Exact Solutions for the Robust Prize-Collecting Steiner Tree Problem

Eduardo Álvarez-Miranda
DEIS
Università di Bologna, Italy
e.alvarez@unibo.it

Ivana Ljubić
Department of Statistics and Operations Research
University of Vienna, Austria
ivana.ljubic@univie.ac.at

Paolo Toth
DEIS
Università di Bologna, Italy
paolo.toth@unibo.it

Abstract

We consider a combinatorial optimization problem that models an expansion of fiber optic telecommunication networks. Thereby, we are given a set of customers with potential gains of revenue and the set of edges with fixed installation costs. The goal is to decide which customers to connect to a given root node so that the sum of edge costs plus the node revenues for the nodes that are left out from the solution is minimized. The problem is known in the literature as the *Prize-Collecting Steiner Tree Problem* (PCStT). In many applications it is unrealistic to assume that the gains of revenue or installation costs are known in advance. In this paper, we extend this well-studied deterministic problem by considering the robust optimization approach in which the input parameters are subject to interval uncertainty. To control the level of conservatism of the solution, we consider Bertsimas & Sim robust optimization approach.

We propose a branch-and-cut approach to solve this problem to optimality and provide an extensive computational study on a set of benchmark instances that are adapted from those previously used to solve the deterministic version of the problem. We show how the price of robustness influences the costs of the solutions and the algorithm performance.

I. Introduction

Supose that a telecommunication company is in the process of defining an expansion plan of a fiber optic network in a given area where a set of potential customers is located. The company is interested in knowing, given a planning horizon, which subset of customers should be served in the network taking into account two elements: the prizes of the customers (that correspond to potential revenues) and the infrastructure costs needed to connect them. This problem can be formulated as a network optimization problem called the *Prize-Collecting Steiner Tree Problem* (PCStT) and it is the problem that we will focus on in this paper.

When facing these types of strategic decisions, companies should consider the presence of uncertainty in problem parameters as an inevitable feature of the decision-making process. In this particular case, customer revenues and connection costs are uncertain parameters since they are affected by many external economic or even social factors. Consequently, uncertainty in both edge costs and node prizes should be part of any decision model in order to obtain reliable or robust solutions from the economic point of view. In our models, robustness can be seen as a guarantee of protection against data uncertainty.

II. The Prize Collecting Steiner Tree Problem

The term *Prize Collecting* was used for the first time by Balas, see [2], in the context of the traveling salesman problem. However, it was in [7] where the PCStT has been introduced. A formal definition of the problem can be given as follows.

Given is an undirected graph $G = (V, E)$ ($|V| = n$, $|E| = m$) with edge costs $c_e \in \mathbb{R}^{>0}$ for all $e \in E$, and with node prizes $p_v \in \mathbb{R}^{\geq 0}$ for all $v \in V$. The PCStT consists of finding a tree $T = (V(T), E(T))$ of G, that minimizes the function

$$f(T) = \sum_{e \in E(T)} c_e + \sum_{v \in V \setminus V(T)} p_v. \tag{1}$$

For a feasible solution T, the function (1) corresponds to the sum of the costs c_e of the edges in the tree, $e \in E(T)$, plus the sum of the prizes p_v of the nodes that are *not spanned* by the tree, $v \in V \setminus V(T)$; this definition of the PCStT is known as the

Goemans and Williamson PCStT (GW-PCStT) [7]. Recalling the application mentioned above, the graph $G = (V, E)$ would correspond to the potential network where we want to find our expansion plan, so edges $e \in E$ are the possible connection links with the corresponding construction costs c_e and nodes $v \in V$ represent customers or street intersections with the corresponding potential revenues $p_v > 0$ or $p_v = 0$, respectively. By $V_{p_i>0}$ ($|V_{p_i>0}| = n'$) we will denote the set of *potential customers* and by $V_{p_i=0}$, the set of Steiner nodes.

The PCStT can be also defined as the problem of finding a tree T that minimizes

$$f_{NW}(T) = \sum_{e \in E(T)} c_e - \sum_{v \in V(T)} p_v. \tag{2}$$

Function 2 corresponds to the minimization version of the Net-Worth PCStT (NW-PCStT) which was introduced in [12]. Although functions (1) and (2) are equivalent in the sense that both produce the same optimal solutions, they are not equivalent regarding approximation algorithms, see [12].

Approximation algorithms for the PCStT are discussed in [7], [11] and [12]. Heuristic procedures are implemented in [8], [13] and [16]. The first published work on polyhedral studies for the PCStT is [15], where a cutting plane algorithm is proposed. The cuts are efficiently generated when a violation of a generalized subtour elimination constraint (GSEC) is verified. In [14], a branch-and-cut algorithm based on a directed cut-set MIP formulation has been designed and implemented. Several state-of-the-art methods are combined and pre-processing techniques are used. The proposed procedure has significantly improved the algorithm presented in [15]. The same set of benchmark instances has been solved by two orders of magnitude faster. Optimal solutions have also been achieved for large-scale real-world instances that have been used in the design of optic fiber networks. This work will play an important role in our algorithmic approach.

A. A Mixed Integer Programming Formulation for PCStT

To characterize the set of feasible solutions for the PCStT, i.e., subtrees of G, we consider a directed graph model and use connectivity inequalities to guarantee connectivity of the solution, see [14].

