Received: 20 October 2021 | Revised: 5 April 2022 | Accepted: 2 August 2022

DOI: 10.1002/net.22124

RESEARCH ARTICLE

WILEY

The resource constrained clustered shortest path tree problem: Mathematical formulation and Branch&Price solution algorithm

Daniele Ferone[1] | **Paola Festa**[1] | **Serena Fugaro**[2] | **Tommaso Pastore**[3]

[1]Department of Mathematics and Applications, University of Naples "Federico II", Naples, Italy
[2]Institute for Applications of Calculus "Mauro Picone", National Research Council of Italy, Rome, Italy
[3]Department of Structures for Engineering and Architecture, University of Naples "Federico II", Naples, Italy

Correspondence
Daniele Ferone, Department of Mathematics and Applications, University of Naples "Federico II", Naples, Italy.
Email: daniele.ferone@unina.it

Abstract

In this article, the Resource Constrained Clustered Shortest Path Tree Problem is defined. It generalizes the classic Resource Constrained Shortest Path Tree Problem since it is defined on an undirected, complete and weighted graph whose set of nodes is partitioned into clusters. The aim is then to find a shortest path tree respecting some resource consumption constraints and inducing a connected subgraph within each cluster. The main support and motivation for studying this problem are related, among the others, to the design of telecommunication networks, and to Disaster Operations Management. In this work, we present a path-based formulation for the problem, addressing the case of local resource constraints, that is, resource constraints on single paths. For its resolution, a Branch&Price algorithm featuring a Column Generation approach with Multiple Pricing Scheme is devised. A comprehensive computational study is conducted, comparing the proposed method with the results achieved by the CPLEX solver, adopted to solve the mathematical model. The numerical results underline that the Branch&Price algorithm outperforms CPLEX, both in terms of solution cost and time.

KEYWORDS

Branch&Price, clustered graph, column generation, Danzig Wolfe, resource constrained paths

1 | INTRODUCTION

In the scientific literature the *clustered*—or *generalized*—versions of several classic combinatorial optimization problems have received a great share of attention, due to their wide range of application [22]. Indeed, in a network problem, the switch from the classic to the generalized version is obtained by partitioning the set of nodes of the underlying graph into a specific number K of subsets, referred to as *clusters*. The introduction of these substructures in the modeling phase provides an efficient representation of social [19] as well as territorial dynamics, thus allowing specific real-world problems to be mathematically formulated. This is the case for the design of backbone networks in telecommunication or in metropolitan areas, the optimization of irrigation systems, and the design of inter-cluster topology in computer and transportation networks [48, 62, 66] which have all been addressed through clustered variants of well-known optimization problems.

Moreover, the class of clustered problems is attracting interest also from a theoretical standpoint. In fact, the generalized version of a problem could become extremely harder to solve [12], since the feasibility conditions entail constraints on clusters in addition to those of the original, that is, not clustered, problem.

Notably, the concept of generalization has been carried out in the formulation by following two different approaches: (i) either it is required that each feasible solution contains exactly/at least/at most one node from each cluster [22, 48], (ii) or that the subgraphs induced by the solution within each cluster are connected [9, 66]. Exploiting a generalization of type (ii), [12] defined the *Clustered Shortest Path Tree Problem* (CluSPTP) as the problem of determining a minimum cost shortest path tree on a clustered graph such that the subgraphs induced within each cluster are connected, and proved that this problem is NP-hard [19]. Given its inherent complexity and its applicability in the context of optimized communication networks and irrigation systems, the CluSPTP has been addressed with different heuristic approaches. [4] and [62] designed two Evolutionary Algorithms based on different encoding functions, [60] proposed a heuristic based on the combination of the randomized greedy method and shortest path tree algorithm. Finally, [10] devised an ad hoc Genetic Algorithm, featuring the use of a compact representation scheme, that allows the exploration of the entire solutions space, the use of efficient mutation and crossover operators that do not generate invalid offspring, and the use of a hybrid initial population; indeed this algorithm outperformed the Genetic Algorithm proposed by the same authors in [9].

At the same time, often in transport and telecommunications, in order for the networks considered to be realistic, additional attributes are assigned to the various links in combination with costs [18]. These attributes can denote for example, flying or travel time, fuel consumption, trajectory length, signal delay. Therefore in the design of realistic optimized routes it is required to find optimal paths—w.r.t. cost—that also account for side constraints related to these attributes.

