Acta Mathematicae Applicatae Sinica, English Series
Vol. 30, No. 1 (2014) 1–26
DOI: 10.1007/s10255-014-0269-z
http://www.ApplMath.com.cn & www.SpringerLink.com

Acta Mathematicae Applicatae Sinica, English Series

Risk Models for the Prize Collecting Steiner Tree Problems with Interval Data

Eduardo Álvarez-Miranda[1,2], Alfredo Candia-Véjar[1], Xu-jin CHEN[3], Xiao-dong HU[3], Bi LI[3]

[1]Industrial Management Department, Universidad de Talca, Chile (E-mail: acandia@utalca.cl)
[2]DEI, University of Bologna, Italy (Email: ealvarez@utalca.cl)
[3]Academy of Mathematics and Systems Science, CAS, Beijing 100190, China
(E-mail: {xchen, xdhu, libi}@amss.ac.cn)

Abstract Given a connected graph $G = (V, E)$ with a nonnegative cost on each edge in E, a nonnegative prize at each vertex in V, and a target set $V' \subseteq V$, the Prize Collecting Steiner Tree (PCST) problem is to find a tree T in G interconnecting all vertices of V' such that the total cost on edges in T minus the total prize at vertices in T is minimized. The PCST problem appears frequently in practice of operations research. While the problem is NP-hard in general, it is polynomial-time solvable when graphs G are restricted to series-parallel graphs.

In this paper, we study the PCST problem with interval costs and prizes, where edge e could be included in T by paying cost $x_e \in [c_e^-, c_e^+]$ while taking risk $(c_e^+ - x_e)/(c_e^+ - c_e^-)$ of malfunction at e, and vertex v could be asked for giving a prize $y_v \in [p_v^-, p_v^+]$ for its inclusion in T while taking risk $(y_v - p_v^-)/(p_v^+ - p_v^-)$ of refusal by v. We establish two risk models for the PCST problem with interval data. Under given budget upper bound on constructing tree T, one model aims at minimizing the maximum risk over edges and vertices in T and the other aims at minimizing the sum of risks over edges and vertices in T. We propose strongly polynomial-time algorithms solving these problems on series-parallel graphs to optimality. Our study shows that the risk models proposed have advantages over the existing robust optimization model, which often yields NP-hard problems even if the original optimization problems are polynomial-time solvable.

Keywords uncertainty modeling; prize collecting Steiner tree; interval data; series-parallel graphs; polynomial-time solvability

2000 MR Subject Classification 68Q25; 68R99; 68W05

1 Introduction

The Prize Collecting Steiner Tree (PCST) problem introduced by [9] has been extensively studied in the areas of computer science and operations research due to its wide range of real-world applications[6,18,22]. A typical application of PCST occurs when a natural gas provider wants to build a most profitable transportation system for natural gas delivery from a station to some customers on scattered locations, where each link (segment of pipeline) is associated with a cost which is incurred if the link is installed, and each location is associated with a profit which is obtained if the location is connected to the station by links installed. Moreover, the transportation system is required to contain some specified customers. One of the most

Manuscript received November 3, 2011. Revised February 6, 2012.
Supported in part by the National Natural Science Foundation of China under Grant No. 11021161 and 10928102, 973 Program of China under Grant No. 2011CB80800, Chinese Academy of Sciences under Grant No. kjcx-yw-s7, project grant of "Center for Research and Applications in Plasma Physics and Pulsed Power Technology, PBCT-Chile-ACT 26" and Dirección de Programas de Investigación, Universidad de Talca, Chile.

important special cases of the PCST problem is the Steiner Minimum Tree (SMT) problem[17] which arises repeatedly in diverse application areas, where all profits associated with locations are zero.

As a natural generalization of the SMT problem, the PCST problem has even wider real-world applications. Over the last decade, a lot of real problems for which the PCST is crucial in the resolution process have been studied. These problems arise from very different industrial and scientific contexts, showing the potential and versatility of the PCST model. Among many other recent operational research applications, [22] carried out a concrete application of the PCST problem to the design of fiber optic networks for some German cities, where both the trade-off between connection costs (represented by edge costs) and customers revenues (represented by vertex prizes) and the goal of establishing the most profitable network were perfectly modeled by the PCST problem. Using the PCST model, [27] designed a leakage detection system for finding the optimal location of detectors and their corresponding transponders in the water distribution network of the Swiss city, Lausanne, such as to provide a desired coverage under budget constraints. An application in a very different bioinformatic context was presented by [13], where the PCST problem together with the algorithmic framework by [22] was applied to find, for the first time, exact solutions for the problem of finding functional modules in protein-to-protein interaction networks. In the same scientific field, [5] and [6] used the PCST model to solve "inference problem" for "inferring" transduction networks in cell communication.

Since the SMT problem is NP-complete in general[9], so is the PCST problem. The latter admits a 2-approximation polynomial-time algorithm[14] for general networks. On the other hand, practical OR applications often aim at optimal solutions and thus take concrete network topology into account. As transportation and communication networks usually possess certain sparse and planar structural properties[25], their representations as series-parallel graphs are often convenient[28], offering clearer representations of real-world instances. The algorithmic design for series-parallel networks often serve as subroutines of more general procedures which employ separator techniques to decompose more general topologies into series-parallel pieces[15]. When restricted to series parallel graphs, [30] proved that the SMT problem is polynomial-time solvable. In this paper, we will extend their approach to an efficient algorithm for the PCST problem on series-parallel graphs. The reader is referred to [20,29,31,32] for more research and applications on series-parallel networks. In particular, the densest series-parallel graphs, known as 2-trees and independently reliable networks[30], play an important role in the reliable broadcasting problem on independently reliable networks in which all pairs of nodes can communicate as long as the failures of nodes and edges are isolated[7].

In contrast to the above single-parameter settings as in the traditional PCST problem, it is often necessary to take interval data into account in the real applications such as scheduling[10], path planning[4] and minimum-cut search[1], collectively known as interval combinatorial optimization problems[21], where intervals are used to indicate possible ranges of values variables can take. For example, in the above PCST applications, the gas (communication service) provider may spend c_e^+ dollars to install a gas pipe (optical cable) link e using the best materials to assure continuous transmission through e over a long period of time without any interruption for maintenance. On the other hand, the provider can also spend $c_e^-(< c_e^+)$ dollars to install the link e using ordinary materials while taking the risk of malfunction at e and service suspension for repair. Generally, lower expense on link construction could lead to higher risk of transmission malfunction. Similarly, if the gas (service) provider wants to collect the highest possible prize p_v^+ from the customer at location v for the gas (communication data) transmitted to v, then the provider faces the highest risk of rejection by customer at v because of the existence of competitors who sell the same kind of products (services). Usually, the smaller prize is demanded, the smaller the risk of being rejected is taken. For easy description, in the remainder of the paper we use the *risks of edges* (*vertices*) to denote the risk of malfunction at links (rejection by customers). Naturally, it is desirable to make good balances between low net expenses and small risks. Under budget constraint, the gas (service) provider may have full control over his/her edge payment and can ask for any reasonable prize at any reachable vertex. Our work contributes not only to modeling this kind of practical trade-off between expenses and risks but also to minimizing risks in polynomial time under budget restriction for series-parallel networks.

Recently, [12] and [16] proposed two novel models for network optimization with interval data on network links. The latter model is to find a solution that minimizes the maximum risk on network links, and the former model is to find a solution with minimum total link risk. The solutions of these models are paths or trees together with the cost x_e spent within the given interval $[c_e^-, c_e^+]$ for every edge e on the paths or trees. The risk of edge e is quantified as $(c_e^+ - x_e)/(c_e^+ - c_e^-)$, which is consistent with the common sense that higher expense often brings about more satisfactory service (prevention of malfunction). The network connection problems under these models are polynomial-time solvable, preserving the polynomial-time solvability of the original optimization problems with single-parameter data, i.e., the shortest path problem and minimum spanning tree problem. In this paper, we will extend their approaches to the PCST problem in series-parallel graphs by considering not only interval costs on edges but also interval prizes at vertices. Our PCST models consider not only the risks of edges, but also the risks of vertices, which are not studied in the previous models. As we know, graphically, vertices behave quite differently from edges, and turn out less amenable in combinatorial optimization. (A typical example consists of the minimum vertex cover problem, which is NP-hard, and its edge-counterpart – the minimum edge cover problem, which is polynomial time solvable.) Algorithmic approaches successful in dealing with edges often fail for vertices. Our success in coping with risks of vertices relies on exploiting the structural properties of series-parallel graphs. Moreover, when restricted to series-parallel graphs, the risk sum model by [12] and minmax risk model by [16] are special cases of our PCST models.

Along a different line, what has been widely studied is the problem under nondeterministic setting, where interval data models uncertainty in the way that one cannot determine which values in the intervals will realize. A lot of literature has been developed under the name of *robust optimization*, in which one optimizes against the worst instance that might arise with realized values in the given intervals[26]. One of the most popular objectives in robust optimization is to find a solution that minimizes the maximum regret against the worst realization of values in the given intervals, where the "worst" is with respect to the *regret* on a value realization which is the difference between the actual value of the solution (selected for all value realizations) and the value of the optimal solution under the value realization. Despite the popularity, many robust optimization problems, such as the robust shortest path[33] and robust spanning tree problems[3], suffer from two major drawbacks: they are NP-hard even though their deterministic counterparts are polynomial-time solvable; their solutions tend to be over-conservative, as the worst-case scenario is always anticipated.

To address the tractability and over-conservatism of the robust solution, [8] proposed an approach to controlling the degree of conservation of the solution by regulating the number of the values which are allowed to vary in the given intervals. They established a bounded probability of their robust solutions being infeasible, but they did not make the theoretical clarification regarding the suboptimality. Their robust solutions might be far from efficient.

The remainder of the paper is organized as follows: In Section 2, we present a linear-time algorithm for the PCST problem on series parallel graphs. In Section 3, we establish the min-max risk model and min-sum risk model for the PCST problem with interval data, and propose two polynomial-time algorithms for the PCST problem on series parallel graphs under these two models. In Section 4, we compare proposed models and algorithms through extensive simulations. Finally, we conclude the paper in Section 5. The preliminary version of the paper is an extended abstract[2], which contains only theoretical results without giving any proofs and simulation study.

2 Efficient Algorithm for PCST Problem on Series-Parallel Graphs

In the PCST problem, we are given a connected graph $G = (V, E)$ with vertex-set V of size n and edge-set E of size m, where each vertex $v \in V$ is assigned a nonnegative prize $p_v \in \mathbb{R}_+$, and each edge $e \in E$ is assigned a nonnegative cost $c_e \in \mathbb{R}_+$. For any subgraph S of G, its vertex-set and edge-set are written as $V(S)$ and $E(S)$, respectively. We abbreviate $\sum\limits_{e \in E(S)} c_e$ to $c(E(S))$ and $\sum\limits_{v \in V(S)} p_v$ to $p(V(S))$. The *value* of subgraph S is defined as

$$\nu(S, c, p) \equiv c(E(S)) - p(V(S)).$$

In our definition of the PCST problem, the input includes a *target set* $V' \subseteq V$, also called a *terminal set*. The objective of the PCST problem is to find a tree in G such that it contains V' and its value is minimum among all trees in G spanning V':

$$\text{(PCST)} \qquad \min \big\{ \nu(T, c, p) \mid T \text{ is a tree in } G \text{ and } V' \subseteq V(T) \big\}.$$

Such a tree is called an *optimal* PCST in $(G, V'; c, p)$ or simply in G, and denoted by $T_{\text{opt}}(G, V'; c, p)$. (PCST problem has another equal version that minimizes the edge-costs for establishing the network plus the penalties of the vertices outside of the solution.)

As the following lemma shows, one can give the target set V' in the problem input in an implicit way by reassigning each vertex in V' a sufficiently large prize.

Lemma 2.1. *Given $G = (V, E)$ with target set $V' \subseteq V$, $c \in \mathbb{R}_+^E$, $p \in \mathbb{R}_+^V$ and real number $M > c(E)$, let $p' \in \mathbb{R}_+^V$ be defined by $p'_v = M$ for every $v \in V'$, and $p'_v = p_v$ for every $v \in V \setminus V'$. If $T^* = T_{opt}(G, \emptyset; c, p')$ is an optimal PCST in $(G, \emptyset; c, p')$, then T^* is an optimal PCST in $(G, V'; c, p)$.*

Proof. First, suppose on the contrary that T^* does not contain some target vertex $u \in V' \setminus V(T^*)$. Let P be a path in G from u to a vertex in T^* such that P intersects T^* only at this vertex, written as t. It follows that the union of T^* and P, written as $T^* \cup P$ is a tree in G and its value $\nu(T^* \cup P, c, p')$ is smaller than $\nu(T^*, c, p')$ as seen from the following (in) equalities:

$$\begin{aligned} \nu(T^* \cup P, c, p') =& \nu(T^*, c, p') + \nu(P, c, p') + p_t \\ \le& \nu(T^*, c, p') - p_u + c(E) = \nu(T^*, c, p') - M + c(E) < \nu(T^*, c, p'). \end{aligned}$$

The contradiction to the optimality of T^* proves $V' \subseteq V(T^*)$. Now, suppose on the contrary that there exists a tree T' in G with $V(T') \supseteq V'$ and $\nu(T', c, p) < \nu(T^*, c, p)$. Then

$$\nu(T', c, p') = \nu(T', c, p) + p(V') - p'(V') < \nu(T^*, c, p) + p(V') - p'(V') = \nu(T^*, c, p'),$$

which contradicts the optimality of T^*. So no such a tree T' exists. The lemma is proved. □

By Lemma 2.1, when seeking for the optimal solution to PCST problem we do not need to consider the target set V'. For brevity, we remove target set V' from the input of the PCST problem in our following discussion unless otherwise noted. Accordingly, we write $T_{\text{opt}}(G, V'; c, p)$ simply as $T_{\text{opt}}(G, c, p)$.

The graph class of 2-*tree*s is defined recursively as follows: An edge is a 2-tree. Given a 2-tree, picking an edge uv in it, adding a new vertex z adjacent with both u and v yields a 2-tree. See Fig.1 for an illustration, where vertex z has degree 2 in the new 2-tree. A spanning subgraph of a

2-tree is called a *series-parallel graph* or a *partial* 2*-tree*. The constructive definition guarantees the following.

Remark 2.1. Every 2-tree is either an edge or has at least one vertex of degree 2 which is contained in a triangle (a complete graph of three vertices).

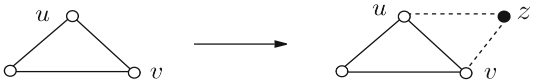


Fig.1. Construction of 2-trees

[30] gave an $O(n)$-time algorithm for finding a minimum Steiner tree in given series-parallel graph of n vertices with target set. Their algorithm is a dynamic programming method which operates in two phases: First, complete the series-parallel graph to a 2-tree, and assign a very large edge cost, say M in Lemma 2.1, to every edge added. Then find a Steiner tree on the 2-tree by repeatedly eliminating vertices of degree 2 until the remaining graph is a single edge. During this vertex elimination procedure, the algorithm records information (so called "measures") associated with triangle on u, v and z, where vertex z has degree 2 in the current 2-tree, on the ordered pairs (u, v) and (v, u) corresponding to the edge uv in G, when considering and deleting z.