We transform the graph $G = (V, E)$ into the directed graph $G_{SA} = (V_{SA}, A_{SA})$. The vertex set $V_{SA} = V \cup \{r\}$ contains the nodes of the input graph G *and an artificial root* vertex r. The arc set A_{SA} is defined as $A_{SA} = \{ri \mid i \in V_{p_i \leq 0}\} \cup A$, where $A = \{ij, ji \mid e = \{i, j\} \in E\}$. A subgraph T_{SA} of G_{SA} that forms a directed tree rooted at r is called a *Steiner arborescence* and is a feasible solution of the problem. We will use the following notation: A set of nodes $R \subset V_{SA}$ and its complement $\bar{R} = V_{SA} \backslash R$ induce two directed cuts: $\delta^+(R) = \left\{ji \mid i \in R, j \in \bar{R}\right\}$ and $\delta^-(R) = \left\{ji \mid i \in \bar{R}, j \in R\right\}$. Let z_{ij}, $\forall ij \in A$, be a binary variable such that $z_{ij} = 1$ if arc ij belongs to a feasible solution T and $z_{ij} = 0$ otherwise, let y_i, $\forall i \in V$ a binary variable such that $y_j = 1$ if node i belongs to a feasible solution T and $y_j = 0$ otherwise, and let $T \equiv (\mathbf{x}, \mathbf{y})$. The set of constraints that characterizes the set of feasible solutions of PCStT is given as follows:

$$\sum_{ji \in A_{SA}} z_{ji} = y_i \qquad \forall i \in V_{SA} \backslash \{r\} \tag{3}$$

$$\sum_{ij \in \delta^-(R)} z_{ij} \geq y_k, \quad k \in R, r \notin R, \qquad \forall R \subset V_{SA} \tag{4}$$

$$\sum_{ri \in A_{SA}} z_{ri} = 1 \tag{5}$$

Let x_e, $\forall e \in E$, be a binary variable such that $x_e = 1$ if edge e belongs to a feasible solution T and $x_e = 0$ otherwise. The connection between $\mathbf{x}$ and $\mathbf{z}$ variables is given by

$$x_e = z_{ij} + z_{ji} \quad \forall e = \{i, j\} \in E \tag{6}$$

The corresponding polytope $\mathscr{T}$ is given as the convex hull of the set of all feasible solutions satisfying these inequalities, i.e.,

$$\mathscr{T} = \{(\mathbf{x}, \mathbf{y}) \in \{0, 1\}^{|E|+|V|} \mid (\mathbf{x}, \mathbf{y}, \mathbf{z}) \text{ satisfies } (3) - (6)\}.$$

The constraints (4), also known as *cut* or *connectivity inequalities* are the directed counterpart of undirected GSECs used in [15]. They ensure that there is a directed path between each customer i such that $y_i = 1$ and the root r. In-degree constraints (3) guarantee that the in-degree of each node of the tree is equal to one. The root-out-degree constraint (5) makes sure that the artificial root is connected to exactly one of the terminals. In addition, the following inequalities are used to improve the quality of lower bounds of the LP-relaxation of the previous model:

$$z_{rj} \leq 1 - y_i, \quad \forall i < j, \quad i \in V_{p_i>0} \tag{7}$$

$$\sum_{ji \in A_{SA}} z_{ji} \leq \sum_{ij \in A_{SA}} z_{ij}, \quad \forall i \in V_{p_i=0}, \quad i \neq r. \tag{8}$$

Constraints (7) are the so-called *asymmetry constraints*, they ensure that for each feasible solution the costumer vertex adjacent to the root is the one with the smallest index. Constraints (7) are the *flow-balance constraints*, originally introduced for the Steiner tree problem – they guarantee that a Steiner node cannot be a leaf in an optimal PCStT solution.

III. Robust Optimization Approaches

In the PCStT a crucial information corresponds to the economic parameters of the problem: edge costs and node prizes. In all the references mentioned before a fundamental assumption is complete knowledge about the value of these parameters, so they are, essentially, deterministic models. However, in most of real-world applications this assumption may not hold and uncertainty should be included as a critical feature of the model.

An alternative to model and include uncertainty in the decision process is Stochastic Optimization (see [17]). This approach assumes that the distribution of the uncertain parameters is known and it typically generates a solution that optimizes a measure of expected performance. However, it is known that - and it has been demonstrated very clearly in [3] - the solutions of stochastic optimization problems can present a remarkable sensitivity to perturbation in problem parameters, turning the solution into a sub-optimal, or even an unfeasible one.

In this paper we consider decision-making environments with lack of complete knowledge about the uncertain state of data and instead of dealing with probabilistic uncertainty (as in Stochastic Optimization) we actually deal with *deterministic uncertainty* [5]. In contrast to probabilistic models that treat the input parameters as random variables, in the deterministic uncertainty models we assume that the input parameters belong to a known, deterministic set. This is in the core of many real world applications and it is the motivation supporting Robust Optimization (RO), where the essential objective is to find solutions that will have a *reasonably good* performance (of optimality and/or feasibility) for all possible realizations of parameter values.