In this spirit, problems such as the Resource Constrained Shortest Path Tree Problem (RC-SPTP) aims to determine a minimum cost shortest path tree that respects restrictions related to a set of available resources [24]. In particular, according to the features of the related application, these restrictions can be either imposed on every path or on the whole tree in the solution; consequently, the corresponding mathematical models feature either local or global resource constraints, respectively [50].

Indeed, the resources can depict heterogeneous attributes, thus several real-world problems are modeled and solved through RC-SPTPs (the interested reader is referred to [37] and references therein). Notably, oftentimes the Clu-SPTPs arising from real-world scenarios inherently entail also decision-making processes related to available resources. However, to the best of our knowledge, this aspect has not been emphasized in any work proposed so far in the related literature.

With these considerations in mind, in the present article we introduce the *Resource Constrained Clustered Shortest Path Tree Problem* (RC-CluSPTP) with local resource constraints. Specifically, given an undirected, complete and weighted graph for which both a partition into clusters for the set of nodes, and a resource function are assigned, the aim is to determine a shortest path tree rooted in a given node, such that: (1) the subgraph induced by this tree within each cluster is connected, and (2) each path from the root in the solution tree respects specific resource consumption constraints.

In the light of the aforementioned supporting motivations, it is quite clear that the optimization problem here described lies at the intersection between the CluSPTP and the RC-SPTP. Accordingly, the possible contexts of application for the RC-CluSPTP naturally comprise, but are not limited to, those of the classic problems. For instance, RC-SPPs are the subproblems deriving from the application of Column Generation approaches for the resolution of Vehicle Routing Problems (VRPs) [36]. Consequently, the resolution of Clustered VRPs [49] would benefit from the investigation of the RC-CluSPTP in that a Column Generation subproblem could include also the cluster-related constraints.

However, with a rationale similar to that with which the RC-CluSPTP has been derived from its counterpart without resources (the Clu-SPTP), one could think of applying this problem for Disaster Operations Management (DOM). Specifically, DOM refers to the set of measures adopted in Humanitarian Logistics to help those most at risk avoiding, limiting, and responding to the consequences of a disaster, both natural and man-made [5]. Indeed, a crucial subset of these measures deals with the *short-term post-disaster* planning operations [7], for example, relief distribution from medical centers to disaster sites, as well as transportation of casualties from affected sites to medical centers. At this purpose, clusters may denote affected areas or their regional partition, while resources can naturally denote available vehicles and commodities. However, the short-term post-disaster operations comprise also the evacuation of displaced people; in this regard, there exist scenarios characterized by cumulative resources representing, for example, the time of evacuation from shelters. In these cases, maximum resource consumption may represent the total congestion-related evacuation time [57]. Additionally, in the related literature, clustering methods have been adopted to classify the areas of the affected region according to distance from/travel time to safer zones or probability that they might be endangered as the disaster progresses [65]. Nonetheless, as thoroughly surveyed in [7] only the aspects related to resource-allocation have been taken into account in the related literature. However, ensuring the rapid supply of goods and the efficient recovery of casualties are equally important concerns in DOM [53]. Consequently, we maintain that well-suited variants of the RC-CluSPTP could be beneficial in the field of Humanitarian Logistics.

The main contributions of the present article are:

1. the definition of a novel problem, the RC-CluSPTP, on a clustered weighted graph equipped with a resource function;
2. the formal characterization of the RC-CluSPTP with Mixed Integer Linear Programming models obtained with an *edge-based formulation*;

3. the implementation of a Branch&Price solution algorithm, which features a Column Generation approach related to a *Dantzig-Wolfe (DW) decomposition* [11] based on the Resource-Constrained Shortest Path Decomposition;
4. a thorough computational experimentation conducted on benchmark instances adapted from the literature, with the aim of determining the extent to which the performance of the solution approach is influenced by the characteristics of the instances.

In particular, with regard to the second contribution it is worth mentioning that in the literature, up to our knowledge, no mathematical formulation has yet been proposed for the CluSPTP. Therefore, the model proposed in this paper could easily be adapted to describe also this problem.

The paper is organized as follows. Section 2 provides a brief literature review on both clustered problems and Resource Constrained Shortest Path Problems. Section 3 presents the path-based mathematical formulation of the RC-CluSPTP. A thorough description of the proposed solution approach, along with a Dantzig-Wolfe decomposition and the corresponding Pricing Problem formulation, are given in Section 4. Then, the obtained computational results are presented and analyzed in Section 5. Finally, Section 6 draws conclusions and identifies future lines of research.

2 | LITERATURE REVIEW

As pointed out in Section 1, the RC-CluSPTP can be seen either as a refinement of the CluSPTP or as a generalized version of the classic RC-SPTP. Therefore, in this section we provide: (i) a brief literature review of generalized combinatorial optimization problems, focusing on both widely studied and recently proposed ones (Section 2.1); (ii) an overview of Resource Constrained Shortest Path Problems (Section 2.2).