Throughout the paper, completing given series-parallel graphs to 2-trees as above is assumed implicitly. We only describe our algorithms with input graphs being 2-trees.

Remark 2.2. Combined with the completion processing, all algorithms proposed in this paper extend to solve the corresponding problems on series-parallel graphs with the same time complexity.

Following the idea of vertex elimination[30], our algorithm for the PCST problem on 2-tree $G = (V, E)$ goes as follows, where the computation of measures (defined below) has to take both vertex prize and edge cost into account and thus uses different formulas from those designed by [30] for the SMT problem on 2-trees; their formulas only need to deal with edge cost. As target set is not presented (explicitly) in the PCST problem by Lemma 2.1, our task is accomplished by introducing five *measures*, instead of six as [30] did for the SMT problem, $st(u, v), dt(u, v), un(u, v), nv(u, v), nn(u, v)$ for each arc (u, v) that corresponds to edge $uv \in E$. These measures record the values computed so far for the subgraph S of G which has been reduced onto the edge uv.

$T_{st(u,v)}$ is a tree in S containing both u and v such that its value, written as $st(u, v)$, is minimum;

$T_{dt(u,v)}$ is the union of two vertex-disjoint trees in S, one containing u and the other containing v, such that its value, written as $dt(u, v)$, is the minimum;

$T_{un(u,v)}$ is a tree in S containing u but v such that its value, written as $un(u, v)$, is minimum;

$T_{nv(u,v)}$ is a tree in S containing v but u such that its value, written as $nv(u, v)$, is minimum;

$T_{nn(u,v)}$ is a tree in S containing neither u nor v such that its value, written as $nn(u, v)$, is minimum.

Note that $T_{nn(u,v)}$ might be empty, i.e., possibly $T_{nn(u,v)} = (\emptyset, \emptyset)$. For brevity, we put $\Pi \equiv \{st, dt, nv, un, nn\}$ and say that measure $\pi(u, v)$ *corresponds* to forest $T_{\pi(u,v)}$ for every $\pi \in \Pi$, where the graph theoretical term "*forest*" refers to any acyclic graph, i.e., a union of vertex-disjoint trees. Since edge uv can also be written as vu, we shall use implicitly the relations:

$$st(u, v) \equiv st(u, v), \qquad dt(u, v) \equiv dt(v, u), \qquad un(u, v) \equiv nu(v, u),$$
$$nv(u, v) \equiv vn(v, u), \qquad nn(u, v) \equiv nn(v, u).$$

If measures $st(u,v), un(u,v), nv(u,v), nn(u,v)$ of an edge uv in G for $S = G$ are available, then the minimum among them is clearly the value of an optimal PCST. On the other hand, computing these measures for $S = G$ is time consuming if one simply enumerates an exponential number of trees in G. To get around the difficulty, we consider an $O(m + n)$ number of subgraphs S of G, and update the measures together with their corresponding forests in a dynamic programming manner (when reducing the graph sequentially) until the measures for $S = G$ are obtained (the graph is reduced to a single edge).

Initially, graph F is set to be G, and every edge $uv \in E(F) = E$ is *reduced* to itself (i.e., $S = uv$). Furthermore, after initial setting

$$\begin{aligned} &st(u,v) = c_{uv} - p_u - p_v, \qquad T_{st(u,v)} = (\{u,v\}, uv);\\ &dt(u,v) = -p_u - p_v, \qquad T_{dt(u,v)} = (\{u,v\}, \emptyset);\\ &un(u,v) = -p_u, \qquad T_{un(u,v)} = (\{u\}, \emptyset);\\ &nv(u,v) = -p_v, \qquad T_{nv(u,v)} = (\{v\}, \emptyset);\\ &nn(u,v) = 0, \qquad T_{nn(u,v)} = (\emptyset, \emptyset); \end{aligned} \tag{1}$$

the following has been satisfied:

Remark 2.3. Every edge uv of the graph F is associated with five measures $\pi(u,v)$, $\pi \in \Pi$, and their corresponding forests such that (i)–(v) are satisfied for the subgraph S of G which has been reduced onto uv.

If F is an edge, then we are done. Consider F of at least three vertices. By Remark 2.1, F has a triangle S on vertices u, v, z, where z has degree 2 in F. We shall *reduce* this triangle S in F onto the edge uv by removing vertex z together its incident edge uz, zv from F, and at the same time updating measures of uv together with their corresponding forests as specified below. We proceed by making a couple of remarks.

Remark 2.4. S is also considered as the subgraph of G which is the union of subgraphs of G that have been reduced onto uv, vz, uz, respectively.

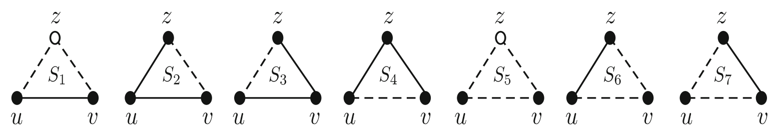


Fig.2. Trees in a Triangle

Remark 2.5. As Fig. 2 shows, there are exactly four trees S_1, S_2, S_3, S_4 in S containing both u and v. These trees are also viewed as trees in G given by

$$\begin{aligned} S_1 &= T_{st(u,v)} \cup T_{un(u,z)} \cup T_{nv(z,v)}, \qquad & S_2 &= T_{st(u,v)} \cup T_{st(u,z)} \cup T_{dt(z,v)},\\ S_3 &= T_{st(u,v)} \cup T_{dt(u,z)} \cup T_{st(z,v)}, \qquad & S_4 &= T_{dt(u,v)} \cup T_{st(u,z)} \cup T_{st(z,v)}. \end{aligned}$$

Remark 2.6. As Fig.2 shows, there are exactly three forests S_5, S_6, S_7 in S each of which consists of two vertex-disjoint trees, one containing u and the other containing v. These forests when viewed in G are given by $S_5 = T_{dt(u,v)} \cup T_{un(u,z)} \cup T_{nv(z,v)}$, $S_6 = T_{dt(u,v)} \cup T_{st(u,z)} \cup T_{dt(z,v)}$, $S_7 = T_{dt(u,v)} \cup T_{dt(u,z)} \cup T_{st(z,v)}$.

Remark 2.7. There are exactly two trees $S_8 = T_{un(u,v)} \cup T_{un(u,z)}$ and $S_9 = T_{un(u,v)} \cup T_{st(u,z)} \cup T_{zn(z,v)}$ in S each of which contains u but v. There are exactly two trees $S_{10} = T_{nv(u,v)} \cup T_{nv(z,v)}$ and $S_{11} = T_{nv(u,v)} \cup T_{nz(u,z)} \cup T_{st(z,v)}$ in S each of which contains v but u.

Remark 2.8. There are exactly four trees $S_{12} = T_{nn(u,v)}$, $S_{13} = T_{nn(u,z)}$, $S_{14} = T_{nn(z,v)}$, $S_{15} = T_{nz(u,z)} \cup T_{zn(z,v)}$ in S each of which contains neither u nor v. $T_{nn}(u, z)$ is a tree containing neither u nor z. We use this tree (notation) just before the step that deletes z. At that moment, $T_{nn}(u, z)$ means a tree in the subgraph S reduced to edge uz that contains neither u nor z. Note the vertices of S except for u and z have been deleted. Because v is still present, we see that it is outside S. It follows that the tree $T_{nn}(u, z)$ does not contain v. Similar reasoning shows that $T_{nn}(z, v)$ is a tree containing neither u nor v.

Remark 2.9. According to Remarks 2.5–2.8, the values of forests S_i, $i = 1, 2, \cdots, 15$, in G can be computed by using current measures as follows:

- $\nu(S_1, c, p) = st(u, v) + un(u, z) + nv(z, v) + p_u + p_v$,
  $\nu(S_2, c, p) = st(u, v) + st(u, z) + dt(z, v) + p_u + p_z + p_v$,
  $\nu(S_3, c, p) = st(u, v) + dt(u, z) + st(z, v) + p_u + p_z + p_v$,
  $\nu(S_4, c, p) = dt(u, v) + st(u, z) + st(z, v) + p_u + p_z + p_v$;
- $\nu(S_5, c, p) = -p_u - p_v = dt(u, v) + un(u, z) + nv(z, v) + p_u + p_v$,
  $\nu(S_6, c, p) = c_{uz} - p_u - p_z - p_v = dt(u, v) + st(u, z) + dt(z, v) + p_u + p_z + p_v$,
  $\nu(S_7, c, p) = c_{zv} - p_u - p_z - p_v = dt(u, v) + dt(u, z) + st(z, v) + p_u + p_z + p_v$;
- $\nu(S_8, c, p) = un(u, v) + un(u, z) + p_u$, $\nu(S_9, c, p) = un(u, v) + st(u, z) + zn(z, v) + p_u + p_z$;
- $\nu(S_{10}, c, p) = nv(u, v) + nv(z, v) + p_v$, $\nu(S_{11}, c, p) = nv(u, v) + nz(u, z) + st(z, v) + p_z + p_v$;
- $\nu(S_{12}, c, p) = nn(u, v)$, $\nu(S_{13}, c, p) = nn(u, z)$, $\nu(S_{14}, c, p) = nn(z, v)$,
  $\nu(S_{15}, c, p) = nz(u, z) + zn(z, v) + p_z$.

For any set $\mathcal{S}$ of subgraphs of G, we use $\mathrm{Min}\,(\mathcal{S})$ to denote an arbitrary subgraph $R \in \mathcal{S}$ whose value $\nu(R, c, p)$ is the minimum among all elements of $\mathcal{S}$. Now we update the measures of uv and their corresponding forests by

$$\begin{aligned}
st(u,v) &= \min_{i=1}^{4} \nu(S_i, c, p), \qquad && T_{st(u,v)} = \mathrm{Min}(\{S_1, S_2, S_3, S_4\}); \\
dt(u,v) &= \min_{i=5}^{7} \nu(S_i, c, p), \qquad && T_{dt(u,v)} = \mathrm{Min}(\{S_5, S_6, S_7\}); \\
un(u,v) &= \min_{i=8}^{9} \nu(S_i, c, p), \qquad && T_{un(u,v)} = \mathrm{Min}(\{S_8, S_9\}); \\
nv(u,v) &= \min_{i=10}^{11} \nu(S_i, c, p), \qquad && T_{nv(u,v)} = \mathrm{Min}(\{S_{10}, S_{11}\}); \\
nn(u,v) &= \min_{i=12}^{15} \nu(S_i, c, p), \qquad && T_{nn(u,v)} = \mathrm{Min}(\{S_{12}, S_{13}, S_{14}, S_{15}\}).
\end{aligned} \tag{2}$$

It is routine to check from Remarks 2.5–2.8 that, with the update (2), *Remark 2.3 still holds* for the new F (with z and its two incident edges deleted), after reducing S which is a triangle in the old F (with z and both incident edges present) and also a subgraph of G (recalling Remark 2.4).

Clearly, the new graph F remains a 2-tree. We continue the process-picking a triangle in F and reducing it. The process is repeated to update measures and their corresponding forests, until F is reduced to a single edge uv (equivalently, G has been reduced onto edge uv). Proceeding inductively, we have shown that Remark 2.3 holds throughout.

The following pseudo-code gives the details of our algorithm, where the first part (Steps 2–4) initializes measures; the second part (Steps 5–10) repeatedly updates measures and corresponding forests until only one edge is left (equivalently, current graph contains no vertex of degree 2); and the last part (Steps 11–13) picks the minimum among the final measures $st(u, v), un(u, v), nv(u, v), nn(u, v)$, and outputs its corresponding tree in G as the solution to the PCST instance on 2-tree G.

Algorithm for PCST on 2-tree (ALG_PCST)

Input 2-tree $G = (V, E)$ with $c \in \mathbb{R}_+^E,\ p \in \mathbb{R}_+^V$
Output An optimal tree T^* of the PCST problem on (G, c, p) with optimal value ν^*

1. $F \leftarrow G$
2. **for** every $uv \in E$ **do begin**
3. Set measure $\pi(u, v)$ and its corresponding forest $T_{\pi(u,v)}$, for $\pi \in \Pi$, as in (1)
4. **end-for**
5. **while** F contains more than two vertices **do begin**
6. Take vertex $z \in V(F)$ of degree 2 in F, and $uz, zv \in E(F)$
7. Compute forests S_i and their values $\nu(S_i, c, p)$, $i = 1, 2, \cdots, 15$ as in Remarks 2.5–2.9
8. Update measure $\pi(u, v)$ and its corresponding forest $T_{\pi(u,v)}$, for $\pi \in \Pi$, as in (1)
9. Remove z from F
10. **end-while**
11. $\nu^* \leftarrow \min\big\{st(u, v), un(u, v), nv(u, v), nn(u, v)\big\}$, where $E(F) = \big\{uv\big\}$
12. $T^* \leftarrow T_{\pi(u,v)}$, where $\pi \in \{st, un, nv, nn\}$ and $\pi(u, v) = \nu^*$
13. Output T^* and ν^*

Theorem 2.2. *Given any PCST instance on a 2-tree of n vertices, Algorithm* ALG_PCST *outputs its optimal PCST and optimal value in $O(n)$ time.*

Proof. Let T^* and ν^* be the output of ALG_PCST at Step 13. Recalling Remark 2.3, it follows from Steps 11–13 that T^* is a tree in G and has value $\nu(T^*, c, p) = \nu^*$. Let uv be the final edge of F at Step 11. Then, by Step 11, $\nu^* = \min\{st(uv), un(u, v), nv(u, v), nn(uv)\}$ equals the value of an optimal PCST, showing that T^* is an optimal PCST in G. Hence ALG_PCST solves the PCST problem on 2-trees exactly.

Note that it takes $O(1)$ time for ALG_PCST to finish a single implementation of Step 3 and that of Steps 6–8. Since the for-loop (Steps 2–4) repeats $|E|$ times, the while-loop (Steps 5–10) repeats $|V|$ times, the algorithm runs in time $O(|V| + |E|)$, which is $O(n)$, as $|E| = 2|V| - 3$ holds for every 2-tree $G = (V, E)$. □

To facilitate understanding, we elaborate the implementation ALG_PCST on the example depicted in Fig.3, where the numbers beside the edges and vertices are their associated costs and prizes.