In the last 20 years several RO models have been proposed, responding to different motivations and conceptual definitions; for a deep and extensive study on RO we refer the reader to [4]. In our opinion there are three main characteristics that define the differences of RO models: **(1)** The nature of input data; whether the data belong to e.g., an ellipsoidal set or polyhedral set, a closed interval, or a set of discrete scenarios; **(2)** If robustness is considered with respect to the value of the objective function (*robust solution*), to the feasibility of the solution (*robust model*) or both; **(3)** The definition of *reasonably good performance* of a solution, which is what determines the main features of the model.

In this paper we consider the RO concept by Bertsimas and Sim (B&S) defined in [5, 6]. This model is considered as one of the most important references in the field of RO. Regarding the first characteristic mentioned above, this approach tackles interval uncertainty. Regarding robustness, the B&S model allows to find solutions that are robust in terms of optimality and/or feasibility of the solution. The definition of what is a *reasonably good performance* of a solution is given by the protection against a pre-defined number of parameters that might be subject to uncertainty.

IV. Formulations for Robust PCStT (RPCStT)

In this paper, we consider interval uncertainty, which means that associated with each input parameter there is a closed interval with its lower and upper bounds. Formally, in the case of the PCStT, an interval $[c_e^-, c_e^+]$, such that $0 < c_e^- \leq c_e^+$, is associated to each edge $e \in E$, and an interval $[p_v^-, p_v^+]$, such that $0 \leq p_v^- \leq p_v^+$ is associated to each customer $v \in V_{p_i>0}$. To simplify the notation, we will define $0 \leq p_v^- \leq p_v^+$ for all nodes $v \in V$, where $p_v^- = p_v^+ = 0$ for Steiner nodes $v \in V_{p_i=0}$. Since we consider deterministic uncertainty, each input parameter can take any value from the corresponding interval without any specific (or known) behavior and independently of the values taken by the other parameters. The lower interval values c_e^- and p_v^- will be referred to as *nominal values*, i.e., they are the values to be considered if the deterministic PCStT is solved. *Deviations* from the nominal values are defined as: $d_e = c_e^+ - c_e^-$, for all $e \in E$ and $d_v = p_v^+ - p_v^-$, for all $v \in V$. In the reminder of this section we will present a mathematical programming formulation for robust counterpart of the PCStT considering the B&S model.

The PCStT under interval uncertainty has been considered before in [1]. The authors used an alternative RO model based on a Risk/Cost trade-off concept defined in [9] and provided polynomial time algorithms for solving both PCStT and RPCStT on 2-trees. In this context, our work is complementary since we consider a different RO model and we provide a more general algorithmic framework focusing on graphs with general structure.

A. The B&S RPCStT

Suppose that a decision maker wants to solve the PCStT problem in which the input parameters, edge costs and node prizes, are subject to interval uncertainty. In many practical applications it is unlikely that all of edge costs and/or node prizes will present an uncertain behavior at the same time. Therefore, we assume that only a subset of input data is subject to uncertainty, while the remaining parameters are fixed to their *nominal* values. More precisely, the decision maker may assume that only Γ_E edges and Γ_V nodes ($\Gamma_E \in [0, m]$ and $\Gamma_V \in [0, n']$) will be subject to uncertainty, although she/he does not know exactly which they are. Without loss of generality, we will assume that the values of Γ_E and Γ_V are integral.

The essence of the model is to find a solution that is "robust" considering all scenarios in which Γ_E edges and Γ_V nodes present an adverse behavior. If $\Gamma_E = 0$ and $\Gamma_V = 0$, then uncertainty is ignored and the problem to solve is nothing but the nominal problem, whereas if $\Gamma_E = m$ and $\Gamma_V = n'$, i.e., full uncertainty is assumed, the most conservative robust solution is sought. Considering the general RO mathematical programming formulation for Combinatorial Optimization problems with interval uncertainty presented in [5], the B&S RPCStT can be formulated as:

$$ROPT(\Gamma_E,\Gamma_V) = \min_{T\in\mathscr{T}} \left\{ \sum_{e\in E_T} c_e^- + \beta_E^*(\Gamma_E) + \sum_{v\in V\setminus V_T} p_v^- + \beta_V^*(\Gamma_V) \right\}, \tag{9}$$

where $\beta_E^*(\Gamma_E) = \max\left\{ \sum_{e\in\tilde{E}\cap E_T} d_e \,\middle|\, \tilde{E}\subseteq E, |\tilde{E}|\le \Gamma_E \right\}$ and $\beta_V^*(\Gamma_V) = \max\left\{ \sum_{v\in\tilde{V}\cap\{V\setminus V_T\}} d_v \,\middle|\, \tilde{V}\subseteq V, |\tilde{V}|\le\Gamma_V \right\}$.

These last two functions are the so-called protection functions and they provide robustness to the solutions in terms of protection of optimality in presence of a given level of data uncertainty, represented by Γ_E and Γ_V. It is easy to notice that an optimal solution for (9) can be interpreted as the one that minimizes the total nominal cost plus the cost of maximal Γ_E deviations in the cost of the edges of the solution plus the maximal Γ_V deviations in the prizes of the nodes that are *not* spanned by the solution. If $\Gamma_E = m$ and $\Gamma_V = n'$, the solution will obviously correspond to the optimal deterministic solution in which all edge costs and node prizes will be set to their upper bounds. The flexibility provided by Γ_E and Γ_V is the main advantage of the model from the practical point of view, because it allows the decision maker to include her/his preferences in order to control the level of conservatism of the solutions. Setting $\Gamma_E = m$ and $\Gamma_V = n'$ leads to the most conservative solutions, $\Gamma_E = 0$ and $\Gamma_V = 0$ to a very optimistic one.