2.1 | Brief overview of generalized combinatorial optimization problems

The class of clustered problems comprises generalizations of well-established problems such as: the *Generalized Traveling Salesman Problem* (GTSP) [3, 26, 27, 58, 59], the *Generalized Minimum Spanning Tree Problem* (GMSTP) [19, 21, 25, 33, 35, 45, 46, 48], and the *Generalized Steiner Tree Problem* (GSteTP) [34, 51, 66, 67]. The GTSP is one of the earliest formulated clustered problem [59]; in particular, in the scientific literature there have appeared contributions dealing with both mentioned concepts of generalization (see Section 1). Just to cite a few, [30] addressed the problem of finding a minimum cost Hamiltonian tour so that the nodes of each cluster are visited consecutively. They proposed approximation algorithms for different variants of the problem, depending on whether or not the starting and ending nodes of a cluster have been specified. In particular, the approximation ratio of their algorithm for this latter variant of GTSP was later improved by [3]. Instead, [26] proposed Integer Linear Programs for the Symmetric GTSPs in which the least cost solution cycle has to visit each cluster at least/exactly once. The same authors tackled these variants with a Branch&Cut algorithm [27]. More recently, [58] addressed the version of the Symmetric GTSP with exactly one visit for each cluster by implementing a heuristic based on adaptive large neighborhood search. However, it is worth noting that, like the classic Traveling Salesman Problem, all its generalized versions are NP-hard problems.

The GMSTP was first formulated by [45] as the problem of finding a minimum cost tree spanning a subset of nodes with exactly one node from each cluster (EGMSTP); later, [35] defined the version in which at least one node from each cluster has to be included in the spanning tree, namely LGMSTP. However, in contrast with the classic counterpart which is polynomially solvable, both these variants have been proved to be NP-hard. However, [19] pointed out that the GMSTP in which the solution tree is required to induce a connected subgraph in each cluster remains polynomially solvable. In the literature, (meta-)heuristic and exact approaches have been proposed for the above mentioned GMSTPs: for example, [46] devised a Tabu Search algorithm for EGMSTP and adapted it to the resolution of LGMSTP; [33] tackled the EGMSTP with a Variable Neighborhood Search heuristic combining three different neighborhood types; finally, [25] found that, independent of the construction heuristic, a GRASP algorithm for EGMSTP performed the best when employing path relinking and Iterated Local Search. As regards the exact approaches, [21] designed a Branch&Cut algorithm for EGMSTP; instead, [48] proposed a Dynamic Programming algorithm for EGMSTP.

As regards the GSteTP, [51] defined a first version in which, supposing that the subset R of *required nodes* is divided in K clusters, the aim is to find a minimum cost tree of the graph containing at least one node from each cluster. [34] tackled this GSteTP with a $(K-1)$-approximation algorithm, while [67] computed a lower bound on the problem using Lagrangean relaxation and subgradient optimization. Exploiting the idea of generalization in terms of cluster connection, [66] defined a variant of GSteTP on metric graphs: this problem aims at finding a minimum cost Steiner tree inducing mutually disjoint minimal spanning trees in the clusters of R. The authors tackled this problem with a $(2+\rho)$-approximation algorithm, where ρ

is the best known approximation ratio for the Minimum Steiner Tree Problem (MSteTP). Trivially, these GSteTPs are NP-hard since they generalize an NP-hard problem; however, unlike the classic MSteTP, they remain computationally hard problems even when the input graph is a tree or there are no Steiner nodes [67].

As thoroughly described in [22], among the other classic problems whose generalization has been formulated with the exactly/at least/at most approach, there are: the *Vehicle Routing Problem* [29], the *Minimum Clique Problem* [38], the *Shortest Path Tree Problem* [39].

To conclude, we mention the *Minimum Routing Cost Clustered Tree Problem* in which the generalization is formulated in terms of cluster connection. [40] recently proposed this problem which aims at finding a clustered spanning tree with minimum routing cost, defined as the total distance summed over all pairs of vertices. The authors proved the NP-hardness of the problem, when there are more than two clusters, and proposed an approximation algorithm.

2.2 | Brief overview of resource constrained shortest path problems

The Resource Constrained Shortest Path Problem (RC-SPP) was formulated by [13] as a subproblem of a Bus Driver Scheduling problem; since then, the investigation of RC-SPPs has represented a flourishing field of research. Indeed, it is worth mentioning that the RC-SPP is also referred to as the Weight Constrained Shortest Path Problem when arcs have associated scalars, rather than vectors of weights. [18] proved the usefulness of exploiting upper bounds and cost information on preprocessing operations in both exact and heuristic solution approach.