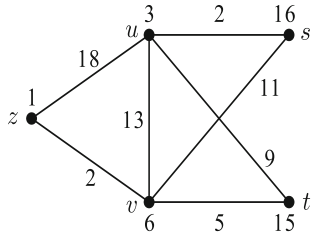


Fig.3. An Instance for the PCST on 2-tree

For brevity, we use $\Pi(\dot{u},\dot{v}) \equiv (st(\dot{u},\dot{v}), dt(\dot{u},\dot{v}), \dot{u}n(\dot{u},\dot{v}), n\dot{v}(\dot{u},\dot{v}), nn(\dot{u},\dot{v})) \in \mathbb{R}^5$ to express the measures of edge $\dot{u}\dot{v}$ in the graph in operation. Firstly, Steps 2–4 initialize the measures together with the corresponding forests

$$\begin{array}{lll} T_{st(\dot{u},\dot{v})} = (\{\dot{u},\dot{v}\},\dot{u}\dot{v}), & T_{dt(\dot{u},\dot{v})} = (\{\dot{u},\dot{v}\},\emptyset), & T_{\dot{u}n(\dot{u},\dot{v})} = (\{u\},\emptyset), \\ T_{n\dot{v}(\dot{u},\dot{v})} = (\{v\},\emptyset), & T_{nn(\dot{u},\dot{v})} = (\emptyset,\emptyset) & \end{array} \tag{3}$$

for every edge $\dot{u}\dot{v}$ in the graph, and obtain

$$\Pi(u,s) = (-17,-19,-3,-16,0), \qquad \Pi(s,v) = (-11,-22,-16,-6,0),$$
$$\Pi(u,v) = (4,-9,-3,-6,0); \tag{4}$$
$$\Pi(u,t) = (-9,-18,-3,-15,0), \qquad \Pi(t,v) = (-16,-21,-15,-6,0); \tag{5}$$
$$\Pi(u,z) = (14,-4,-3,-1,0), \qquad \Pi(z,v) = (-5,-7,-1,-6,0). \tag{6}$$

Then Steps 5–9 update the measures and the corresponding forests when deleting vertices s,t,z from the graph in order. So the measures and the corresponding forests of uv are updated three times before the whole graph is reduced to a single edge uv.

First, when deleting s, by using (3) for $\dot{u}\dot{v} \in \{us, sv, uv\}$ and (4), Alg_Pcst finds

$$\begin{aligned}
&\mathrm{Min}(\{S_1,S_2,S_3,S_4\}) = S_4 = T_{dt(u,v)} \cup T_{st(u,s)} \cup T_{st(s,v)} = (\{u,s,v\},\{us,sv\}),\\
&\mathrm{Min}(\{S_5,S_6,S_7\}) = S_6 = T_{dt(u,v)} \cup T_{st(u,s)} \cup T_{dt(s,v)} = (\{u,s,v\},us),\\
&\mathrm{Min}(\{S_8,S_9\}) = S_9 = T_{un(u,v)} \cup T_{st(u,s)} \cup T_{sn(s,v)} = (\{u,s\},us),\\
&\mathrm{Min}(\{S_{10},S_{11}\}) = S_{11} = T_{nv(u,v)} \cup T_{ns(u,s)} \cup T_{st(s,v)} = (\{s,v\},sv),\\
&\mathrm{Min}(\{S_{12},S_{13},S_{14},S_{15}\}) = S_{15} = T_{ns(u,s)} \cup T_{sn(s,v)} = (\{s\},\emptyset);
\end{aligned}$$

and accordingly updates

$$\begin{aligned}
&\Pi(u,v) = (-12,-23,-17,-11,-16); \quad T_{st(u,v)} = (\{u,s,v\},\{us,sv\}), \quad T_{dt(u,v)} = (\{u,s,v\},us),\\
&T_{un(u,v)} = (\{u,s\},us), \quad T_{nv(u,v)} = (\{s,v\},sv), \quad T_{nn(u,v)} = (\{s\},\emptyset).
\end{aligned} \tag{7}$$

Second, when deleting t, by using (3) for $\dot{u}\dot{v} \in \{ut, tv\}$, (5) and (7), Alg_Pcst finds

$$\begin{aligned}
&\mathrm{Min}(\{S_1,S_2,S_3,S_4\}) = S_4 = T_{dt(u,v)} \cup T_{st(u,t)} \cup T_{st(t,v)} = (\{u,v,s,t\},\{us,ut,tv\}),\\
&\mathrm{Min}(\{S_5,S_6,S_7\}) = S_7 = T_{dt(u,v)} \cup T_{dt(u,t)} \cup T_{st(t,v)} = (\{u,s,v,t\},\{us,tv\}),\\
&\mathrm{Min}(\{S_8,S_9\}) = S_9 = T_{un(u,v)} \cup T_{st(u,t)} \cup T_{tn(t,v)} = (\{u,s,t\},\{us,ut\}),\\
&\mathrm{Min}(\{S_{10},S_{11}\}) = S_{11} = T_{nv(u,v)} \cup T_{nt(u,t)} \cup T_{st(t,v)} = (\{s,v,t\},\{sv,tv\}),\\
&\mathrm{Min}(\{S_{12},S_{13},S_{14},S_{15}\}) = S_{12} = T_{nn(u,v)} = (\{s\},\emptyset);
\end{aligned}$$

and accordingly updates

$$\begin{aligned}
&\Pi(u,v) = (-24,-33,-23,-21,-16);\\
&T_{st(u,v)} = (\{u,v,s,t\},\{us,ut,tv\}), \qquad T_{dt(u,v)} = (\{u,s,v,t\},\{us,tv\}),\\
&T_{un(u,v)} = (\{u,s,t\},\{us,ut\}), \qquad T_{nv(u,v)} = (\{s,v,t\},\{sv,tv\}), \qquad T_{nn(u,v)} = (\{s\},\emptyset).
\end{aligned} \tag{8}$$

Third, when deleting z, by using (3) for $\dot{u}\dot{v} \in \{uz, zv\}$, (6) and (8), ALG_PCST finds

$$\begin{aligned}
&\text{Min}(\{S_1, S_2, S_3, S_4\}) = S_1 = T_{st(u,v)} \cup T_{un(u,z)} \cup T_{nv(z,v)} = (\{u, v, s, t\}, \{us, ut, tv\}),\\
&\text{Min}(\{S_5, S_6, S_7\}) = S_5 = T_{dt(u,v)} \cup T_{un(u,z)} \cup T_{nv(z,v)} = (\{u, s, v, t\}, \{us, tv\}),\\
&\text{Min}(\{S_8, S_9\}) = S_8 = T_{un(u,v)} \cup T_{un(u,z)} = (\{u, s, t\}, \{us, ut\}),\\
&\text{Min}(\{S_{10}, S_{11}\}) = S_{10} = T_{nv(u,v)} \cup T_{nv(z,v)} = (\{s, v, t\}, \{sv, tv\}),\\
&\text{Min}(\{S_{12}, S_{13}, S_{14}, S_{15}\}) = S_{12} = T_{nn(u,v)} = (\{s\}, \emptyset);
\end{aligned}$$

and accordingly updates

$$\begin{aligned}
&\Pi(u, v) = (-24, -33, -23, -21, -16); \qquad T_{st(u,v)} = (\{u, v, s, t\}, \{us, ut, tv\}),\\
&T_{dt(u,v)} = (\{u, s, v, t\}, \{us, tv\}), \qquad T_{un(u,v)} = (\{u, s, t\}, \{us, ut\}),\\
&T_{nv(u,v)} = (\{s, v, t\}, \{sv, tv\}), \qquad T_{nn(u,v)} = (\{s\}, \emptyset).
\end{aligned}$$

Finally, since only one edge, which is uv, is left, ALG_PCST outputs

$$\nu^* = \min\{-24, -23, -21, -16\} = -24 = st(u, v) \qquad T^* = T_{st(u,v)} = (\{u, v, s, t\}, \{us, ut, tv\}).$$

It is easy to see from Lemma 2.1 that with easy preprocessing, ALG_PCST works for instances with nonempty target sets. As observed from the above example, our algorithm for the PCST is more versatile than Wald-Colbourn algorithm for the SMT: with an empty target set, our algorithm outputs a tree of value -24, saying that it is profitable; while Wald-Colbourn algorithm returns nothing because it never starts.

3 Two Risk Models

In this section, we consider the PCST problem with interval data. Given a undirected graph $G = (V, E)$, each edge $e \in E$ is associated with a cost interval $[c_e^-, c_e^+]$, and each vertex $v \in V$ is associated with a prize interval $[p_v^-, p_v^+]$. These intervals indicate possible ranges of construction cost of edge e and collection prize of vertex v, respectively. We define the *risk* at edge e as

$$r(x_e) \equiv \frac{c_e^+ - x_e}{c_e^+ - c_e^-}$$

when charging cost $x_e \in [c_e^-, c_e^+]$ and the *risk* at vertex v as

$$r(y_v) \equiv \frac{y_v - p_v^-}{p_v^+ - p_v^-}$$

when collecting prize $y_v \in [p_v^-, p_v^+]$. For ease of description, we make the notational convention that $\frac{0}{0} = 0$. With these definitions, risks $r(x_e)$ and $r(y_v)$ both range from 0 to 1. In particular, $r(x_e) = 0$ when $x_e = c_e^+$ $(r(y_v) = 0$ when $y_v = p_v^-)$, meaning no risk occurs if the payment is high enough (the expected prize is low enough). On the other hand, $r(x_e) = 1$ when $x_e = c_e^-$ $(r(y_v) = 1$ when $y_v = p_v^+)$, meaning a full risk is doomed at the lowest payment (the highest prize). Let $\mathfrak{T}$ denote the set of trees in G. We define the value of $T \in \mathfrak{C}$ with charged payment $x \in \mathbb{R}_+^{E(T)}$ and collected prize $y \in \mathbb{R}_+^{V(T)}$ as

$$\nu(T, x, y) \equiv x(E(T)) - y(V(T)),$$

where by $x \in \mathbb{R}_+^{\emptyset}$ (in case of $E(T) = \emptyset$) we mean that x is a null vector which will be written as NULL.

Let B be a given budget bound on constructing a PCST in G. It is required that $T \in \mathfrak{T}$ with charged payment x and collected prize y satisfy $\nu(T,x,y) \le \mathsf{B}$. So in case of negative B, the construction of tree T must make profit. To assure the feasibility of $\nu(T,x,y) \le \mathsf{B}$ subject to

$$x_e \in [c_e^-, c_e^+], \quad \forall\, e \in E(T) \qquad \text{and} \qquad y_v \in [p_v^-, p_v^+], \quad \forall\, v \in V(T), \tag{9}$$

bound B obviously cannot be smaller than, $\mathsf{B}_{\min}$, the optimal value of the PCST problem with respect to $c_e = c_e^-$ for every $e \in E$ and $p_v = p_v^+$ for every $v \in V$. Meanwhile, $\mathsf{B}_{\max} \equiv \mathsf{c}^+(\mathsf{E})$ is a trivial upper bound on $\nu(T,x,y)$ for any $T \in \mathfrak{T}$, $x \in \mathbb{R}_+^{E(T)}$ and $y \in \mathbb{R}_+^{V(T)}$ satisfying (9). We assume $\mathsf{B} \in [\mathsf{B}_{\min}, \mathsf{B}_{\max}]$ throughout.

In Subsections 3.1 and 3.2 below, we will study, respectively, two risk models for the PCST problem that adopt distinct objective functions: min-max risk and min-sum risk, under budget constraints. We deduce evenness property and extremeness property for the optimal solutions of these two models, respectively. Building on these properties, we design strongly polynomial-time algorithms solving the PCST problem on series-parallel graphs under these two risk models to the optimality.

3.1 PCST Problem under Min-Max Risk Model

The PCST problem under min-max risk model, denoted by MMR_PCST, is to find a tree T along with payment x and prize y such that the maximum risk at edges and vertices in T is minimized and the value $\nu(T,x,y)$ is no greater than the given budget B. This problem can be formulated as follows:

$$\begin{aligned}
\text{(MMR_PCST)} \qquad & \min_{T\in\mathfrak{T},\nu(T,x,y)\le\mathsf{B}} \ \max_{e\in E(T), v\in V(T)} \{(c_e^+ - x_e)/(c_e^+ - c_e^-), (y_v - p_v^-)/(p_v^+ - p_v^-)\} \\
& \text{s.t. } x_e \in [c_e^-, c_e^+],\ \forall\, e \in E(T); \quad y_v \in [p_v^-, p_v^+],\ \forall\, v \in V(T).
\end{aligned}$$

Let (T^*, x^*, y^*) be an optimal solution to the MMR_PCST problem, where T^* is called an *optimal tree*. We reserve symbol r^* for the value $r_m(T^*, x^*, y^*)$ of the optimal solution, i.e.,

$$r^* \equiv r_m(T^*, x^*, y^*) \equiv \max_{e\in E(T^*), v\in V(T^*)} \Big\{ \frac{c_e^+ - x_e^*}{c_e^+ - c_e^-}, \frac{y_v^* - p_v^-}{p_v^+ - p_v^-} \Big\}. \tag{10}$$

The following lemma shows that (T^*, x^*, y^*) possesses an evenness property-the risks of edges and vertices are all equal, which will play an important role in our algorithm design.

Lemma 3.1. *Evenness property] For every edge e and every vertex v in T^*, it holds that*

$$\frac{c_e^+ - x_e^*}{c_e^+ - c_e^-} = \frac{y_v^* - p_v^-}{p_v^+ - p_v^-} = r^*.$$

Proof. Note from (10) that

$$Q = \Big\{ e \in E(T^*) : \frac{c_e^+ - x_e^*}{c_e^+ - c_e^-} = r^* \Big\} \bigcup \Big\{ v \in V(T^*) : \frac{y_v^* - p_v^-}{p_v^+ - p_v^-} = r^* \Big\} \ne \emptyset$$

and if the lemma fails, then there exists $f \in E(T^*)$ with $0 \le (c_f^+ - x_f^*)/(c_f^+ - c_f^-) < r^*$ or $u \in V(T^*)$ with $0 \le (y_u^* - p_u^-)/(p_u^+ - p_u^-) < r^*$. Consequently, we can take sufficiently small $\varepsilon > 0$ such that the MMR_PCST has a solution (T^*, x', y') with $x' \in \mathbb{R}_+^{E(T^*)}$ and $y' \in \mathbb{R}_+^{V(T^*)}$

given by

$$x'_e = \begin{cases} x^*_e + \varepsilon, & \text{if } e \in Q \cap E, \\ x^*_e - |Q|\varepsilon, & \text{if } e = f, \\ x^*_e, & \text{otherwise,} \end{cases} \qquad y'_v = \begin{cases} y^*_v - \varepsilon, & \text{if } v \in Q \cap V, \\ y^*_v, & \text{otherwise,} \end{cases}$$

$$x'_e = \begin{cases} x^*_e + \varepsilon, & \text{if } e \in Q \cap E, \\ x^*_e, & \text{otherwise,} \end{cases} \qquad y'_v = \begin{cases} y^*_v - \varepsilon, & \text{if } v \in Q \cap V, \\ y^*_v + |Q|\varepsilon, & \text{if } v = u, \\ y^*_v, & \text{otherwise.} \end{cases}$$

In either case, we have $r_m(T^*, x', y') < r^*$, a contradiction to the optimality of r^*, proving the lemma. □

For determining the optimal value r^*, we introduce more notations. For any $r \in [0, 1]$, let

$$x^r_e = c^+_e - r(c^+_e - c^-_e) \text{ for every } e \in E, \qquad y^r_v = p^-_v + r(p^+_v - p^-_v) \text{ for every } v \in V, \tag{11}$$

and let $T^r = T_{\text{opt}}(G, \emptyset; x^r, y^r) \in \mathfrak{T}$ be an optimal PCST with respect cost x^r and prize y^r. Then

$$\nu(T^r, x^r, y^r) = \min_{T \in \mathfrak{T}} \nu(T, x^r, y^r) \quad \text{ for all } \ r \in [0, 1]. \tag{12}$$

Observe that the restriction of x^{r^*} to $E(T^*)$, denoted by $x^{r^*}|_{E(T^*)}$, is exactly x^*, and the restriction of y^{r^*} to $V(T^*)$, denoted by $y^{r^*}|_{V(T^*)}$, is exactly y^*. Therefore (12) guarantees $\nu(T^{r^*}, x^{r^*}, y^{r^*}) \leq \nu(T^*, x^*, y^*) \leq \mathsf{B}$. By the definition of r^* in (10), it turns out that $(T^{r^*}, x^{r^*}, y^{r^*})$ is an optimal solution to the MMR_PCST problem, which allows us to assume $T^* = T^{r^*}$. The next lemma shows that the budget bound B acts as a threshold in deriving an estimation on the value of r in comparison with r^*.