Since there are two types of binary variables that define the tree T, the objective function in (9) contains two non-linear nested maximization problems. To overcome this, it is necessary to use strong duality; let $T^*\equiv(\mathbf{x}^*,\mathbf{y}^*)$ be an optimal tree for (9). Consider the next subproblem:

$$\beta_E^*(\Gamma_E) = \max\left\{ \sum_{e\in E} d_e u_e x_e^* \mid \sum_{e\in E} u_e \le \Gamma_E,\, 0\le u_e\le 1,\, \forall e\in E \right\}. \tag{10}$$

By strong duality, the former linear programming problem can also be written as:

$$\beta_E^*(\Gamma_E) = \min\left\{ \theta\Gamma_E + \sum_{e\in E} h_e \mid h_e + \theta \ge d_e x_e^*,\ h_e\ge 0\ \forall e\in E,\ \theta\ge 0 \right\}. \tag{11}$$

Similarly, the second nested problem can be also be written as the following linear problem:

$$\beta_V^*(\Gamma_V) = min\left\{ \lambda\Gamma_E + \sum_{v\in V} k_v \mid k_v + \lambda \ge d_v\left(1-y_v^*\right),\ k_v\ge 0\ \forall v\in V,\ \lambda\ge 0 \right\}. \tag{12}$$

Combining (9), (11) and (12), we can formulate the B&S RPCStT as the following mixed integer programming (MIP) model:

$$ROPT(\Gamma_E,\Gamma_V) = \min \sum_{e\in E} c_e^- x_e + \theta\Gamma_E + \sum_{e\in E} h_e + \sum_{v\in V} p_v^-\left(1-y_v\right) + \lambda\Gamma_V + \sum_{v\in V} k_v \tag{13}$$

$$\text{s.t. } h_e + \theta \ge d_e x_e,\ \forall e\in E \tag{14}$$

$$k_v + \lambda \ge d_v\left(1-y_v\right),\ \forall v\in V \tag{15}$$

$$(\mathbf{h},\mathbf{k}) \ge \mathbf{0}, \theta, \lambda \ge 0 \tag{16}$$

$$(\mathbf{x},\mathbf{y}) \in \mathscr{T} \tag{17}$$

By following the same ideas presented before, the B&S Robust counterpart of the NW-PCStT is defined as

$$ROPT_{NW}(\Gamma_E,\Gamma_V) = \min_{(\mathbf{x},\mathbf{y})\in\mathscr{T}} \left\{ \sum_{e\in E} c_e^- x_e + \beta_E^*(\Gamma_E) - \left(\sum_{v\in V} p_v^+ y_v - \eta_V^*(\Gamma_V) \right) \right\} \tag{18}$$

where $\eta_V^*(\Gamma_V)$ =max $\left\{ \sum_{v\in V} d_v u_v y_v \mid \sum_{v\in V} u_v \le \Gamma_V,\ u_v\in[0,1]\forall v\in V \right\}$. It is important to note that in the case of the B&S Robust NW-PCStT, the nominal value of node prizes is the upper limit of the corresponding interval, $p_v^+\ \forall v\in V$, and not the lower limit as it is in the case of the robust PCStT. It can be easily seen from (18) that larger values of Γ_V will increase the total value of the solution as it is expected in this RO model. An MILP formulation can be easily obtained by following the same procedure explained for the PCStT.

It is known that for the deterministic case the connection between $f(T)$ and $f_{NW}(T)$ is given as $f_{NW}(T) = f(T) - \sum_{v\in V} p_v$. The next result provides further information about connections between these two versions of the problem for their robust counterparts.

Observation 1: For a fixed value of $\tilde{\Gamma}_E\in[0,m]$, and $\Gamma_V\in\{0,n'\}$, the robust counterparts of the PCStT and the NW-PCStT are equivalent, i.e., they produce identical optimal subtrees. The following connection exists between the corresponding objective values:

$$ROPT_{NW}(\tilde{\Gamma}_E,0) = ROPT(\tilde{\Gamma}_E,n') - \sum_{v\in V} p_v^+$$

and

$$ROPT_{NW}(\tilde{\Gamma}_E, n') = ROPT(\tilde{\Gamma}_E, 0) - \sum_{v \in V} p_v^- .$$

Intuitively it seems that the robust counterparts of these problems are equivalent when taking $\Gamma_V \in [0, n']$ for the PCStT and $\Gamma'_V = n' - \Gamma_V$ for the NW-PCStT. However, although this is true for the case described in the Observation 1, this does not hold for $\Gamma_V = \{1, \ldots, n'-1\}$. For a fixed value of $\tilde{\Gamma}_E \in [0, m]$, let us take, for example, $\Gamma_V = 1$ which leads to

$$ROPT(\Gamma_E, 1) = \min_{(\mathbf{x},\mathbf{y}) \in \mathscr{T}} \left\{ \sum_{e \in E} c_e^- x_e + \beta^*(\Gamma_E) + \sum_{v \in V} p_v^- (1 - y_v) + \max_{v \in V} \{ d_v (1 - y_v) \} \right\} \tag{19}$$

$$ROPT_{NW}(\Gamma_E, n' - 1) = \min_{(\mathbf{x},\mathbf{y}) \in \mathscr{T}} \left\{ \sum_{e \in E} c_e^- x_e + \beta^*(\Gamma_E) - \sum_{v \in V} p_v^- y_v - \min_{v \in V} \{ d_v y_v \} \right\} \tag{20}$$

for the NW-PCStT. It is clear that (19) and (20) are not equivalent.