However, in the scientific literature, different variants of the problem have been addressed for two main reasons: the resource constraints are versatile, thus they allow one to depict a heterogeneous variety of real-world problems, and the formulation of these constraints depends itself on the nature of the attributes represented through the resources [37].

For this purpose, [2] classified these attributes into three categories and presented the different formulations of the corresponding constraints. According to them, resources can represent attributes that are: *numerical* and *cumulative* (e.g., travel time, fuel consumption); *numerical* and *noncumulative* (e.g., road width, height); *indexed* or *categorical* (e.g., parking restrictions, type of roads).

Moreover, the NP-hardness of the RC-SPP regardless of the type and the number of resources [24] motivates the continuous development of new solution approaches, both exact and heuristic.

Just to mention a few contributions, [56] devised an algorithm which encompasses the binary partition strategy adopted in the construction of the first k shortest paths ($\mathcal{K}$-SP Path Ranking algorithm) [47] and several pruning strategies, obtaining a sensible speedup on large instances, compared to the state of the art. [52] improved the classic Dynamic Programming approach for RC-SPP [15] through bidirectional search with resource-based bounding; moreover, the authors adopted the Dynamic Programming paradigm to compute lower bounds in a Branch&Bound algorithm and proposed the well-performing *Decremental State-Space Relaxation*, in which the two previously mentioned algorithms are special cases. Instead, [55] proposed an exact algorithm which exploits the resource constraints to define a geometric search direction guiding efficiently the $\mathcal{K}$-SP algorithm [31] in the resolution of the RC-SPP. Indeed, this approach outperforms both the basic $\mathcal{K}$-SP and the version which solves a Lagrangian relaxation of the problem. Then, [32] generalized and extended the valid inequality devised by [28] for a Branch&Bound approach to solve RC-SPP. Moreover, a heuristic procedure to obtain an initial feasible solution at each node of the branching tree and a variable fixing technique are presented in this paper. Finally, recent papers propose exact approaches combining the bidirectional search procedure with heuristics. Namely, [63] adopted the A* strategy, devising an algorithm which exploits the resource constraints to outperform the state-of-the-art in the resolution of large instances. Instead, [6] equipped the bidirectional search with a *pulse-based* heuristic which turns out to find near optimal solutions quickly. In particular, [41] were the first to propose an exact search algorithm for RC-SPP based on the idea of propagating a *pulse* through the graph, that is, a partial path storing relevant information about the solution being explored.

Among the proposed heuristic approaches, it is worthwhile mentioning [17], who devised an Ant Colony algorithm to solve a particular RC-SPP arising from a Multimodal Transport Problem featuring time windows and constraints on the overall number of transshipments. Also, to solve the RC-SPP, [43] proposed a hybridized version of a Particle Swarm Optimization method featuring a Variable Neighborhood Search and two different Expanding Neighborhood topologies. Specifically, different local search strategies were tested in their computational experiments to detect the most effective one.

Finally, it is well known that RC-SPP often appears as a Pricing Problem in Column Generation, for example, for the Vehicle Routing Problem [1] and the Crew Scheduling Problem [8]; consequently, its resolution has been further refined in order to optimize this type of procedure. Notably, [20], discussed how the solution of the Elementary SPP with resource constraints can be leveraged in column generation schemes designed for VRPs. To this end, the authors adapted the label-correcting algorithm proposed by [14] showing that their approach successfully solves the Elementary RC-SPP, and can be used to achieve effective lower bounds for a wide variety of routing problems. Similarly, studying a scenario in which RC-SPP arises as a subproblem,

[68] devised a pseudo-polynomial three-stage solution approach which performs the first two stages, namely *pre-processing* and *setup*, only once to transform RC-SPP into a Shortest Path Problem; then, the last stage solves this problem at each iteration of Column Generation.

3 | PROBLEM DESCRIPTION

Let $G = (V, E)$ be a simple undirected and complete graph, and let $c : E \to \mathbb{R}^+$ be a non-negative cost function, with $c(i,j) \overset{def}{=} c_{ij}$. Moreover, let $r : E \to \mathbb{R}^+$ define a *resource consumption* function, where $r(i,j) \overset{def}{=} r_{ij}$ and the resources are supposed to represent *numerical cumulative attributes* [2]. For any node-set $S \subseteq V$, the *subgraph induced* by S is the graph $(S, E(S))$ such that $\forall i,j \in S$, the corresponding edge $[i,j]$ is in $E(S)$ if $[i,j] \in E$, that is, $E(S) \subseteq E$.