Lemma 3.2. *If $\nu(T^r, x^r, y^r) > \mathsf{B}$, then $r < r^*$; otherwise $r \geq r^*$.*

Proof. If $r \geq r^*$, then by definitions in (11), we have $x^r_e \leq x^{r^*}_e$ for every $e \in E$, $y^r_v \geq y^{r^*}_v$ for every $v \in V$, and therefore $\nu(T^{r^*}, x^r, y^r) \leq \nu(T^{r^*}, x^{r^*}, y^{r^*}) \leq \mathsf{B}$. Note from (12) that $\nu(T^r, x^r, y^r) \leq \nu(T^{r^*}, x^r, y^r)$. Thus $\nu(T^r, x^r, y^r) \leq \mathsf{B}$, provided $r \geq r^*$. The equivalent statement reads: if $\nu(T^r, x^r, y^r) > \mathsf{B}$, then $r < r^*$.

Recalling (10), the optimality of r^* implies $\nu(T^r, x^r, y^r) > \mathsf{B}$ provided $r < r^*$. Hence, if $\nu(T^r, x^r, y^r) \leq \mathsf{B}$, then $r \geq r^*$. □

From the definitions of x^r, y^r, T^r, we have (T^0, x^0, y^0) a trivial optimal solution to the MMRPCST problem if $\nu(T^0, x^0, y^0) \leq \mathsf{B}$. To avoid triviality, we assume $\nu(T^0, x^0, y^0) > \mathsf{B}$ whenever we discuss the MMRPCST problem. It follows from Lemma 3.2 and (10) that

$$r^* \in (0, 1], \quad \text{ which in turn implies } \quad \nu(T^*, x^*, y^*) = \mathsf{B}, \tag{13}$$

as otherwise we could increase x^*_e a little bit for every edge $e \in E(T^*)$, decrease y^*_v a little bit for every vertex $v \in V(T^*)$, and obtain a smaller r^*. By virtue of (13), Lemmas 3.1 and 3.2, we can apply Megiddo's parametric search method[24], to determine in polynomial time the value of r^* for the MMR_PCST on series-parallel graphs, as shown in the following algorithm Alg_Mmr. Given input of 2-tree G, Alg_Mmr works on a graph F, which is reduced from G by deleting degree-2 vertices step by step, and a search interval $[r_l, r_u]$, which is narrowed down from $[0, 1]$ recursively, and is always guaranteed to contain r^*. (The detailed pseudo-code of Alg_Mmr is available upon request to the corresponding author).

Algorithm for MMR_PCST on 2-tree (Alg_Mmr)

Input 2-tree $G = (V, E)$ with $c^- \in \mathbb{R}^E_+$, $c^+ \in \mathbb{R}^E_+$, $p^- \in \mathbb{R}^V_+$, $p^+ \in \mathbb{R}^V_+$, $\mathsf{B} \in \mathbb{R}_+$
Output The optimal value r^* of the MMR_PCST problem along with its optimal tree T^*

1. Initially, set $F \leftarrow G$ and $[r_l, r_u] \leftarrow [0, 1]$.

2. Let r be a symbol representing a variable in $[r_l, r_u]$. The algorithm proceeds to solve the PCST problem on (G, x^r, y^r) in a similar way to ALG_PCST (see Page 7). To avoid confusion, the five measures associated with $uv \in E$ (cf. (i)–(v) on Page 5), for the PCST on (G, x^r, y^r), are written as $st^r(u,v), dt^r(u,v), un^r(u,v), nv^r(u,v), nn^r(u,v)$ with superscript r indicating the cost metric x^r and prize metric y^r.

3. Similar to Steps 2–4 of ALG_PCST, the algorithm initializes five measures $st^r(u,v) \leftarrow x^r_{uv} - y^r_u - y^r_v$, $dt^r(u,v) \leftarrow -y^r_u - y^r_v, \cdots$ for every $uv \in E$, where $st^r(u,v), dt^r(u,v), un^r(u,v), nv^r(u,v), nn^r(u,v)$ are all set to be linear functions of $r \in [r_l, r_u]$ (recalling (1) and (11)).

4. The algorithm repeatedly deletes degree-2 vertices from F, update measures in a way analogous to Steps 5–10 of ALG_PCST, and narrows down $[r_l, r_u]$ as well.

   4.1. Take degree-2 vertex z in F and $uz, zv \in E(F)$.

   4.2. Compute $\nu(S_1, x^r, y^r) \leftarrow st^r(u,v) + un^r(u,z) + nv^r(z,v) + y^r_u + y^r_v, \cdots \nu(S_{15}, x^r, y^r) \leftarrow nz^r(u,z) + zn^r(z,v) + y^r_z$ (cf. Remark 2.9), which are fifteen linear functions of $r \in [r_l, r_u]$.

   4.3. Update measures $st^r(u,v) \leftarrow \min_{i=1}^{4} \nu(S_i, x^r, y^r), \cdots, nn^r(u,v) \leftarrow \min_{i=12}^{15} \nu(S_i, x^r, y^r)$ (cf. (2)) by applying the $O(k \log k)$ algorithm[24] for finding the minimum of k linear functions on the same interval $[a, b]$, which is a piecewise linear function with at most $k-1$ nondifferentiable points in (a, b).

   4.4. Narrow down $[r_l, r_u]$ to make every measure $\pi^r(u,v)$, $\pi \in \{st, dt, un, nv, nn\}$ a linear function of $r \in [r_l, r_u]$ as follows. Whenever $\pi^r(u,v)$ is not linear on $[r_l, r_u]$, the algorithm finds a nondifferentiable point $r_\circ \in (r_l, r_u)$ of it, applies ALG_PCST to compute $\nu(T^{r_\circ}, x^{r_\circ}, y^{r_\circ})$, and then narrows interval $[r_l, r_u] \leftarrow [r_\circ, r_u]$ if $\nu(T^{r_\circ}, x^{r_\circ}, y^{r_\circ}) > \mathsf{B}$ and $[r_l, r_u] \leftarrow [r_l, r_\circ]$ otherwise. (After the narrowing, r_o is not a nondifferentiable point of $\pi^r(u,v)$ on (r_l, r_u), and by Lemma 3.2, interval $[r_l, r_u]$ still contains r^*.) Hence the step accomplishes its goal in $O(n)$ time.

   4.5. Delete z along with its incident edges uz, zv from F.

   Steps 4.1–4.5 are repeated (in the process the algorithm also builds up forests corresponding to measures) until F becomes a single edge uv.

5. Applying the approach used in Step 4.4, the algorithm first derives piecewise linear function $\nu^r \leftarrow \min\{st^r(u,v), un^r(u,v), nv^r(u,v), nn^r(u,v)\}$, and then narrows down the interval $[r_l, r_u]$ such that $\nu^r = \pi^r(u,v)$ is a linear function of $r \in [r_l, r_u]$, for some $\pi \in \{st, un, nv, nn\}$. (Note that the tree $T^* \leftarrow T_{\pi^r(u,v)} \in \mathfrak{T}$ corresponding to $\pi^r(u,v)$ is an optimal PCST on (G, x^r, y^r) for all $r \in [r_l, r_u]$.)

6. Let $r^* \in [r_l, r_u]$ be a solution to the equation $\nu^r = \mathsf{B}$ in variable $r \in [r_l, r_u]$. Then output r^* and T^*.

By (13), the r^* found this way is the optimal value of the MMR_PCST problem. Clearly, $T^* = T_{\pi^{r^*}(u,v)}$ is the corresponding optimal tree. The proof of the following theorem could not be included due to the space limit, which is available upon request.

Theorem 3.3. *Given any MMR_PCST instance on a 2-tree of n vertices, Algorithm* ALG_MMR *outputs its optimal value r^* in $O(n^2)$ time.*

3.2 PCST Problem under Min-Sum Risk Model

The PCST problem under min-sum risk model, denoted by MSR_PCST, is to find a tree T in given graph $G=(V,E)$ along with payment $x\in\mathbb{R}_+^{E(T)}$ and prize $y\in\mathbb{R}_+^{V(T)}$ such that the sum of risks at edges and vertices in T:

$$r_s(T,x,y)\equiv\sum_{e\in E(T)}\frac{c_e^+-x_e}{c_e^+-c_e^-}+\sum_{v\in V(T)}\frac{y_v-p_v^-}{p_v^+-p_v^-}=\sum_{e\in E(T)}r(x_e)+\sum_{v\in V(T)}r(y_v)$$

is minimized and the value $\nu(T,x,y)$ is no greater than the given budget bound $\mathsf{B}\in[\mathsf{B}_{\min},\mathsf{B}_{\max}]$. This problem can be formulated as follows:

$$\begin{aligned}(\text{MSR\_PCST})\qquad &\min_{T\in\mathfrak{T},\nu(T,x,y)\le\mathsf{B}}\Big(\sum_{e\in E(T)}\frac{c_e^+-x_e}{c_e^+-c_e^-}+\sum_{v\in V(T)}\frac{y_v-p_v^-}{p_v^+-p_v^-}\Big)\\ &\text{s.t. } x_e\in[c_e^-,c_e^+],\quad\forall\, e\in E(T);\quad y_v\in[p_v^-,p_v^+],\quad\forall\, v\in V(T).\end{aligned}$$

The MSR_PCST problem has an optimal solution that enjoys the following extremeness property-the edge payments and vertex prizes hit lower or upper limits with at most one exceptional edge and at most one exceptional vertex.

Lemma 3.4 (**Extremeness Property**). *There exists an optimal solution (T^*,x^*,y^*) to the MSR_PCST problem which contains an edge $f\in E(T^*)$ if $E(T^*)\neq\emptyset$ and a vertex $u\in V(T^*)$ such that*

$$\begin{aligned}&x_f\in[c_f^-,c_f^+],\ x_e\in\{c_e^-,c_e^+\}\qquad \textit{for all}\quad e\in E(T^*)\setminus\{f\}\ \textit{if}\ E(T^*)\neq\emptyset;\\ &y_u\in[p_u^-,p_u^+],\ y_v\in\{P_v^-,p_v^+\}\qquad \textit{for all}\quad v\in V(T^*)\setminus\{u\}.\end{aligned}\tag{14}$$

Proof. Let (T^*,x^*,y^*) be an optimal solution to the MSR_PCST problem such that the union of sets

$$\begin{aligned}\mathcal{E}(T^*,x^*,y^*)&\equiv\{e:e\in E(T^*),x_e^*\in(c_e^-,c_e^+)\},\\ \mathcal{V}(T^*,x^*,y^*)&\equiv\{v:v\in V(T^*),y_v^*\in(p_v^-,p_v^+)\}\end{aligned}$$

contains as few elements as possible. The minimality guarantees $|\mathcal{E}(T^*,x^*,y^*)|\le 1$ and $|\mathcal{V}(T^*,x^*,y^*)|\le 1$, verifying the lemma.

If $|\mathcal{E}(T^*,x^*,y^*)|>1$, then there exist distinct edges $g,f\in\mathcal{E}(T^*,x^*,y^*)$ with $c_g^+-c_g^-\le c_f^+-c_f^-$. Take $\delta=\min\{c_g^+-x_g^*,x_f^*-c_f^-\}$ and define $x'\in\mathbb{R}_+^{E(T^*)}$ by setting $x_g'=x_g^*+\delta$, $x_f'=x_f^*-\delta$ and $x_e'=x_e^*$ for every $e\in E(T^*)\setminus\{g,f\}$. It follows that (T^*,x',y^*) is an optimal solution to the MSR_PCST with $\mathcal{E}(T^*,x',y^*)\subsetneq\mathcal{E}(T^*,x^*,y^*)$ and $\mathcal{V}(T^*,x',y^*)=\mathcal{V}(T^*,x^*,y^*)$ violating the minimality of $|\mathcal{E}(T^*,x^*,y^*)|+|\mathcal{V}(T^*,x^*,y^*)|$.

If $|\mathcal{V}(T^*,x^*,y^*)|>1$, then there exist distinct vertices $u,z\in\mathcal{V}(T^*,x^*,y^*)$ with $p_u^+-p_u^-\le p_z^+-p_z^-$. Take $\delta=\min\{y_u^*-p_u^-,p_v^+-y_v^*\}$ and define $y'\in\mathbb{R}_+^{V(T^*)}$ by setting $y_u'=y_u^*-\delta$, $y_z'=y_z^*+\delta$ and $y_v'=y_v^*$ for every $v\in V(T^*)\setminus\{u,z\}$. It follows that (T^*,x^*,y') is an optimal solution to the MSR_PCST with $\mathcal{V}(T^*,x^*,y')\subsetneq\mathcal{V}(T^*,x^*,y^*)$ and $\mathcal{E}(T^*,x^*,y')=\mathcal{E}(T^*,x^*,y^*)$ violating the minimality of $|\mathcal{E}(T^*,x^*,y^*)|+|\mathcal{V}(T^*,x^*,y^*)|$. □

To find an optimal solution specified in Lemma 3.4, we employ the following algorithm ALG_GT to transform the original given 2-tree $G=(V,E)$ with $c^-,c^+\in\mathbb{R}_+^E$ and $p^-,p^+\in\mathbb{R}_+^V$ to a new graph $\widetilde{G}=(\widetilde{V},\widetilde{E})$ with $c,w\in\mathbb{R}_+^{\widetilde{E}}$ and $p,q\in\mathbb{R}_+^{\widetilde{V}}$. See Fig.4 for an illustration.

Algorithm for Graph Transformation (ALG_GT)

Input (G, c^-, c^+, p^-, p^+)
Output $(\widetilde{G}, c, p, w, q)$ along with $(\overline{G}, \overline{c}, \overline{w})$ and $(\widehat{G}, \widehat{c}, \widehat{w}, \widehat{p}, \widehat{q})$

1. Construct graph $\overline{G} = (\overline{V}, \overline{E})$ with $\overline{c}, \overline{w} \in \mathbb{R}_+^{\overline{E}}$ as follows: Set $\overline{V} \equiv V$ and $\overline{E} \equiv \{\underline{e}, \overline{e} : e \in E\}$ in such a way that every edge $e \in E$ corresponds to two edges $\underline{e}, \overline{e} \in \overline{E}$ both having the same ends as e. For every $e \in E$, set $\overline{c}_{\underline{e}} \equiv c_e^-, \overline{c}_{\overline{e}} \equiv c_e^+$; $\overline{w}_{\underline{e}} \equiv 1, \overline{w}_{\overline{e}} \equiv 0$ if $c_e^- \neq c_e^+$ and set $\overline{w}_{\underline{e}} = \overline{w}_{\overline{e}} \equiv 0$ otherwise.