Although in this paper we present a MIP-based exact approach for solving the robust counterpart of the PCStT, it is also possible to solve the RPCStT by successively solving $(n'+1)(m+1)$ classical instances of the problem. The latter result follows from the result given in [5] and its usefulness strongly depends on how efficient one can solve the deterministic version of the PCStT.

The following lemma *improves* the previous result:

Lemma 2: Given $\Gamma_E \in [0, m]$ and a given $\Gamma_V \in [0, n']$, the B&S Robust Counterpart of the PCStT can be solved by solving $(m - \Gamma_E + 1)(n' - \Gamma_V + 1)$ nominal problems

$$ROPT(\Gamma_E, \Gamma_V) = \min_{\substack{a \in \{\Gamma_E, \ldots, m+1\} \\ b \in \{\Gamma_V, \ldots, n'+1\}}} G^{a,b},$$

where for $a \in \{\Gamma_E, \ldots, m+1\}$ and $b \in \{\Gamma_V, ..., n'+1\}$:

$$G^{a,b} = \Gamma_E d_a + \Gamma_V d_b + \min_{(\mathbf{x},\mathbf{y}) \in \mathscr{T}} \left(\sum_{e \in E} c_e^- x_e + \sum_{e=1}^{a} (d_e - d_a) x_e + \sum_{v \in V} p_v^- (1 - y_v) + \sum_{v=1}^{b} (d_v - d_b)(1 - y_v) \right).$$

and nodes and edges are sorted in increasing order with respect to their deviations so that $d_{i+1} \leq d_i$.

The result can be easily extended to the NW-RPCStT.

V. A Branch-and-Cut Algorithm for the RPCStT

The MIP formulation (13) - (17) can not be solved directly, even for small instances, since there is an exponential number of constraints of type (4); consequently, more sophisticated and specific techniques should be designed and implemented to solve this model.

To solve the B&S RPCStT to optimality a Branch-and-cut algorithm is implemented using CPLEX 12.2 and Concert Technology. The algorithm is an adaptation of the exact approach presented in [14] and its main feature is the separation procedure applied to constraints (4). The algorithm can be used to solve both, NW-RPCStT and GW-RPCStT. However, due to the space limitation, we will present computational results for the RPCStT only.

a) Initialization: The integer linear program initially contains all variables and the constraints (3), (5)-(8). In addition, we explicitly insert the subtour elimination constraints of size 2:

$$z_{ij} + z_{ji} \leq y_i, \quad \forall i \in V_{SA} \setminus \{r\}, \quad (i,j) \in A_{SA}$$

to avoid too frequent calls of the maximum flow procedure.

b) Separation: At each node of the branch-and-bound tree, these constraints are separated as long as they are violated by the current LP-solution. For a given LP-solution $(\hat{\mathbf{z}}, \hat{\mathbf{y}})$, we construct a support graph $\hat{G}_{SA}$ with arc capacities set to $\hat{z}_{ij}$, for all $ij \in A$. Then we calculate the maximum flow from the root node r to each potential customer node $v \in V_{p_i > 0}$ such that $\hat{y}_v > 0$. If this maximum flow value is less than y_v, we have found a violated inequality (4), induced by the corresponding min-cut in the graph $\hat{G}_{SA}$, and we insert it into the LP. For the calculation of the maximum flow we used an adaptation of the maximum flow algorithm given in [10]. In addition, this separation randomly selects terminals and detects the violated cut-sets using *nested*, *back-flow* and *minimum cardinality* cuts (see [14] for more details). We restrict the number of inserted cuts within each separation callback to 100.

c) Branching: Branching on single arc variables produces a huge imbalance in the branch-and-bound tree. Whereas discarding an edge from the solution (i.e., setting x_{ij} to zero) doesn't bring much, setting the customer variable y_v to one significantly reduces the size of the search subspace. Therefore we set the highest branching priorities to variables y_v, $v \in V_{p_i > 0}$.

d) Further Settings: We used default CPLEX settings, with the only exception of the following parameters that have been tuned during our preliminary experiments: the usage of primal heuristics and CPLEX cuts was turned off; the emphasis was set to *optimality*.

VI. Computational Results and Conclusions

In our computational experiments four sets of benchmark instances have been tested: P, K, C and D. These instances have been used in most of the papers discussing algorithm design for the PCStT ([15],[14],[16]). Instances of the group P were introduced in [12] – they are unstructured and designed to have constant expected degree and profit to weight ratio. Group K are randomly generated geometric graphs designed to have a structure similar to street maps [12]. Groups C and D were presented in [8]. They are derived from the instances of the Steiner tree problem in graphs provided in the OR-Library[1].