Given a source node $s \in V$, consider a collection $\{C_k\}_{k=1,\dots,K}$ of node-sets, referred to as *clusters*, that partition V; namely, the intersection of C_k and C_j is empty for all $k \neq j$, $(k,j \leq K)$ and $\cup_{k \leq K} C_k = V$. Then, the *Resource Constrained Clustered Shortest Path Tree Problem* (RC-CluSPTP) aims to find a minimum-cost tree T_s in G rooted at s, such that:

(i) T_s spans all the nodes of the graph G,
(ii) for each cluster C_k the subgraph induced by T_s is connected,
(iii) every path in T_s respects a maximum resource consumption constraint.

In particular, the set E can be partitioned in two subsets: the set of edges connecting nodes within the same cluster, called *intra-cluster edges*, and the set of edges connecting nodes belonging to two distinct clusters, that is, the *inter-cluster edges*. In other words, given $[i,j] \in E$, it is an intra-cluster edge if both i and j belong to the same cluster C_k; otherwise, if $i \in C_k$ and $j \in C_h$ with $k \neq h$, then $[i,j]$ is an inter-cluster edge.

In the following, we will denote with E_k the set of intra-cluster edges of cluster C_k, $k = 1, \dots, K$. Finally, let n be the cardinality of the node-set V and n_k be the cardinality of the cluster C_k, $k = 1, \dots, K$; clearly, it holds that $\sum_{k \leq K} n_k = n$.

Figure 1A shows an example instance with 7 nodes and 3 clusters: though this graph is complete, we only show a subset of the edges with their relative costs, while the remaining ones—having prohibitively high costs equal to 10—are omitted. We assume that: the source is node 1, r_{ij} is equal to 1 for each edge, and the maximum resource consumption is equal to 3 for each $i \in \{2, 3, 4\}$, and to 2 for each $i \in \{5, 6, 7\}$. Figure 1B reports the Shortest Path Tree which has a cost equal to 17: note that the cluster C_2 does not induce a connected subgraph in the tree. The Clustered SPT, with cost equal to 19, is reported in Figure 1C: note that the resource consumption constraints for nodes 7 and 6 are not respected since 3 and 4 resources are consumed in them, respectively. Finally, Figure 1D depicts the Resource Constrained Clustered SPT, with cost equal to 29.

3.1 | Edge-based formulation

The RC-CluSPTP can be formulated as a *multi-commodity flow problem*. For this purpose, we will use the directed graph $D = (V, A)$ underlying G, in which $A = \{(i,j), (j,i) : [i,j] \in E\}$; that is, for each edge $[i,j] \in E$, we define both the arc (i,j) and the corresponding anti-parallel arc (j,i) in the arc-set of the graph D. Moreover, D is weighted extending by symmetry the cost function defined on G, that is, $c : A \to \mathbb{R}^+$ is such that $c_{ij} = c_{ji}$. Then, the path-based formulation relies on the definition of two kind of variables:

1. non-negative continuous flow variables f_{hij}, defined $\forall h \in V \setminus \{s\}$ and $\forall (i,j) \in A$, which assume a positive value if the arc (i,j) is used in the path from s to h;
2. binary edge variables x_{ij} defined $\forall [i,j] \in E$, such that $x_{ij} = 1$ if $[i,j] \in T_s$.

Then, to describe the particular tree structure of a generic feasible solution (conditions (i)-(ii)), it is necessary to state:

- the *activation* constraints (1) which couple the f variables with the corresponding x ones:

$$f_{hij} + f_{hji} \leq x_{ij}, \quad \forall [i,j] \in E, h \in V \setminus \{s\}, \tag{1}$$

- the *flow balancing* constraints given in (2), where $FS(i) = \{j : (i,j) \in A\}$ and $BS(i) = \{j : (j,i) \in A\}$:

$$\sum_{j \in FS(i)} f_{hij} - \sum_{j \in BS(i)} f_{hji} = \begin{cases} 1 & \text{if } i = s, \\ -1 & \text{if } i = h, \quad \forall i,h \in V,\ h \neq s, \\ 0, & \text{otherwise,} \end{cases} \tag{2}$$