2. Construct 2-tree $\widehat{G} = (\widehat{V}, \widehat{E})$ with $\widehat{c}, \widehat{w} \in \mathbb{R}_+^{\widehat{E}}$ and $\widehat{p}^-, \widehat{p}^+ \in \mathbb{R}_+^{\widehat{V}}$ as follows: Set $\widehat{V} \equiv \overline{V} \cup \{v_{\overline{e}} : e \in E\}$ and $\widehat{E} \equiv \{\underline{e} \in \overline{E} : e \in E\} \cup \{\overline{e}_1 \equiv v_{\overline{e}}u, \overline{e}_2 \equiv v_{\overline{e}}v : u, v \in \overline{V}, uv = \overline{e} \in \overline{E}\}$. For every $e \in E$, set $\widehat{c}_{\overline{e}_1} = \widehat{c}_{\overline{e}_2} \equiv \frac{1}{2}\overline{c}_{\overline{e}}$, $\widehat{w}_{\overline{e}_1} = \widehat{w}_{\overline{e}_2} \equiv \frac{1}{2}\overline{w}_{\overline{e}}$, $\widehat{p}^+_{v_{\overline{e}}} = \widehat{p}^-_{v_{\overline{e}}} \equiv 0$. For every $v \in V = \overline{V}$, set $\widehat{p}_v^- \equiv p_v^-$ and $\widehat{p}_v^+ \equiv p_v^+$.

3. Construct graph $\widetilde{G} = (\widetilde{V}, \widetilde{E})$ with $c, w \in \mathbb{R}_+^{\widetilde{E}}$ and $p, q \in \mathbb{R}_+^{\widetilde{V}}$ as follows: Set $\widetilde{V} \equiv \{v_1, v_2 : v \in \widehat{V}\}$ and $\widetilde{E} \equiv \{u_1v_1, u_1v_2, u_2v_1, u_2v_2 : uv \in \widehat{E}\}$. For every $uv \in \widehat{E}$ and $i, j \in \{1, 2\}$, set $c_{u_iv_j} \equiv \widehat{c}_{uv}$, $w_{u_iv_j} \equiv \widehat{w}_{uv}$. For every $v \in \widehat{V}$, set $p_{v_1} \equiv \widehat{p}_v^+, p_{v_2} \equiv \widehat{p}_v^-$; set $q_{v_1} \equiv 1, q_{v_2} \equiv 0$ if $\widehat{p}_v^- \neq \widehat{p}_v^+$ and $q_{v_1} = q_{v_2} \equiv 0$ otherwise.

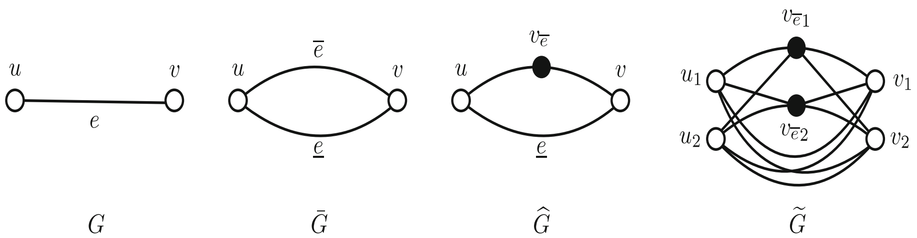


Fig.4. Producing graphs $\overline{G}$, $\widehat{G}$ and $\widetilde{G}$ from G.

The main idea behind the transformation is to make a correspondence between solutions to the MSR_PCST on (G, c^-, c^+, p^-, p^+) of extremeness property and PCSTs in $\widetilde{G}$. Take the instances depicted in Fig.4 as an example. Consider tree $T = G$ together with payment $x_e = c_e^+$ and prizes $y_u = p_u^+$, $y_v = p_v^-$. It corresponds tree $\overline{T}$ in $\overline{G}$ spanned by $\overline{e}$ with cost $\overline{c}_{\overline{e}} = c_e^+$, weight $\overline{w}_{\overline{e}} = 0$ (indicating risk $r(x_e) = 0$ of e) and prizes $y_u = p_u^+$, $y_v = p_v^-$. In turn, $\overline{T}$ together with its associated cost, weight and prizes corresponds to tree $\widehat{T}$ in $\widehat{G}$ going through $\overline{e}_1 = uv_{\overline{e}}$ and $\overline{e}_2 = v_{\overline{e}}v$ of costs $\widehat{c}_{\overline{e}_i} = \overline{c}_{\overline{e}}/2 = c_e^+/2$, weights $\widehat{w}_{\overline{e}_i} = \overline{w}_{\overline{e}}/2 = 0$, $i = 1, 2$, and prizes $y_u = p_u^+$, $y_v = p_v^-$, $y_{v_{\overline{e}}} = 0$. Finally, the terminal correspondence gives tree $\widetilde{T}$ in $\widetilde{G}$ going through $f_1 = u_1v_{\overline{e}_1}, f_2 = v_{\overline{e}_1}v_2$ of costs $c_{f_i} = \widehat{c}_{\overline{e}_i} = \overline{c}_{\overline{e}}/2 = c_e^+/2$, weights $w_{f_i} = \widehat{w}_{\overline{e}_i} = \overline{w}_{\overline{e}}/2 = 0$, $i = 1, 2$, prizes $p_{u_1} = y_u = p_u^+$, $p_{v_2} = y_v = p_v^-$, $p_{v_{\overline{e}_1}} = y_{v_{\overline{e}}} = 0$, and qualities $q_{u_1} = 1$ (indicating risk $r(y_u) = 1$ of u), $q_{v_2} = 0$ (indicating risk $r(y_v) = 0$ of v), $q_{v_{\overline{e}_1}} = 0$. It is clear that $\nu(T, x, y) = c_e^+ - p_u^+ - p_v^- = \nu(\widetilde{T}, c, p)$ and $r_s(T, x, y) = r(x_e) + r(y_u) + r(y_v) = 0 + 1 + 0 = w(E(\widetilde{T})) + q(V(\widetilde{T}))$.

Observe that $\widetilde{G}$ in Fig.4 contains no vertex of degree 2, and thus it is not a 2-tree by Remark 2.1. Generally, we have the following.

Remark 3.1. The graph $\widetilde{G}$ output by ALG_GT is not a 2-tree.

Observe that $|\widehat{V}| = n + |E|$, and every tree in

$$\widetilde{\mathfrak{T}} \equiv \big\{\widetilde{T} : \widetilde{T} \text{ is a tree in } \widetilde{G}, \text{ and } |V(\widetilde{T}) \cap \{v_1, v_2\}| \le 1,\ \forall\, v \in \widehat{V}\big\}.$$

has at most $|\widehat{V}|$ vertices. Since 2-tree G has $|E| = 2n-3$ edges,

$$\text{Every tree } T \in \widetilde{\mathfrak{T}} \text{ contains at most } 3n-3 \text{ vertices.} \tag{15}$$

Note that there is a 1-1 correspondence between the set $\widetilde{\mathfrak{T}}$ and the set of pairs $(\widehat{T}, \widehat{y})$, where $\widehat{T}$ is a tree in $\widehat{G}$ and $\widehat{y} \in \mathbb{R}_+^{V(\widehat{T})}$ with $\widehat{y}_v \in \{p_v^+, p_v^-\}$ for every $v \in V(\widehat{T})$. The bijection "$\text{pair}_1(\widetilde{T}) \equiv (\widehat{T}, \widehat{y})$ if and only if $\text{tree}\,(\widehat{T}, \widehat{y}) \equiv \widetilde{T}$" satisfies the following conditions for every edge $uv \in \overline{E}$:

- $u_1v_1 \in E(\widetilde{T})$ if and only if $uv \in E(\widehat{T})$ and $\widehat{y}_u = \widehat{p}_u^+$, $\widehat{y}_v = \widehat{p}_v^+$;
- $u_1v_2 \in E(\widetilde{T})$ if and only if $uv \in E(\widehat{T})$ and $\widehat{y}_u = \widehat{p}_u^+$, $\widehat{y}_v = \widehat{p}_v^-$;
- $u_2v_1 \in E(\widetilde{T})$ if and only if $uv \in E(\widehat{T})$ and $\widehat{y}_u = \widehat{p}_u^-$, $\widehat{y}_v = \widehat{p}_v^+$;
- $u_2v_2 \in E(\widetilde{T})$ if and only if $uv \in E(\widehat{T})$ and $\widehat{y}_u = \widehat{p}_u^-$, $\widehat{y}_v = \widehat{p}_v^-$.

Lemma 3.5. *If tree* $(\widehat{T}, \widehat{y}) = \widetilde{T}$, *then* $\nu(\widetilde{T}, c, p) = \nu(\widehat{T}, \widehat{c}, \widehat{y})$.

In addition, there is a 1-1 correspondence between the pairs $(\overline{T}, y)$, where $\overline{T}$ is a tree in $\overline{G}$ and $y \in \mathbb{R}_+^{V(\overline{T})}$ with $y_v \in \{p_v^+, p_v^-\}$ for every $v \in V(\overline{T})$, and the pairs $(\widehat{T}, \widehat{y})$, where $\widehat{T} \in \widehat{\mathfrak{T}} \equiv \big\{\widehat{T}' : \widehat{T}' \text{ is a tree in } \widehat{G}, E(\widehat{T}') \cap \{\overline{e}_1, \overline{e}_2,\ \underline{e}\} = \{\underline{e}\} \text{ or } \{\overline{e}_1, \overline{e}_2\} \text{ or } \emptyset,\ \forall\, e \in E\big\}$ and $\widehat{y} \in \mathbb{R}_+^{V(\widehat{T})}$ with $\widehat{y}_v \in \{\widehat{p}_v^+, \widehat{p}_v^-\}$ for every $v \in V(\widehat{T})$. The bijection "$\text{add}\,(\overline{T}, y) \equiv (\widehat{T}, \widehat{y})$ if and only if $\text{con}\,(\widehat{T}, \widehat{y}) \equiv (\overline{T}, y)$" satisfies the following conditions for every edge $e = uv \in E$:

- $\underline{e} \in E(\overline{T})$ if and only if $\underline{e} \in E(\widehat{T})$ and $\widehat{y}_u = y_u$, $\widehat{y}_v = y_v$;
- $\overline{e} \in E(\overline{T})$ if and only if $\{\overline{e}_1, \overline{e}_2\} \subseteq E(\widehat{T})$ and $\widehat{y}_u = y_u$, $\widehat{y}_v = y_v$, $\widehat{y}_{v_{\overline{e}}} = 0$.

Lemma 3.6. *If add* $(\overline{T}, y) = (\widehat{T}, \widehat{y})$, *then* $\nu(\widehat{T}, \widehat{c}, \widehat{y}) = \nu(\overline{T}, \overline{c}, y)$.

Moreover, there is a 1-1 correspondence between the pairs $(\overline{T}, y)$, where $\overline{T}$ is a tree in $\overline{G}$ and $y \in \mathbb{R}_+^{V(\overline{T})}$ with $y_v \in \{p_v^+, p_v^-\}$ for every $v \in V(\overline{T})$, and the triples (T, x, y), where T is a tree in G and $x \in \mathbb{R}_+^{E(T)}$ with $x_e \in \{c_e^+, c_e^-\}$ for every $e \in E(T)$. The bijection " $\text{triple}\,(\overline{T}, y) \equiv (T, x, y)$ if and only if $\text{pair}_2(T, x, y) \equiv (\overline{T}, y)$" satisfies $V(\overline{T}) = V(T)$ and the following conditions for every edge $e \in E$:

- $\underline{e} \in E(\overline{T})$ if and only if $e \in E(T)$ and $x_e = c_e^-$;
- $\overline{e} \in E(\overline{T})$ if and only if $e \in E(T)$ and $x_e = c_e^+$.

Lemma 3.7. *If triple* $(\overline{T}, y) = (T, x, y)$, *then* $\nu(T, x, y) = \nu(\overline{T}, \overline{c}, y)$.

From Lemmas 3.5–3.7, we can establish a 1-1 correspondence between trees $\widetilde{T} \in \widetilde{\mathfrak{T}}$ and triples (T, x, y) such that T is a tree in G, $x \in \mathbb{R}_+^{E(T)}$, $x_e \in \{c_e^+, c_e^-\}$ for every edge $e \in E(T)$, and $y \in \mathbb{R}_+^{V(T)}$, $y_v \in \{p_v^+, p_v^-\}$ for every vertex $v \in V(T)$. The following theorem summarizes the correspondence.

Theorem 3.8. *Let tree* $(T, x, y) \equiv$ *tree(add(* $\text{pair}_2(T, x, y))) = \widetilde{T}$, *i.e., triple* $(\widetilde{T}) \equiv$ *triple(con (pair*$_1(\widetilde{T}))) = (T, x, y)$. *Then* $\nu(T, x, y) = \nu(\widetilde{T}, c, p)$ *and* $r_s(T, x, y) = W(\widetilde{T}, w, q)$, *where* $W(\widetilde{T}, w, q) \equiv w(E(\widetilde{T})) + q(V(\widetilde{T}))$.

For solving the MSR_PCST problem on 2-*tree* $G = (V, E)$, we resort to the Weight Constrained PCST problem (WC_PCST). Given $(\widetilde{G}, c, p, w, q, \zeta)$, where ζ is an integer, and $(\widetilde{G}, c, w, p, q)$ with $c, w \in \mathbb{R}_+^{E(\widetilde{G})}$ and $p, q \in \mathbb{R}_+^{V(\widetilde{G})}$ is the output of ALG_GT on input (G, c^-, c^+, p^-, p^+), the WC_PCST problem consists of finding a tree $T \in \widetilde{\mathfrak{T}}$ with $W(T, w, q) \le \zeta$ and a minimum value with respect to c, p as follows:

$$\text{(WC\_PCST)} \qquad \min_{T \in \widetilde{\mathfrak{T}}, W(T,w,q) \leq \zeta} \big(c(E(T)) - p(V(T))\big).$$

The WC_PCST problem on $\widetilde{G}$ is solved following similar ideas presented in Section 2. With each ordered pair (u, v) corresponding to an edge uv of $\widehat{G}$, we associate $(13\zeta + 13)$ measures to summarize the value incurred so far in the subgraph S of $\widetilde{G}$ which has been reduced onto the edge uv. Specifically, for each $\xi = 0, 1, 2, \cdots, \zeta$, we have the following measures.

- $st_{ij}(u, v, \xi)$ is the minimum among all values of trees T in S with $W(T, w, q) \leq \xi$ and u_i, $v_j \in V(T)$, for $i, j \in \{1, 2\}$.
- $dt_{ij}(u, v, \xi)$ is the minimum among all total values of any two vertex disjoint trees T_1 and T_2 in S with $W(T_1, w, q) + W(T_2, w, q) \leq \xi$ and $u_i \in V(T_1)$ while $v_j \in V(T_2)$, for $i, j \in \{1, 2\}$.
- $un_i(u, v, \xi)$ is the minimum among all values of trees T in S with $W(T, w, q) \leq \xi$ and $u_i \in V(T)$ while $v_1,\ v_2 \notin V(T)$, for $i = 1, 2$.
- $nv_i(u, v, \xi)$ is the minimum among all values of trees T in S with $W(T, w, q) \leq \xi$ and $v_i \in V(T)$ while $u_1,\ u_2 \notin V(T)$, for $i = 1, 2$.
- $nn(u, v, \xi)$ is the minimum among all values of trees T in S with $W(T, w, q) \leq \xi$ and $u_i, v_j \notin V(T)$, for $i, j \in \{1, 2\}$.

