing the advantage of the proposed Benders decomposition approach which handles the uncertainty sets through an efficient optimization oracle. The performance of CPLEX is however highly dependent on the type of the uncertainty set. As shown in Figure 1 (b), CPLEX achieves the best performance when gamma uncertainty is used outperforming the cases with scenario and ellipsoidal uncertainty, with the latter being the most computationally challenging.

5. Conclusion

This paper provides a Benders decomposition algorithm for the robust critical node selection problem for a very general class of uncertainty sets. The novel Benders decomposition approach exploits the structure of the Benders subproblem while explicitly dealing with the uncertainty. A polynomial time algorithm based on the Floyd-Warshall approach is devised to solve the Benders subproblem. Extensive computational testing is conducted and a comparison with CPLEX is presented using three classes of uncertainty sets: scenario based uncertainty, gamma uncertainty, and ellipsoidal uncertainty. For all classes of uncertainty sets, the proposed Benders approach achieved good performance.

References

Addis, Bernardetta, Marco Di Summa, Andrea Grosso. 2013. Identifying critical nodes in undirected graphs: Complexity results and polynomial algorithms for the case of bounded treewidth. *Discrete Applied Mathematics* **161** 2349 – 2360.

Arulselvan, Ashwin, Clayton W. Commander, Panos M. Pardalos, Oleg Shylo. 2007. Managing network risk via critical node identification. Nalân Gülpınar, Berç Rustem, eds., *Risk management in telecommunication networks*. Springer, 474–477.

Arulselvan, Ashwin, Clayton W. Commander, Oleg Shylo, Panos M. Pardalos. 2011. Cardinality-constrained critical node detection problem. Nalân Gülpınar, Peter Harrison, Berç Rustem, eds., *Performance Models and Risk Management in Communications Systems*, *Springer Optimization and Its Applications*, vol. 46. Springer New York, 79–91.

Ben-Tal, Aharon, Laurent El Ghaoui, Arkadi Nemirovski. 2009. *Robust Optimization*. Princeton Series in Applied Mathematics.

Benders, Jacques F. 1962. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik* **4** 238–252.

Bertsimas, Dimitris, Melvyn Sim. 2004a. The price of robustness. *Operations Research* **52** 35–53.

Bertsimas, Dimitris, Melvyn Sim. 2004b. Robust discrete optimization under ellipsoidal uncertainty sets Http://web.mit.edu/dbertsim/www/papers.html.

Cohen, Reuven, Shlomo Havlin, Daniel Ben-Avraham. 2003. Efficient immunization strategies for computer networks and populations. *Physical review letters* **91** 247901–247905.

Commander, Clayton W., Panos M. Pardalos, Valeriy Ryabchenko, Stan Uryasev, Grigoriy Zrazhevsky. 2007. The wireless network jamming problem. *Journal of Combinatorial Optimization* **14** 481–498.

Di Summa, Marco, Andrea Grosso, Marco Locatelli. 2011. Complexity of the critical node problem over trees. *Computers & Operations Research* **38** 1766 – 1774.

Di Summa, Marco, Andrea Grosso, Marco Locatelli. 2012. Branch and cut algorithms for detecting critical nodes in undirected graphs. *Computational Optimization and Applications* **53** 649–680.

Dolan, Elizabeth D., Jorge J. Moré. 2002. Benchmarking optimization software with performance profiles. *Mathematical Programming* **91** 201–213.

Fan, Neng, Panos M. Pardalos. 2010a. Linear and quadratic programming approaches for the general graph partitioning problem. *Journal of Global Optimization* **48** 57–71.

Fan, Neng, Panos M. Pardalos. 2010b. Robust optimization of graph partitioning and critical node detection in analyzing networks. Weili Wu, Ovidiu Daescu, eds., *Combinatorial Optimization and Applications*, *Lecture Notes in Computer Science*, vol. 6508. Springer Berlin Heidelberg, 170–183.

Fan, Neng, Qipeng P. Zheng, Panos M. Pardalos. 2012. Robust optimization of graph partitioning involving interval uncertainty. *Theoretical Computer Science* **447** 53–61.

Floyd, Robert. 1962. Algorithm 97: Shortest Path. *Communications of the ACM* **5** 345.

Garg, Naveen, Vijay V. Vazirani, Mihalis Yannakakis. 1996. Approximate max-flow min-(multi) cut theorems and their applications. *SIAM Journal on Computing* **25** 235–251.

Geoffrion, Arthur M. 1972. Generalized Benders decomposition. *Journal of optimization theory and applications* **10** 237–260.

Ghaddar, Bissan, Miguel Anjos, Frauke Liers. 2011. A branch-and-cut algorithm based on semidefinite programming for the minimum k-partition problem. *Annals of Operations Research* **188** 155–174.

Hopcroft, John, Robert Tarjan. 1973. Efficient algorithms for graph manipulation. *Communications of the ACM* **16** 372–378.

Li, Mian, Steven A Gabriel, Yohan Shim, Shapour Azarm. 2011. Interval uncertainty-based robust optimization for convex and non-convex quadratic programs with applications in network infrastructure planning. *Networks and Spatial Economics* **11** 159–191.

Magnanti, Thomas L., Richard T. Wong. 1981. Accelerating Benders decomposition: Algorithmic enhancement and model selection criteria. *Operations Research* **29** 464–484.

Montemanni, Roberto. 2006. A Benders decomposition approach for the robust spanning tree problem with interval data. *European Journal of Operational Research* **174** 1479–1490.

Naoum-Sawaya, Joe, Samir Elhedhli. 2013. An interior-point Benders based branch-and-cut algorithm for mixed integer programs. *Annals of Operations Research* **210** 33–55.

Saito, Hiroo, Kazuo Murota. 2007. Benders decomposition approach to robust mixed integer programming. *Pacific Journal of Optimization* **3** 99–112.

Shen, Siqian, J. Cole Smith. 2012. Polynomial-time algorithms for solving a class of critical node problems on trees and series-parallel graphs. *Networks* **60** 103–119.

Shen, Siqian, J. Cole Smith, Roshan Goli. 2012. Exact interdiction models and algorithms for disconnecting networks via node deletions. *Discrete Optimization* **9** 172 – 188.

Siddiqui, Sauleh, Shapour Azarm, Steven Gabriel. 2011. A modified Benders decomposition method for efficient robust optimization under interval uncertainty. *Structural and Multidisciplinary Optimization* **44** 259–275.

Siddiqui, Sauleh, Steven A. Gabriel, Shapour Azarm. 2014. Solving mixed-integer robust optimization problems with interval uncertainty using Benders decomposition. *Journal of the Operational Research Society* .

Tao, Zhou, Fu Zhongqian, Wang Binghong. 2006. Epidemic dynamics on complex networks. *Progress in Natural Science* **16** 452–457.

Ventresca, Mario. 2012. Global search algorithms using a combinatorial unranking-based problem representation for the critical node detection problem. *Computers & Operations Research* **39** 2763 – 2775.

Ventresca, Mario, Dionne Aleman. 2014. A derandomized approximation algorithm for the critical node detection problem. *Computers & Operations Research* **43** 261–270.