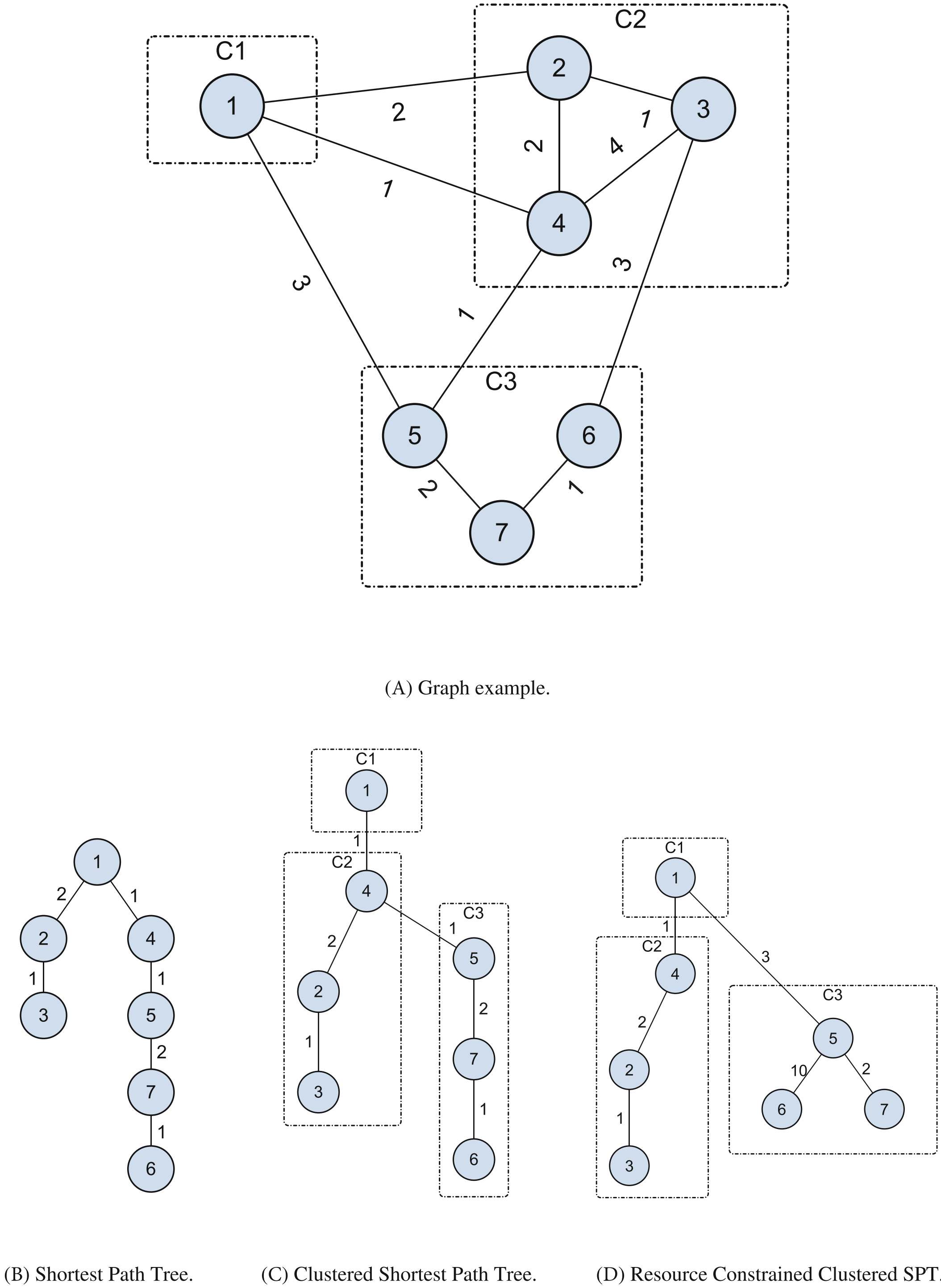

FIGURE 1 Example instance with three corresponding shortest path trees

- the *dimension* constraints, for the whole solution (3a) as well as for each cluster C_k (3b):

$$\sum_{[i,j]\in E} x_{ij} = n - 1, \tag{3a}$$

$$\sum_{[i,j]\in E_k} x_{ij} = n_k - 1, \quad \forall k = 1, \dots, K. \tag{3b}$$

In particular, constraints (3a)-(3b) yield that each feasible solution should contain $n - K$ intra-cluster edges and $K - 1$ inter-cluster edges, connecting clusters so that there is no isolated cluster, otherwise constraints (2) would be violated. Moreover, condition (ii) for a generic solution holds because of the combination of constraints (2) and (3b). Conversely, constraints (2) and (3b) are necessary conditions for (ii); in fact, if a feasible solution would exist in which (2) or (3b) are violated, then at least one cluster should contain a cycle. However, since any feasible solution contains $n-1$ edges, there would be at least one isolated node within a cluster, thus violating condition (ii).