Algorithm for Weight Constrained PCST Problem (ALG_WC)

Input $(\widetilde{G}, c, p, w, q, \zeta)$
Output The optimal value ν_ζ and an optimal solution $\widetilde{T}_\zeta \in \widetilde{\mathfrak{T}}$ of the WC_PCST problem

1. Initially, put $F \leftarrow \widehat{G}$ and $\mathsf{L} \leftarrow 2\mathsf{n}(\mathsf{c}^+(\mathsf{E}) + \mathsf{p}^+(\mathsf{V}))$. For every (u, v, ξ) with $uv \in E(G)$, set
   $st_{ij}(u, v, \xi) \leftarrow c_{u_i v_j} - p_{u_i} - p_{v_j}$, if $w_{u_i v_j} + q_{u_i} + q_{v_j} \leq \xi$; L otherwise, $i, j = 1, 2$;
   $dt_{ij}(u, v, \xi) \leftarrow -p_{u_i} - p_{v_j}$, if $q_{u_i} + q_{v_j} \leq \xi$; L otherwise, $i, j = 1, 2$;
   $un_i(u, v, \xi) \leftarrow -p_{u_i}$, if $q_{u_i} \leq \xi$; L otherwise, $i = 1, 2$;
   $nv_i(u, v, \xi) \leftarrow -p_{v_i}$, if $q_{v_i} \leq \xi$; L otherwise, $i = 1, 2$;
   $nn(u, v, \xi) \leftarrow 0$.

2. Update the measures when sequentially deleting degree-2 vertex z in F (its corresponding two degree-4 vertices in $\widetilde{G}$). Let u, v be the neighbors of z in the current graph F. Using measures for (u, v), (u, z) and (z, v) which have been computed, the algorithm updates the measures associated with (u, v, ξ), $\xi = 0, 1, 2, \cdots, \zeta$, for all $i, j \in \{1, 2\}$ as follows:

$$\begin{aligned} st_{ij}(u, v, \xi) \leftarrow \min \big\{ & st_{ij}(u, v, \xi_1) + un_i(u, z, \xi_2) + nv_j(z, v, \xi_3) + p_{u_i} + p_{v_j}, \\ & st_{ij}(u, v, \xi_{3h+1}) + st_{ih}(u, z, \xi_{3h+2}) + dt_{hj}(z, v, \xi_{3h+3}) + p_{u_i} + p_{z_h} + p_{v_j}, \\ & st_{ij}(u, v, \xi_{3h+7}) + dt_{ih}(u, z, \xi_{3h+8}) + st_{hj}(z, v, \xi_{3h+9}) + p_{u_i} + p_{z_h} + p_{v_j}, \\ & dt_{ij}(u, v, \xi_{3h+13}) + st_{ih}(u, z, \xi_{3h+14}) + st_{hj}(z, v, \xi_{3h+15}) + p_{u_i} + p_{z_h} + p_{v_j}, \\ & h = 1, 2 \mid \sum_{h=3k-2}^{3k} \xi_h = \xi,\ 1 \leq k \leq 7;\ \xi_1, \xi_2, \cdots, \xi_{21} \geq 0 \big\}. \end{aligned}$$

$$\begin{aligned} dt_{ij}(u, v, \xi) \leftarrow \min \big\{ & dt_{ij}(u, v, \xi_1) + un_i(u, z, \xi_2) + nv_j(z, v, \xi_3) + p_{u_i} + p_{v_j}, \\ & dt_{ij}(u, v, \xi_{3h+1}) + st_{ih}(u, z, \xi_{3h+2}) + dt_{hj}(z, v, \xi_{3h+3}) + p_{u_i} + p_{z_h} + p_{v_j}, \end{aligned}$$

$$dt_{ij}(u,v,\xi_{3h+7})+dt_{ih}(u,z,\xi_{3h+8})+st_{hj}(z,v,\xi_{3h+9})+p_{u_i}+p_{z_h}+p_{v_j},$$
$$h=1,2 \mid \sum_{h=3k-2}^{3k}\xi_h=\xi, 1\le k\le 5;\ \xi_1,\xi_2,\cdots,\xi_{15}\ge 0\big\}.$$

$$\begin{aligned} un_i(u,v,\xi) \leftarrow \min\big\{&un_i(u,v,\xi_1)+un_i(u,z,\xi_2)+p_{u_i},\\ &un_i(u,v,\xi_{3h})+st_{ih}(u,z,\xi_{3h+1})+zn_h(z,v,\xi_{3h+2})+p_{u_i}+p_{z_h}, h=1,2\\ &\mid \xi_1+\xi_2=\xi_3+\xi_4+\xi_5=\xi_6+\xi_7+\xi_8=\xi,\xi_1,\xi_2,\ldots,\xi_8\ge 0\big\}.\end{aligned}$$

$$\begin{aligned} nv_i(u,v,\xi) \leftarrow \min\big\{&nv_i(u,v,\xi_1)+nv_i(z,v,\xi_2)+p_{v_i},\\ &nv_i(u,v,\xi_{3h})+nz_h(u,z,\xi_{3h+1})+st_{hi}(z,v,\xi_{3h+2})+p_{z_h}+p_{v_i}, h=1,2\\ &\mid \xi_1+\xi_2=\xi_3+\xi_4+\xi_5=\xi_6+\xi_7+\xi_8=\xi,\xi_1,\xi_2,\ldots,\xi_8\ge 0\big\}.\end{aligned}$$

$$\begin{aligned} nn(u,v,\xi) \leftarrow \min\big\{\min\big\{&nz_h(u,z,\xi_{2h-1})+zn_h(z,v,\xi_{2h})+p_{z_h}, h=1,2 \mid \xi_{2h-1},\xi_{2h}\ge 0,\\ &\xi_{2h-1}+\xi_{2h}=\xi, h=1,2\big\}, nn(u,v,\xi), nn(u,z,\xi), nn(z,v,\xi)\big\}.\end{aligned}$$

3. In the end, $\widehat{G}$ is reduced to a single edge uv (corresponding to four edges u_1v_1, u_1v_2, u_2v_1, u_2v_2 in $\widetilde{G}$), where $E(F)=\{uv\}$. Take the minimum, denoted as ν_ζ, among the nine final measures $st_{ij}(u,v,\zeta)$, $un_i(u,v,\zeta)$, $nv_i(u,v,\zeta)$, $nn(u,v,\zeta)$, $i,j\in\{1,2\}$. Output ν_ζ and an optimal solution $\widetilde{T}_\zeta$ (which is a tree in $\widetilde{G}$ corresponding to the minimum measure, and has been constructed during the process of computing measures in a way analogous to ALG_PCST). Note that $\widetilde{T}_\zeta\in\widetilde{\mathfrak{T}}$.

The above algorithm ALG_WC turns out an extension of the dynamic programming algorithm by [11] for solving the constrained SMT problem on 2-trees. The difference is that our algorithm makes additional consideration on vertex prizes (see the formulas following "←" in Steps 1–2 above) for graph $\widetilde{G}$ which is not a 2-tree (recall Remark 3.1), while Chen-Xue algorithm uses simpler formulas to update measures for an input 2-tree without vertex prize parameter.

Theorem 3.9. *Given any 2-tree $G=(V,E)$ of n vertices, Algorithm* ALG_WC *outputs an optimal solution $\widetilde{T}_\zeta$ and the optimum value ν_ζ of the WC_PCST problem on $\widetilde{G}$ in $O(n^3)$ time, where $\widetilde{G}=(\widetilde{V},\widetilde{E})$ is constructed by Algorithm* ALG_GT.

Proof. The correctness of the algorithm can be proved in a similar way to that in the proof of Theorem 2.2. To see the $O(n^3)$ time complexity, by Theorem 3.8 we notice that $W(\widetilde{T},w,q)=r_s(T,x,y)\le 2n-1$ for all $\widetilde{T}\in\widetilde{\mathfrak{T}}$, where $(T,x,y)=\text{TRIPLE}(\widetilde{T})$. So we may assume $\zeta\le 2n-1$. Thus, it suffices to prove that ν_ζ can be computed in $O(|V(\widehat{G})|\zeta^2)$. For this purpose, we only need to show that all measures can be updated in $O(\zeta^2)$ time when a degree-2 vertex is considered and deleted in Step 2. The task is accomplished by applying the technique proposed in the proof of Theorem 3.2 of [11]. For instance, to obtain $st_{ij}(u,v,\xi_1)+un_i(u,z,\xi_2)+nv_j(z,v,\xi_3)$ with $\xi_1+\xi_2+\xi_3=\xi$, for all possible $\xi_1,\xi_2,\xi_3\in\{0,1,\cdots,\xi\}$, we can first compute $LR(\xi_0)=st_{ij}(u,v,\xi_1)+un_i(u,z,\xi_2)$ with $\xi_1+\xi_2=\xi_0$, for all $\xi_0=0,1,\cdots,\xi$ and $\xi_1=0,1,\cdots,\xi_0$; and then compute $LR(\xi_0)+nv_j(z,v,\xi_3)$ with $\xi_0+\xi_3=\xi$, for all $\xi_0=0,1,\cdots,\xi$. □

Now we are ready to present our algorithm for the MSRPCST problem on $(G,c^-,c^+,p^-,p^+,\mathsf{B})$, where G is a 2-tree. In the following pseudo-code, (T,x,y) denotes a solution to the MSR_PCST problem with at most one edge $f\in E(T)$ and one vertex $u\in V(T)$ for which extremeness property (14) holds. Moreover, for the specified edge f (when $E(T)\ne\emptyset$) and vertex u, the risk sum $r_s(T,x,y)$ is minimum among all solutions (T',x',y') of the MSR_PCST problem such that $f\in E(T')$ (when $E(T)\ne\emptyset$), $u\in V(T')$, and (14) holds with T' in place of T. The basic idea makes use of the two equalities in the conclusion of Theorem 3.8, which imply that $\lfloor r_s(T,x,y)\rfloor$ or $\lfloor r_s(T,x,y)\rfloor+1$ is the smallest value of integer ζ that can guarantee the optimal value of the WC_PCST on $(\widetilde{G}, c, p, w, q, \zeta)$ not exceeding B $(\geq \nu(T, x, y))$. Since $\zeta \in [0, 2n-1]$, as argued in the proof of Theorem 3.9, the following algorithm Alg_Msr performs a binary search in Steps 7–11 to determine integer β with $\lfloor r_s(T, x, y) \rfloor \leq \beta \leq \lfloor r_s(T, x, y) \rfloor + 2$ by utilizing Alg_Wc. Then Alg_Msr considers in Steps 12–14 every one-vertex tree T. Subsequently in Steps 15–33, for every $u \in V$ and every $f \in E$, Alg_Msr finds a solution (T, x, y) with minimum risk sum such that $u \in V(T)$, $f \in E(T)$ and (14) holds. To enforce $u \in V(T)$ and $f = ab \in E(T)$, Alg_Msr reassigns u, a, b prize intervals $[\mathcal{M}, \mathcal{M}]$, $[p_a^- + \mathcal{M}, p_a^+ + \mathcal{M}]$, $[p_b^- + \mathcal{M}, p_b^+ + \mathcal{M}]$, respectively, where $\mathcal{M} = c^+(E) + 1$, and assigns f cost interval $[0, 0]$. In turn, for $\zeta = \beta, \beta-1, \beta-2$, Alg_Msr finds optimal trees $\widetilde{T}_\zeta$ of the WC_PCST on $(\widetilde{G}, c, p, w, q, \zeta)$, and modifies x'_f, y'_u in triple $(\widetilde{T_\zeta}) = (T, x', y')$ to make it a best possible solution (T, x, y) of the MSR_PCST problem on $(G, c^-, c^+, p^-, p^+, \mathsf{B})$. Finally, Alg_Msr outputs the (T, x, y) with smallest $r_s(T, x, y)$.

Algorithm for MSR_PCST on 2-tree (Alg_Msr**)**

Input 2-tree $G = (V, E)$ with $c^- \in \mathbb{R}_+^E$, $c^+ \in \mathbb{R}_+^E$, $p^- \in \mathbb{R}_+^V$, $p^+ \in \mathbb{R}_+^V$, and $\mathsf{B} \in \mathbb{R}_+$
Output An optimal solution (T^*, x^*, y^*) of the MSR_PCST problem that satisfies Lemma 3.4.

1. Call Alg_Gt to construct $\widetilde{G} = (\widetilde{V}, \widetilde{E})$, $w \in \{0,1\}^{\widetilde{E}}, q \in \{0,1\}^{\widetilde{V}}$, $c \in \mathbb{R}_+^{\widetilde{E}}, p \in \mathbb{R}_+^{\widetilde{V}}$
2. Call Alg_Wc to find optimal value ν_{2n-1} for WC_PCST on $(\widetilde{G}, c, p, w, q, 2n-1)$
3. **if** $\nu_{2n-1} > \mathsf{B}$ **then** stop (No feasible solution!)
4. Call Alg_Wc to find the optimal value ν_0 for WC_PCST on $(\widetilde{G}, c, p, w, q, 0)$
5. **if** $\nu_0 \leq \mathsf{B}$ **then** output triple $(\widetilde{T_0})$, where $\widetilde{T_0} \in \widetilde{\mathfrak{T}}$, $\nu(\widetilde{T_0}, c, p) = \nu_0$, and stop
6. $\mathfrak{T}^* \leftarrow \emptyset$, $\alpha \leftarrow 0$, $\beta \leftarrow 2n-1$, $\mathcal{M} \leftarrow \sum_{e \in E} c_e^+ + 1$
7. **while** $\beta - \alpha > 1$ **do begin**
8. $\quad$ $\gamma \leftarrow \lfloor (\beta + \alpha)/2 \rfloor$
9. $\quad$ Call Alg_Wc to find optimal value ν_γ for WC_PCST on $(\widetilde{G}, c, p, w, q, \gamma)$
10. $\quad$ **if** $\nu_\gamma \leq \mathsf{B}$ **then** $\beta \leftarrow \gamma$ **else** $\alpha \leftarrow \gamma$
11. **end-while**
12. **for** every $u \in V$ **do begin**
13. $\quad$ **if** $-p_u^+ \leq \mathsf{B}$ **then** $T \leftarrow (\{u\}, \emptyset)$, $y_u \leftarrow \max\{p_u^-, -\mathsf{B}\}$, $\mathfrak{T}^* \leftarrow \mathfrak{T}^* \cup \{(T, \text{NULL}, y)\}$
14. **end-for**
15. **for** every $u \in V$ **do begin**
16. $\quad$ $k^- \leftarrow p_u^-$, $k^+ \leftarrow p_u^+$, $p_u^- \leftarrow \mathcal{M}$, $p_u^+ \leftarrow \mathcal{M}$
17. $\quad$ **for** every $f = ab \in E$ **do begin**
18. $\quad\quad$ $t^- \leftarrow c_f^-$, $t^+ \leftarrow c_f^+$, $t_a^+ \leftarrow p_a^+$, $t_a^- \leftarrow p_a^-$, $t_b^+ \leftarrow p_b^+$, $t_b^- \leftarrow p_b^-$
19. $\quad\quad$ $c_f^- \leftarrow 0$, $c_f^+ \leftarrow 0$, $p_a^+ \leftarrow p_a^+ + \mathcal{M}$, $p_a^- \leftarrow p_a^- + \mathcal{M}$, $p_b^+ \leftarrow p_b^+ + \mathcal{M}$, $p_b^- \leftarrow p_b^- + \mathcal{M}$