Given an original instance Prob for the deterministic PCStT, the corresponding robust instance, named Prob-α-β, ($\alpha \in [0,1]$ and $\beta \in [0,1]$) is derived as follows: the number of nodes and edges are left unchanged. Lower limits for intervals defining edge costs and node prizes are set to the corresponding deterministic values c_e and p_v, i.e., $c_e^- = c_e$ $\forall e \in E$ and $p_v^- = p_v$ $\forall v \in V$. The upper limit of edge costs, c_e^+ is a value drawn uniformly randomly from $[c_e, (1+\alpha)\, c_e]$ $\forall e \in E$. Similarly, the upper limit of node prizes, p_v^+, is taken uniformly randomly from $[p_v, (1+\beta)\, p_v]$ $\forall v \in V$. Parameters α and β allow to control the width of the corresponding intervals and, consequently, the level of uncertainty of the problems.The algorithm was run in a AMD Athlon(TM) 64 X2 Dual Core 3800+ 2.0 GHz machine with 1.00 GB in RAM.

Figure 1 illustrates the *price of robustness* [5], i.e., the changes of $ROPT$ with respect to Γ_E and Γ_V for instance K400.10-0.5-0.5. As expected, the objective value $ROPT$ is a concave function on Γ_E and Γ_V, which increases when increasing the values of Γ_E and Γ_V. For the instance K400.10-0.5-0.5, Γ_V seems to have more influence than Γ_E on the behavior of $ROPT$ up to values of $\Gamma_E = 4$, which can be explained by the fact that the construction of a solution depends more on node prizes than on edge costs. From the $ROPT$ axis in Figure 1, we can observe that the relative increase of $ROPT$ does not exceed 10% even for high values of Γ_V. This increase is smaller than what we would have expected since both edge costs and node prizes were allowed to increase up to 50% of their nominal values ($\alpha = 0.5$ and $\beta = 0.5$). This is an example of the *robustness* of the solutions and the protection provided by the model even under the high data uncertainty. Another observation is that, although for instance K400.10-0.5-0.5 we have $n' = 50$, only values of Γ_V in the interval $[0, 20]$ seem to produce different solutions, which means that $\Gamma_V = 20$ is enough to provide robustness to the solutions. This is explained by the size of the obtained solutions; in the case of this particular instance, no more than 20 customers nodes are not connected for the obtained solutions and that is why only values of Γ_V up to 20 might produce changes in the objective function. A similar behavior was observed for the remaining instances as well.

We wanted to test the influence of parameters Γ_E and Γ_V to the performance of the algorithm and to the increase of the cost function. Therefore, we performed a study in which we considered settings with $\Gamma_E \in \{0, 5, 10, 25\}$ and $\Gamma_V \in \{0, 5, 10, 25\}$. Tables I - IV summarize our computational results obtained on the set of instances derived from the groups c1-10, d1-d10, K and P, respectively, with $\alpha = \beta = 0.1$. Each entry in these tables represents an average value over a subset of corresponding instances. For each group we show: the running time in seconds (t(s)), the number of inserted cuts of type (4) (#cuts), the number of branch-and-bound nodes (#BBn), the number of edges ($E(T)$) and the number of terminals ($R(T)$) in the optimal solution, and the relative increase of $ROPT$ with respect to the optimal value obtained for the nominal cost ($\Delta ROPT$).

One can see how the value of $ROPT$ increases with increasing values of Γ_E and Γ_V. However, since these instances were created considering $\alpha = 0.1$ and $\beta = 0.1$ the relative increase is not as perceptible as in the case of K400.10-0.5-0.5. We did not provide results for more values of Γ_E and Γ_V since no interesting differences have been observed.

Information about the structure of the solutions can be read from columns $E(T)$ and $R(T)$. Although the value of the objective function increases, the structure of the solutions remains almost the same. For instances C and K, basically the same number of customers remains connected despite the changes in Γ_E or Γ_V. In the case of D and P, there are minor variations in the number of customers connected, but these changes are not relevant. This means that the solutions provided by the robust model are able to *resist* data perturbations since they are likely to maintain their optimality even when allowing the higher level of uncertainty.

Figure 2 illustrates the increase of the running time for instance K400.10-0.5-0.5 and smaller values of Γ_E. For values of $\Gamma_V > 20$ and $\Gamma_E > 50$, our algorithm has not been able to find an optimal solution within a given time limit of 10000 seconds.

Comparing the running times provided in Figure 2 and in columns $t(s)$ of Tables I - IV, one observes that the performance of our exact approach it heavily affected by the values of Γ_E or Γ_V. It can be seen that, despite the good performance of the algorithm for the nominal problem, the running times can reach extremely high values for some values of Γ_E or Γ_V, particularly for values of Γ_V between 5 and 20. However, when considering values of Γ_V greater than 20 and values of Γ_E smaller than 5 it is possible to obtain more reasonable running times.