Remark 1. It is worth noting that, for any feasible solution, condition (ii) could be guaranteed replacing constraints (3b) with the *subtour elimination* constraints (4) for each cluster:

$$\sum_{[i,j]\in E(S)} x_{ij} \leq |S|-1, \quad \forall S \subseteq C_k, |S| \geq 2. \tag{4}$$

In fact, in each feasible solution T'_s, condition (4) has to hold as an equality when $S = C_k$, for each $k = 1, \dots, K$ otherwise there would exist a cluster with an isolated vertex. Consequently, also constraints (4) state that each feasible solution should contain $n-K$ intra-cluster edges.

Finally, the resource constraints can be added to each single path, that is, the single commodity flows, by extending the resource function over A as done for the cost function c (cf. (5f)).

The model for the RC-CluSPTP is given in (5), where the objective function (5a) minimizes the cost of the resulting tree.

$$(\text{RC-CluSPTP}) \quad \min \sum_{h\in V\setminus\{s\}} \sum_{(i,j)\in A} c_{ij} f_{hij} \tag{5a}$$

subject to

$$f_{hij} + f_{hji} \leq x_{ij}, \qquad \forall [i,j] \in E, h \in V \setminus \{s\}, \tag{5b}$$

$$\sum_{j\in FS(i)} f_{hij} - \sum_{j\in BS(i)} f_{hji} = \begin{cases} 1 & \text{if } i = s, \\ -1 & \text{if } i = h, \\ 0, & \text{otherwise,} \end{cases} \qquad \forall i, h \in V,\ h \neq s, \tag{5c}$$

$$\sum_{[i,j]\in E} x_{ij} = n-1, \tag{5d}$$

$$\sum_{[i,j]\in E_k} x_{ij} = n_k - 1, \qquad \forall k = 1, \dots, K, \tag{5e}$$

$$\sum_{(i,j)\in A} r_{ij} f_{hij} \leq R_h, \qquad \forall h \in V \setminus \{s\}, \tag{5f}$$

$$x_{ij} \in \{0,1\}, \qquad \forall [i,j] \in E, \tag{5g}$$

$$f_{hij} \geq 0, \qquad \forall (i,j) \in A, h \in V \setminus \{s\}. \tag{5h}$$

Remark 2. It is widely known that the classic RC-SPTP is computationally intractable, namely it is an NP-hard problem even in case of a single resource function [24]. As regards RC-CluSPTP, it is NP-hard too since otherwise, letting the number of clusters be $K = 1$, the RC-SPTP would be tractable.

4 | BRANCH&PRICE AND COLUMN GENERATION

The idea of tackling RC-CluSPTP with a Branch&Price approach comes from the observation that the mathematical model (5) presents a *block diagonal form*. Specifically, for each node $h \in V \setminus \{s\}$, the constraints (5c), (5f), and (5h) describe the feasible region of a Resource Constrained Shortest Path Problem from s to h in D, while constraints (5b) are the *linking constraints*.

Thus, it is possible to apply a *Dantzig-Wolfe (DW) decomposition* [11] to the model (5) to solve the resulting problem with Branch&Price. Indeed, this approach entails that Column Generation (CG) is adopted to solve the linear relaxations in each node of the search tree, involving a *Multiple Pricing* scheme.

The DW decomposition of (5), the resulting Pricing Problems and the strategy adopted to solve them are described in Sections 4.1–4.3.

4.1 | Dantzig-Wolfe decomposition

The DW decomposition of the model (5) follows from the observation that, for each node $h \in V \setminus \{s\}$, the constraints (5c), (5f) and (5h) describe the feasible region of a Resource Constrained Shortest Path Problem from s to h in D. Thus, it is possible to decompose the variables f_{hij} using variables representing the paths from s to h in D which respect the resource constraints.