20. Call Alg_Gt to construct $\widetilde{G} = (\widetilde{V}, \widetilde{E})$, $c, w \in \mathbb{R}_+^{\widetilde{E}}$, $p, q \in \mathbb{R}_+^{\widetilde{V}}$
21. **for** $i = 0 : 2$ **do begin**
22. Call Alg_Wc to find an optimal solution $\widetilde{T}_{\beta-i} \in \widetilde{\mathfrak{T}}$ for WC_PCST on $(\widetilde{G}, c, p, w, q, \beta - i)$, where the tree in TRIPLE $(\widetilde{T}_{\beta-i})$ contains u, f
23. **end-for**
24. $c_f^- \leftarrow t^-, c_f^+ \leftarrow t^+, \quad p_a^+ \leftarrow t_a^+,\, p_a^- \leftarrow t_a^-,\, p_b^+ \leftarrow t_b^+,\, p_b^- \leftarrow t_b^-$
25. **for** $i = 0 : 2$ **do begin**
26. $(T, x, y) \leftarrow \text{TRIPLE}(\widetilde{T}_{\beta-i})$
27. **if** $\nu(\widetilde{T}_{\beta-i}, c, p) - x_f + y_u + t^- - k^+ \le \mathsf{B}$
28. **then** find an optimal solution (x_f^0, y_u^0) to $\min \frac{t^+ - x_f}{t^+ - t^-} + \frac{y_u - k^-}{k^+ - k^-}$ subject to $x_e - y_v \le \mathsf{B} - \nu(\widetilde{\mathsf{T}}_{\beta-\mathsf{i}}, \mathsf{c}, \mathsf{p}) + \mathsf{x_f} - \mathsf{y_u}$, $t^- \le x_f \le t^+$, $k^- \le y_u \le k^+$
29. $x_f \leftarrow x_f^0$, $y_u \leftarrow y_u^0$, $\quad \mathfrak{T}^* \leftarrow \mathfrak{T}^* \cup \{(T, x, y)\}$
30. **end-for**
31. **end-for**
32. $p_u^- \leftarrow k^-$, $p_u^+ \leftarrow k^+$
33. **end-for**
34. Output $(T^*, x^*, y^*) \in \mathfrak{T}^*$ with minimum $r_s(T^*, x^*, y^*)$

Theorem 3.10. *Given any MSR_PCST instance on a* 2*-tree of* n *vertices, Algorithm* Alg_Msr *outputs its optimal solution in* $O(n^5)$ *time.*

Proof. Since algorithm Alg_Wc runs in $O(n^3)$ by Theorem 3.9, Steps 2–11 take $O(n^3 \log n)$ time, Steps 12–14 take $O(n)$ time and Steps 15–33 take $O(n^5)$ time, the total running time of Alg_Msr is $O(n^5)$.

To prove the correctness, we note that the algorithm returns an optimal solution if it stops before Step 7. If some optimal tree consists of only one vertex, then its corresponding optimal solution is put to $\mathfrak{T}^*$ in Steps 12–14, and the solution output in Step 34 must be optimal. It remains to consider the case where every optimal tree contains at least one edge. Let (T^*, x^*, y^*) be an optimal solution *specified* in Lemma 3.4, where $f \in E(T^*)$ and $u \in V(T^*)$ satisfy (14). Putting $x_f^{**} = c_f^-$, $y_u^{**} = p_u^+$, $x_e^{**} = x_e^*$, for every $e \in E(T^*) \setminus \{f\}$, and $y_v^{**} = y_v^*$, for every $v \in V(T^*) \setminus \{u\}$, we have $r_s(T^*, x^{**}, y^{**}) = r_s(T^*, x^*, y^*) - r(x_f^*) - r(y_u^*) + 2$. Let $\widetilde{T} \equiv \text{tree}\,(T^*, x^{**}, y^{**})$. It follows from Theorem 3.8 that $W(\widetilde{T}, w, q) = r_s(T^*, x^{**}, y^{**})$ and $\nu(\widetilde{T}, w, q) = \nu(T^*, x^{**}, y^{**}) \le \nu(T^*, x^*, y^*) \le \mathsf{B}$. After Step 11, the optimal value $\nu_{\beta-1}$ of the WC_PCST on $(\widetilde{G}, c, p, w, q, \beta - 1)$ is greater than B, and the WC_PCST on $(\widetilde{G}, c, p, w, q, \beta)$ has an optimal tree $\widetilde{T}'$ with $\nu(\widetilde{T}', c, p) = \nu_\beta \le \mathsf{B}$. Therefore $W(\widetilde{T}, w, q) > \beta - 1$ and $r_s(T^*, x^*, y^*) \le W(\widetilde{T}', w, q) \le \beta$, implying

$$r_s(T^* \setminus f \setminus u, x^*, y^*) \equiv r_s(T^*, x^*, y^*) - r(x_f^*) - r(y_u^*) = \beta - \theta, \qquad \text{where } \theta \in \{0, 1, 2\}. \quad (16)$$

Next, we consider the execution of Alg_Msr when Step 15 and Step 17 deal with the specified $u \in V(T^*)$ and $f \in E(T^*)$. Let us first show that in Step 22, we can always find the desired $\widetilde{T}_{\beta-i}$. To avoid confusion, let c', p' denote the c, p on which Step 22 works. Using similar approach to proving Lemma 2.1, we can guarantee $u, a, b \in V(T)$, where $(T, x, y) =$ TRIPLE$(\widetilde{T}_{\beta-i})$, by the optimality of $\widetilde{T}_{\beta-i}$ and the correspondence between $\widetilde{T}_{\beta-i}$ and T. In case of $f \notin E(T)$, since Step 19 associates f with cost interval $[0, 0]$, it follows from the optimality of $\nu(T, x, y) = \nu(\widetilde{T}_{\beta-i}, c', p')$ that there exists a path P in T connecting a and b such that $x_e = 0$ for every $e \in E(P)$. Let T be modified by adding f and deleting one of edges on P, denoted as g. Let x be modified by removing x_g and setting $x_f = 0$. Corresponding modifications apply to $\widetilde{T}_{\beta-i}$ to assure TRIPLE$(\widetilde{T}_{\beta-i})= (T, x, y)$. The validity of Step 22 follows. Moreover, by (16) and the setting in Step 19, the WC_PCST on $(\widetilde{G}, c', p', w, q, \beta-\theta)$ examined by Step 22 has a solution $\widetilde{T}'' =$ TREE(T^*, x', y'), where $x'_f = 0$, $y'_u = \mathcal{M}$, $x'|_{E(T^*)-\{f\}} = x^*|_{E(T^*)-\{f\}}$, $y'|_{V(T^*)-\{u\}} = y^*|_{V(T^*)-\{u\}}$, and $W(\widetilde{T}'', c', p') = r_s(T^*, x', y') = r_s(T^* \backslash f \backslash u, x^*, y^*)$. So this WC_PCST has optimal value

$$\nu(\widetilde{T}_{\beta-\theta}, c', p') \le \nu(T^*, x', y') = \nu(T^*, x^*, y^*) - x^*_f + y^*_u + 0 - \mathcal{M} \le \mathsf{B} - \mathsf{x}^*_\mathsf{f} + \mathsf{y}^*_\mathsf{u} - \mathcal{M}. \quad (17)$$

Furthermore, let us consider the for-loop (Steps 25–30) dealing with $i = \theta$. In Step 26, (T, x, y) is set to TRIPLE$(\widetilde{T}_{\beta-\theta})$. It can be deduced from Step (19) and (17) that $\nu(\widetilde{T}_{\beta-\theta}, c, p) - x_f + y_u \le \mathsf{B} - \mathsf{x}^*_\mathsf{f} + \mathsf{y}^*_\mathsf{u}$, giving

$$\nu(\widetilde{T}_{\beta-\theta}, c, p) - x_f + y_u + x^*_f - y^*_u \le \mathsf{B}. \quad (18)$$

Hence we have $\nu(\widetilde{T}_{\beta-\theta}, c, p) - x_f + x_u + c^-_f - p^+_u \le \mathsf{B}$, which leads the algorithm to Step 28 for finding an optimal solution (x^0_f, y^0_u) of the LP over there. Observe from (18) that (x^*_f, y^*_u) is a feasible solution to the LP, which yields

$$r(x^0_f) + r(y^0_u) \le r(x^*_f) + r(y^*_u).$$

Subsequently, in Step 29, x, y are modified and then (T, x, y) is put to $\mathfrak{T}^*$, for which we have

$$r_s(T, x, y) = r_s(T^* \setminus f \setminus u, x^*, y^*) + r(x^0_f) + r(y^0_u) \le \beta - \theta + r(x^*_f) + r(y^*_u) = r_s(T^*, x^*, y^*),$$
$$\nu(T, x, y) = \nu(\widetilde{T}_{\beta-\theta}, c, p) - x_f + y_u + x^0_f - y^0_u \le \mathsf{B}.$$

So (T, x, y) is an optimal solution to MSR_PCST problem, and ALG_MSR outputs an optimal solution to MSR_PCST problem at Step 34. □

3.3 Discussion

In the preceding subsections we propose two risk models, the MSR_PCST and the MMR_PCST, for the PCST problem with interval data. These two models with the same data setting may yield different solutions as shown by the example depicted in Fig.5.

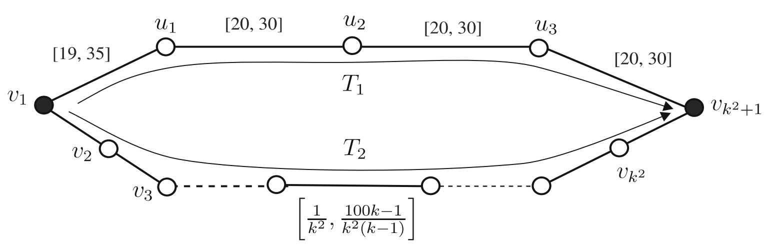


Fig.5. Min-max Risk Model and Min-sum Model are Different.

Both models investigate a cycle on k^2+4 vertices, where $k \geq 8$ is an integer. The cost interval of every edge is specified beside the edge. In particular, $c_e^- = 1/k^2$ and $c_e^+ = (100k-1)/(k^2(k-1))$, for edge $e = v_iv_{i+1},\ \ i = 1, 2, \cdots, k^2$. Every vertex has prize interval $[0, 0]$. The target set consists of two black vertices v_1 and v_{k^2+1}. The tree in any solution to the MSR_PCST problem or to the MMR_PCST problem must be one of the two paths $T_1 = v_1u_1u_2u_3v_{k^2+1}$ and $T_2 = v_1v_2v_3 \cdots v_{k^2+1}$ between v_1 and v_{k^2+1}. Given budget bound $\mathsf{B} = \mathsf{100}$, using Lemmas 3.1 and 3.2, one can easily verify that the optimal solution to the MMR_PCST is $(T_2, \frac{100}{k^2}\mathbf{1}, \mathbf{0})$ with maximum risk $1/k$, noting that every feasible solution $(T_1, x, \mathbf{0})$ must have some $e \in E(T_1)$ with $x_e \leq 25$, giving a risk at least $\min\left\{\frac{25-20}{30-20}, \frac{25-19}{35-19}\right\} = \frac{3}{8} > \frac{1}{k}$. On the other hand, the optimal solution to the MSR_PCST problem is $(T_1, x, \mathbf{0})$ with $x = (20, 30, 30, 20)$ that has risk sum $\frac{17}{16}$, noting that every feasible solution on T_2 has risk sum at least $k \geq 8$. In this example, the optimal solution to the MMR_PCST problem is very inefficient for the MSR_PCST problem in case of large k, and vice versa.

4 Simulation

We simulate the MMR_PCST and the MSR_PCST models in various of network situations to investigate the solution behaviors and average performance of both models. Besides the number N of target vertices and the budget bound B, we introduce the *prize factor* $\rho \equiv \left(\frac{1}{n}\sum\limits_{v\in V}\frac{p_v^- + p_v^+}{2}\right)/\left(\frac{1}{m}\sum\limits_{ij\in E}\frac{c_{ij}^- + c_{ij}^+}{2}\right)$ to balance the costs at edges and the prizes at vertices, which may be using different metrics in applications.

4.1 Methodology

To simulate the networks, we adopt some ideas of [18,23] and use the recursive definition of 2-trees (see Section 2) to randomly generate two different graphs (depicted in Fig.6) with various of parameters. The instances are designed to have a local structure similar to street map instances of fiber optics networks

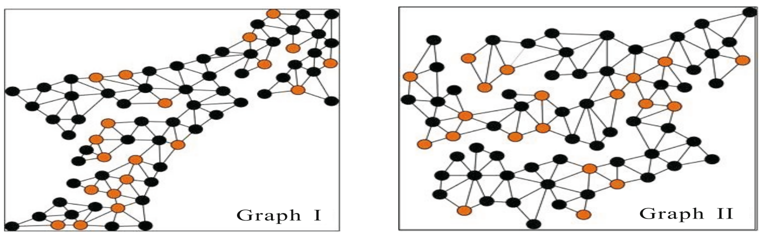


Fig.6. The Two Generated Instances: Graph I and Graph Ⅱ (with 20 Target Vertices in Grey).

Simulation Method

1. Randomly generate two 75-vertex instances on 1×1 Euclidian square as shown in Fig.6. Then randomly select N target vertices with $\mathsf{N} = \mathsf{10}, \mathsf{15}$ and 20, respectively.
2. Randomly generate the cost interval $[c_{ij}^-, c_{ij}^+]$ of edge v_iv_j as follows: $c_{ij}^- = (1-\alpha_{ij})l_{ij}$ and $c_{ij}^+ = (1+\alpha_{ij})l_{ij}$, where l_{ij} is the Euclidian distance between v_i and v_j, which is multiplied by a factor of 10^3 and rounded up to the nearest integer, and α_{ij} is a uniform distributed random value in $[0, 1]$.

3. Randomly generate value d_v for every vertex $v \in V$ following a uniform distribution such that $\frac{1}{n}\sum_{v\in V} d_v = \rho\big(\frac{1}{m}\sum_{\{ij\}\in E}\frac{c_{ij}^- + c_{ij}^+}{2}\big)$, where the *prize factor* ρ takes three different values, 0.5, 1 and 2, respectively. Generate the prize interval of vertex v as follows: $p_v^- = (1-\lambda_v)d_v$ and $p_v^+ = (1+\lambda_v)d_v$, where λ_v is a uniform distributed random value in $[0, 1]$.