It is important to compare the two charts given in Figures 1 and 2. One can see that for the values of Γ_V between 0 and 20 the changes in the objective function are more dramatic, which means that the optimal solutions are also different. Therefore,

[1] OR-library: J. E. Beasley, http://mscmga.ms.ic.ac.uk/info.html

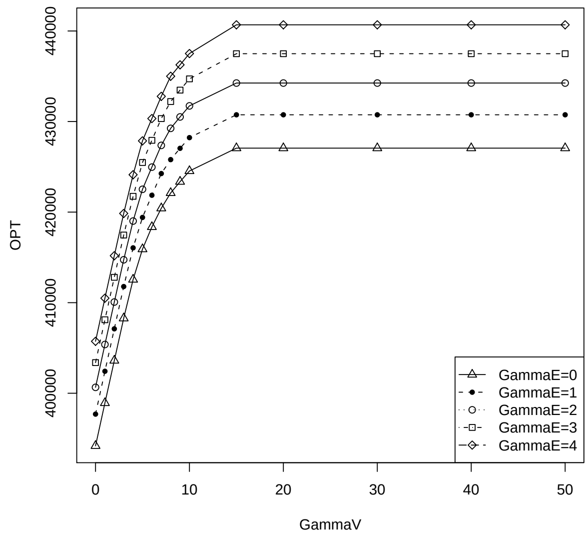

Fig. 1. The value of the optimal solution depending on Γ_E and Γ_V parameters for the instance `K400.10-0.5-0.5`.

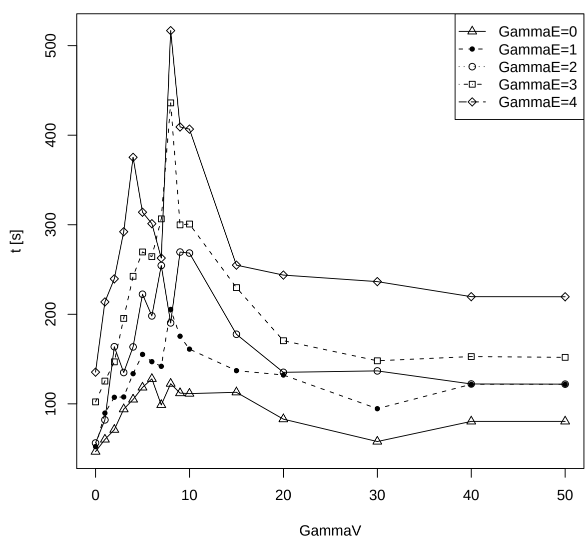

Fig. 2. The running times depending on Γ_E and Γ_V parameters for the instance `K400.10-0.5-0.5`.

some values of Γ_E and Γ_V can drastically affect the set of feasible solutions increasing the running times but also allowing to find different solutions.

VII. Conclusions and Future Work

In this paper we studied the PCStT problem with interval uncertainty associated to its parameters. To include and handle this uncertainty we considered the B&S Robust Optimization approach, formulating the robust counterpart of the problem as a mixed integer problem. A specific branch-and-cut algorithm was implemented to solve the problem. The algorithm was tested on a set of problems generated from state-of-the-art instances of the deterministic version of the problem. The computational results suggest: (1) the RO model allows to provide robustness to the solutions since even if uncertainty is considered in most of problem parameters the obtained solutions do not change considerably, and (2) the algorithmic performance is strongly influenced by the two model parameters, Γ_E and Γ_V. A possible direction for future work could be the development of some algorithmic strategy to reduce this dependance in order to improve the algorithmic performance and be able to solve more challenging instances for any value of Γ_E and Γ_V in a reasonable time.

Γ_V	Γ_E	t(s)	#cuts	#BBn	$E(T)$	$R(T)$	$\%\Delta ROPT$
0	0	0.61	208	0	100	66	0
0	5	0.86	234	0	99	66	0.87
0	10	0.90	242	0	99	66	1.29
0	25	1.06	305	0	99	65	1.85
5	0	0.665	228	0	100	66	1.19
5	5	0.89	243	0	99.	66	2.06
5	10	0.97	270	0	99	66	2.49
5	25	1.04	309	0	98	65	3.00
10	0	0.64	238	0	100	66	1.40
10	5	0.87	226	0	100	66	2.27
10	10	1.00	281	0	99	66	2.70
10	25	1.10	336	0	99	65	3.21
25	0	0.65	243	0	100	66	1.65
25	5	0.87	227	0	100	66	2.53
25	10	1.00	275	0	100	66	2.96
25	25	1.04	306	0	99	66	3.47

TABLE I
AVERAGE RESULTS OBTAINED FOR THE GROUP C-0.1-0.1 OF INSTANCES.

Γ_V	Γ_E	t(s)	#cuts	#BBn	$E(T)$	$R(T)$	$\%\Delta ROPT$
0	0	4.69	726	0	198	128	0
0	5	9.35	765	0	198	129	0.68
0	10	9.59	836	0	198	129	1.03
0	25	8.32	823	0	196	128	1.51
5	0	4.96	682	0	199	130	0.89
5	5	9.50	764	0	198	129	1.58
5	10	9.08	773	0	198	129	1.94
5	25	10.09	911	1	196	128	2.43
10	0	4.84	750	0	199	130	1.08
10	5	8.78	734	0	198	129	1.76
10	10	9.84	830	0	198	130	2.13
10	25	9.10	756	1	197	128	2.61
25	0	4.84	713	0	199	130	1.26
25	5	8.95	696	0	199	130	1.95
25	10	8.70	650	1	199	130	2.31
25	25	9.41	814	0	197	129	2.79

TABLE II
AVERAGE RESULTS OBTAINED FOR THE GROUP D-0.1-0.1 OF INSTANCES.