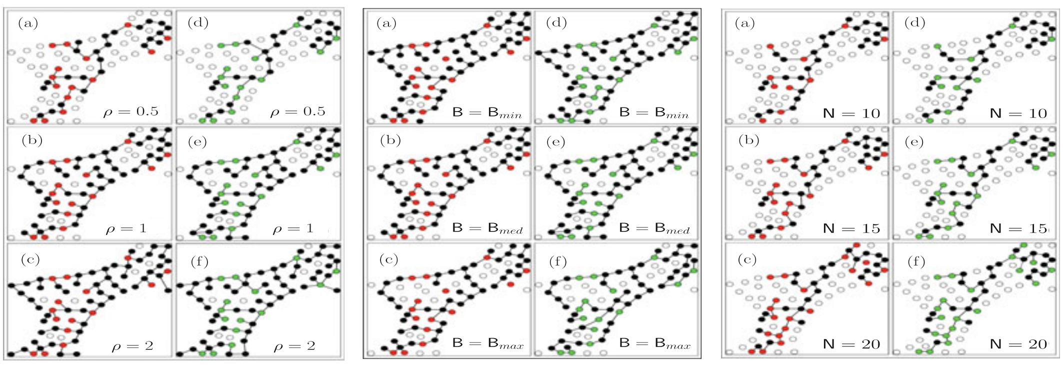
4. For each set of generated cost and prize intervals c^-, $c^+ \in \mathbb{R}_+^E$, p^-, $p^+ \in \mathbb{R}_+^V$, determine the range $[\mathsf{B}_{\min}, \mathsf{B}_{\max}]$ of the budget bound B, as specified at the beginning of Section 3 (see Page 9). Set $\mathsf{B}_{\mathsf{med}} = (\mathsf{B}_{\min} + \mathsf{B}_{\max})/2$.

5. For each generated instance, solve the MMR_PCST and the MSR_PCST problems by using CPLEX as a program platform to implement the algorithms Alg_Mmr and Alg_Msr, respectively.

4.2 Numerical Results and Analysis

We present some of the numerical results and analysis derived from Graph I, while the rest of others are available upon request to the corresponding author.

Fig.7 demonstrates the optimal solutions to the MSR_PCST problem as shown in (a,b,c) and that to the MMR_PCST problem as shown in (d,e,f) with different prize factors ρ, budget bounds B, and target-vertex numbers N. As seen from Fig.7(1), when ρ becomes bigger, the optimal trees of both models grow larger and contain more non-target leaves. This is because more vertices have larger prizes and are more profitable to be included in the trees. Fig.7(2) shows that the optimal trees of both models become smaller as B becomes bigger. The reason behind this phenomenon is that collecting prizes by adding vertices may lead to adding edges with the same amount of payments, so the trees tend to include a small number of edges and vertices. From Fig.7(3) where $\rho = 0.5$ and $\mathsf{B} = \mathsf{B}_{\min}$, we observe that the optimal trees obtained grow larger as N becomes bigger, and almost most of leaves in the trees are target vertices since prizes at vertices are relatively smaller than costs on edges. In addition, there is no significant difference in structures between optimal trees of two models. This is because, in this case, few trees are feasible since the budget bounds are tight.



(1) $\mathsf{B} = \mathsf{B}_{med}$ & $\mathsf{N} = 15$ (2) $\mathsf{N} = 15$ & $\rho = 1$ (3) $\rho = 0.5$ & $\mathsf{B} = \mathsf{B}_{min}$

Fig.7. Optimums Vary with Prize Factor ρ, Budget Bound B and Target-vertex Number N.

Fig.8 demonstrates the optimum of the MSR_PCST problem (on the left) and that of the MMR_PCST problem (on the right) with different parameters ρ, B, for $\mathsf{N} = 15$. In addition to the natural observation that optimums of these two models decrease as B increases, we find the minimum risk sums more sensitive to variation of prize factor ρ.

Fig.9 demonstrates, with prize factor $\rho = 0.5$, target-vertex number $\mathsf{N} = 10$ and five different budget bounds B, the risk sums of the optimal solutions (on the left) and the maximum risks of the optimal solutions (on the right) of both models. The differences under the same metric of risk sum are much less notable than that under the same metric of maximum risk. This could be even more clearly observed in Fig.10 which shows the absolute and relative deviations of the optimums of the two models, respectively. In particular, under the metric of risk sum the relative deviation is no greater than 0.75; under the metric of maximum risk, however, the relative deviation could be as high as 3. In other words, optimal solutions to MMR_PCST problem appear to be good solutions in terms of risk sum objective.

Following the above analysis, we conclude that the model of MMR_PCST has better performance than the model of MSR_PCST in average. In addition to that, it takes only $O(n^2)$ time to obtain an optimal solution to MMR_PCST problem in contrast to $O(n^5)$ time for MSR_PCST problem.

5 Conclusion

In this paper, we have proposed two risk models for the PCST problem with interval data and solved them to optimality for series-parallel graphs. Compared with other models for uncertain optimization problems, their superiorities lie in not only keeping computational complexity unchanged but also providing flexibility for decision makers. So far risk models have been proved successful in dealing with several polynomial-time solvable problems. In the future, it is worthwhile studying how to apply risk models to some other optimization problems with interval data, particularly NP-hard optimization problems. Moreover, it is interesting to see if our models and approaches can be extended to the problems where uncertainty is not described using intervals.

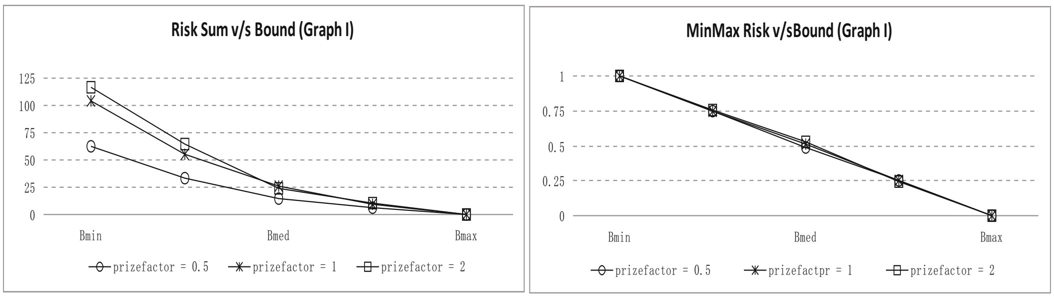


Fig.8. Optimums Vary with the Budget Bound B and Prize Factor ρ, where Target-vertex Number $\mathsf{N} = 15$.

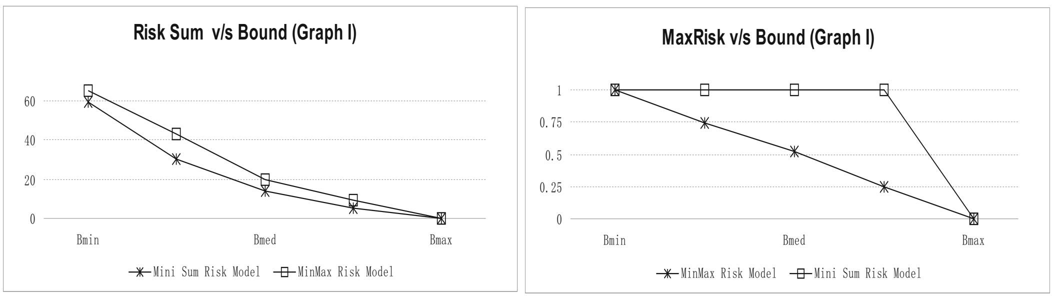


Fig.9. Risk Sum (Left) and Maximum Risk (Right) Vary with Budget Bound B, where $\rho = 0.5$ and $\mathsf{N} = 10$.

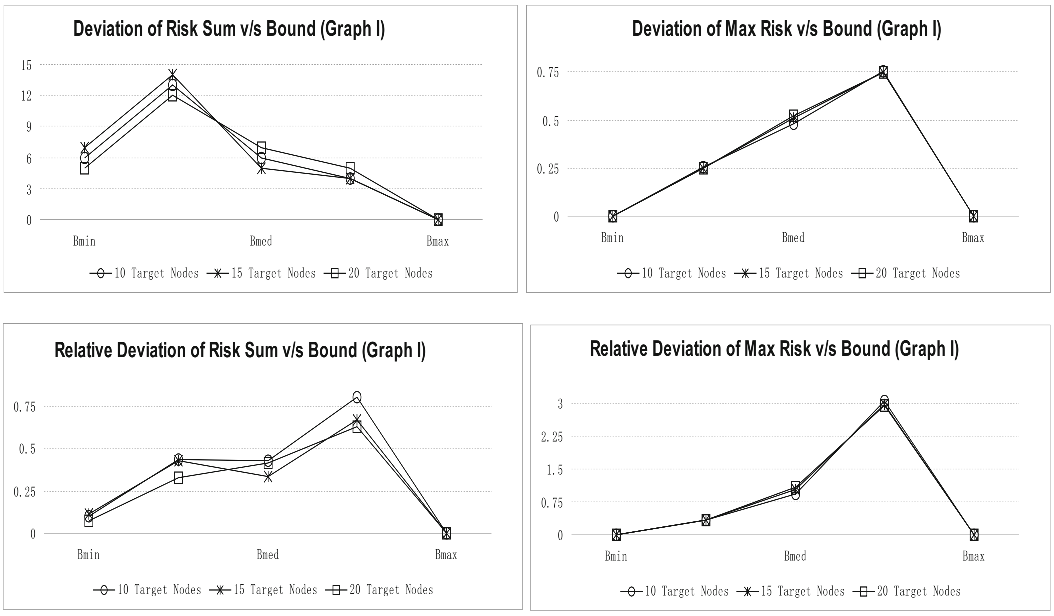


Fig.10. (Relative) Deviations of the Risk Sum v/s Bound B (Left), the Maxrisk v/s Bound B (Right), where $\rho = 0.5$.

Acknowledgements. The authors are indebted to two anonymous referees for their invaluable comments and suggestions which have greatly improved the presentation of this paper.

References

[1] Aissi, H., Bazgan, C., Vanderpooten, D. Min-max (regret) versions of cut problems. *Lecture Notes in Computer Science*, 3827: 789–798 (2005)

[2] Álvarez-Miranda, E., Candia, A., Chen, X., Hu, X., Li, B. Efficient algorithms for the prize collecting Steiner tree problems with interval data. *Lecture Notes in Computer Science*, 6124: 13–24 (2010)

[3] Aron, I.D., Hentenryck, P.V. On the complexity of the robust spanning tree problem with interval data. *Operations Research Letters*, 32: 36–40 (2004)

[4] Averbakh, I., Lebedev, V. Interval data min-max regret network optimization problems. *Discrete Applied Mathematics*, 138: 289–301 (2004)

[5] Bailly-Bechet, M., Borgs, C., Braunstein, A., Chayes, J., Dagkessamanskaia, A., Franois, J.-M., Zecchina, R. Finding undetected protein associations in cell signaling by belief propagation. *Proceedings of the National Academy of Sciences*, 108: 882–887 (2011)

[6] Bailly-Bechet, M., Braunstein, A., Zecchina, R. A prize-collecting steiner tree approach for transduction network inference. *Lecture Notes in Computer Science*, 5688: 83–95 (2009)

[7] Beltran, H.F., Skorin-Kapov, D. On minimum cost isolated failure immune networks. *Telecommunication Systems*, 3: 183–200 (1994)

[8] Bertsimas, D., Sim, M. Robust discrete optimization and network flows. *Mathematical Programming Series B*, 98: 49–71 (2003)

[9] Bienstock, D., Goemans, M. X., Simchi-Levi, D., Williamson, D. A note on the prize collecting traveling salesman problem. *Mathematical Programming* 59: 413–420 (1993)

[10] Chanas, S., Zieliński, P. The computational complexity of the criticality problems in a network with interval activity times. *European Journal of Operational Research*, 136: 541–550 (2001)

[11] Chen, G., Xue, G. A PTAS for weight constrained Steiner trees in series parallel graphs. *Theoretical Computer Science*, 304: 237–247 (2003)

[12] Chen, X., Hu, J., Hu, X. A polynomial solvable minimum risk spanning tree problem with interval data. *European Journal Operational Research*, 198: 43–46 (2009)

[13] Dittrich, M., Klau, G., Rosenwald, A., Dandekar, T., Muller, T. Indentifying functional modules in protein-protein interaction networks: an integrated exact approach. *ISMB Conference Proceedings*, 24: 223–231

(2008)

[14] Feofiloff, P., Fernandes, C.G., Ferreira, C.E., Pina, J.C. Primal-dual approximation algorithms for the prize collecting Steiner tree problem. *Information Processing Letters*, 103: 195–202 (2007)

[15] Grötschel, M., Monma, C.L., Stoer, M. Computational results with a cutting plane algorithm for designing communication networks with low-connectivity constraints. *Operations Research*, 40: 309–330 (1992)

[16] Hu, J. Minimizing maximum risk for fair network connection with interval data. *Acta Mathematicae Applicatae Sinica (English Series)*, 26: 33–40 (2010)

[17] Hwang, F.K., Richards, D.S., Winter, P. The Steiner minimum tree problems. North-Holland, Amsterdam, 1992

[18] Johnson, D., Minkoff, M., Phillips, S. The prize collecting Steiner tree problem: theory and practice. In: Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms, 1999, 760–769

[19] Karp, R.M. Reducibility among combinatorial problems. In: Complexity of Computer Computations, eds. by R.E. Miller and J.W. Tatcher, Plenum, New York, 1972

[20] Kasperski, A., Zieliński P. The robust shortest path problem in series-parallel multidigraphs with interval data. *Operations Research Letters*, 34: 69–76 (2006)

[21] Kasperski, A., Zieliński P. Minmax regret approach and optimality evaluation in combinatorial optimization problems with interval and fuzzy weights. *European Journal of Operational Research*, 200: 680–687 (2010)

[22] Ljubić, I., Weiskircher, R., Pferschy, U., Klau, G., Mutzel, P., Fischetti, M. An algorithmic framework for the exact solution of the prize-collecting steiner tree problem. *Mathematical Programming B*, 105: 427–449 (2006)

[23] Lucena, A., Resende, M.G.C. Strong lower bounds for the prize collecting Steiner tree problem in graphs. *Discrete Applied Mathematics*, 141: 277–294 (2004)

[24] Megiddo, N. Combinatorial optimization with rational objective functions. *Mathematics of Operations Research*, 4: 414–424 (1979)

[25] Monma, C.L., Shallcross, D.F. Methods for designing communications networks with certain two-connected survivability constraints. *Operations Research*, 37: 531–541 (1989)

[26] Panos, K., Gang, Y. Robust discrete optimization and its applications. 1st ed., Kluwer Academic Publishers, Dordrecht, The Netherlands, 1997

[27] Prodon, A., DeNegre, S., Liebling, T.M. Locating leak detecting sensors in a water distribution network by solving prize-collecting steiner arborescence problems. *Mathematical Programming B*, 124: 119–141 (2010)

[28] Raghavan, S. Low-connectivity network design on series-parallel graphs. *Networks*, 43: 163–176 (2004)

[29] Ruzika, S., Sperber, S., Steiner, M. Earliest arrival flows on series-parallel graphs. *Networks*, 57: 169–173 (2011)

[30] Wald, J.A., Colbourn, C.J. Steiner trees, partial 2-trees, and minimum IFI networks. *Networks*, 13: 159–167 (1983)

[31] Wang, J.-B., Ng, C.T., Cheng, T.C.E. Single-machine scheduling with deteriorating jobs under a series-parallel graph constraint. *Computers & Operations Research*, 35: 2684–2693 (2008)

[32] Yen, C.-C., Lee, R.C.T. A linear time algorithm to solve the weighted perfect domination problem in series-parallel graphs. *European Journal Operational Research*, 73: 192–198 (1994)

[33] Zielinski, P. The computational complexity of the relative robust shortest path problem with interval data. *European Journal Operational Research*, 158: 570–576 (2004)