Γ_V	Γ_E	t(s)	#cuts	#BBn	$E(T)$	$R(T)$	$\%\Delta ROPT$
0	0	12.47	1178	0	12	9	0
0	5	17.57	1319	3	12	9	0.52
0	10	19.19	1335	6	12	9	0.65
0	25	22.24	1398	6	12	9	0.78
5	0	14.76	1355	0	12	9	2.49
5	5	17.69	1370	3	12	9	3.02
5	10	19.65	1373	6	12	9	3.19
5	25	22.51	1453	7	12	9	3.35
10	0	15.94	1445	0	12	9	3.33
10	5	21.20	1527	4	13	9	3.86
10	10	22.06	1458	6	12	9	4.03
10	25	24.01	1480	4	13	9	4.19
25	0	18.37	1567	0	12	9	4.16
25	5	22.85	1634	3	13	9	4.69
25	10	26.46	1675	7	13	9	4.86
25	25	31.04	1637	7	13	9	5.01

TABLE III
AVERAGE RESULTS OBTAINED FOR THE GROUP K-0.1-0.1 OF INSTANCES.

Γ_V	Γ_E	t(s)	#cuts	#BBn	$E(T)$	$R(T)$	$\%\Delta ROPT$
0	0	0.42	92	0	44	27	0.00
0	5	0.57	57	1	44	27	0.63
0	10	0.76	42	2	44	27	1.03
0	25	2.12	65	12	43	27	1.89
5	0	0.50	57	1	44	27	0.39
5	5	0.66	45	2	44	27	1.06
5	10	1.01	66	4	44	27	1.44
5	25	3.66	66	22	44	27	2.32
10	0	0.48	76	1	44	28	0.49
10	5	0.62	39	2	44	27	1.16
10	10	0.96	40	4	44	27	1.55
10	25	3.11	66	19	44	27	2.42
25	0	0.40	71	0	45	28	0.53
25	5	0.58	37	1	44	28	1.21
25	10	0.79	53	3	45	28	1.60
25	25	2.35	66	13	45	27	2.48

TABLE IV
AVERAGE RESULTS OBTAINED FOR THE GROUP `P-0.1-0.1` OF INSTANCES.

REFERENCES

[1] E. Álvarez Miranda, A. Candia, X. Chen, B. Li, and X. Hu. Efficient algorithms for the PCStT with interval data. *AAIM 2010*, LNCS 6124:13–24, 2010.

[2] E. Balas. The prize-collecting traveling salesman problem. *Networks*, 19:621–636, 1989.

[3] A. Ben-Tal and A. Nemirovski. Robust solutions of linear programming problems contaminated with uncertain data. *Math. Prog.*, 88:411–421, 2000.

[4] A. Ben-Tal, L. El-Ghaoui, and A. Nemirovski, editors. *Robust Optimization*. Princeton Series in Applied Mathematics, 1st edition, 2010.

[5] D. Bertsimas and M. Sim. Robust discrete optimization and network flows. *Math. Prog.*, B(98):49–71, 2003.

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

[7] D. Bienstock, M. X. Goemans, D Simchi-Levi, and D. Williamson. A note on the prize collecting traveling salesman problem. *Math. Prog.*, 59:413–420, 1993.

[8] S. A. Canuto, M. G. C. Resende, and C. C. Ribeiro. Local search with perturbations for the prize-collecting Steiner tree problem in graphs. *Networks*, 38:50–58, 2001.

[9] X. Chen, J. Hu, and X. Hu. A new model for path planning with interval data. *Computers & OR*, 36(6):1893–1899, 2009.

[10] B.V. Cherkassky and A.V. Goldberg. On implementing a push-relabel method for the maximum flow problem. *Algorithmica*, 19:390–410, 1997.

[11] M. X. Goemans and D. P. Williamson. The primal-dual method for approximation algorithms and its application to network design problems. *Approximation algorithms for NP-hard problems (Hochbaum, D. S. (Ed.), World Scientific Publishing Company)*, 1996.

[12] D. S. Johnson, M. Minkoff, and S. Phillips. The prize-collecting Steiner tree problem: Theory and practice. *SODA 2000*, 2000.

[13] G.W. Klau, I. Ljubić, A. Moser, P. Mutzel, P. Neuner, U. Pferschy, and R. Weiskircher. Combining a memetic algorithm with integer programming to solve the prize-collecting Steiner tree problems. *GECCO 2004*, LNCS 3102:1304–1315, 2004.

[14] I. Ljubić, R. Weiskircher, U. Pferschy, G. Klau, P. Mutzel, and M. Fischetti. An algorithmic framework for the exact solution of the prize-collecting Steiner tree problem. *Math. Prog.*, Series B 105:427–449, 2006.

[15] A. Lucena and M. G. Resende. Strong lower bounds for the prize-collecting Steiner problem in graphs. *Discrete Applied Mathematics*, 141:277–294, 2004.

[16] A. Salles da Cunha, A. Lucena, N. Maculan, and M. G. C. Resende. A relax-and-cut algorithm for the prize-collecting Steiner problem in graphs. *Discrete Applied Mathematics*, 157:1198–1217, 2009.

[17] S.P. Uryasev and P.M. Pardalos, editors. *Stochastic Optimization: Algorithms and Applications*. Springer (Applied Optimization Volume 54), 1st edition, 2001.