Formally, given a node $h \in V \setminus \{s\}$, let P_h be the set of all the possible paths in D from s to h which respect the corresponding resource constraint (5f), and define a continuous non-negative variable λ_h^p for every path $p \in P_h$. Then, according to Minkowski's Theorem, the flow along each arc $(i,j) \in D$ in a path from s to h can be expressed as a convex combination of the extreme points of P_h, with these being all the possible paths from s to h in P_h (see (6)):

$$f_{hij} = \sum_{p \in P_h} \lambda_h^p f_{hij}^p. \tag{6}$$

Thus, for each node $h \in V \setminus \{s\}$, the cost of the path from s to h in the solution is a convex combination of the costs c_h^p of the paths $p \in P_h$ (see (7)).

$$\sum_{(i,j)\in A} c_{ij} \sum_{p \in P_h} \lambda_h^p f_{hij}^p = \sum_{p \in P_h} \lambda_h^p \underbrace{\sum_{(i,j)\in A} c_{ij} f_{hij}^p}_{c_h^p} = \sum_{p \in P_h} c_h^p \lambda_h^p. \tag{7}$$

Then, the DW decomposition (8) of (5) is obtained by: replacing the flow variables with the equation (6), and reformulating constraints (5b) and the objective function (5a) with (7). The resulting problem is the so-called *Main Problem*.

$$\text{(DW-MP)} \quad \min \sum_{h \in V\setminus\{s\}} \sum_{p \in P_h} c_h^p \lambda_h^p \tag{8a}$$

subject to

$$\sum_{p \in P_h} \left(f_{hij}^p + f_{hji}^p\right) \lambda_h^p \le x_{ij}, \qquad \forall [i,j] \in E, \forall h \in V \setminus \{s\}, \tag{8b}$$

$$\sum_{[i,j]\in E} x_{ij} = n - 1, \tag{8c}$$

$$\sum_{[i,j]\in E_k} x_{ij} = n_k - 1, \qquad \forall k = 1, \dots, K, \tag{8d}$$

$$\sum_{p \in P_h} \lambda_h^p = 1, \qquad \forall h \in V \setminus \{s\}, \tag{8e}$$

$$x_{ij} \in \{0, 1\}, \qquad \forall [i,j] \in E, \tag{8f}$$

$$\lambda_h^p \ge 0, \qquad \forall p \in P_h, \forall h \in V \setminus \{s\}. \tag{8g}$$

The objective function (8a) minimizes the cost of the resulting tree; (8b) represent the activation constraints while constraints (8c) and (8d) are analogous to the corresponding (5d) and (5e). Finally, (8e) are the *convexity* constraints for the continuous non-negative path variables.

4.2 | Pricing problem formulation

As previously mentioned, each node of the branching tree is solved by applying a Column Generation with a Multiple Pricing Scheme. Specifically, this means that at each iteration of the CG, it is required to solve $n-1$ Pricing Problems, one for each node $h \in V \setminus \{s\}$ in order to find the columns to add to the *Restricted Main Problem* (RMP), which is obtained by considering a subset of path variables. Moreover, as proposed in [64], we consider a linear relaxation of DW-MP, removing the binary constraints (8f).

In order to obtain the mathematical formulation of the Pricing Problem corresponding to a generic node $h \in V \setminus \{s\}$, we explicitly state the dual formulation of DW-MP in (9).

$$\text{(Dual-DW)} \quad \max \left[\sum_{h \ne s} \eta_0^h + (n-1)\mu_2 + \sum_{i \le K} (n_i - 1)\nu_i\right] \tag{9a}$$

subject to

$$\sum_{[i,j]\in E} (f_{hij}^p + f_{hji}^p)\pi_{hij} + \eta_0^h \le c_h^p, \qquad \forall p \in P_h, \forall h \in V \setminus \{s\}, \tag{9b}$$

$$\pi_{hij} \le 0, \qquad \forall [i,j] \in E, \forall h \in V \setminus \{s\}. \tag{9c}$$

In particular, for each edge $[i,j] \in E$, the negative dual variables π_{hij} are associated to the corresponding constraints (8b); then, variable $\eta_0^h \in \mathbb{R}$ corresponds to the convexity constraints (8e). Finally, variables $\nu_i \in \mathbb{R}$ are related to constraints (8d) and variable $\mu_2 \in \mathbb{R}$ corresponds to constraint (8c).

The expression of the reduced costs featured in the Pricing Problem to solve for each node $h \in V \setminus \{s\}$ is obtained as detailed in the following (cf. [42]). Firstly, recall the mathematical formulation (5): let us denote with $\bar{c}$ the vector of costs, with A_h the matrix of coefficients associated with left-hand side of constraints (5b) for any fixed $h \in V \setminus \{s\}$, and with $\bar{f}_h$ the vector of flow variables relative to this node. Finally, consider the subvector $\left(\bar{\pi}_h, \eta_0^h\right)$ relative to an optimal dual solution; the reduced costs c_h featured in the Pricing Problem, are obtained as:

$$c_h = \left(c^T - (\bar{\pi}_h)^T A_h\right)\bar{f}_h - \eta_0^h.$$

The resulting formulation of the Pricing Problem to solve for each node $h \in V \setminus \{s\}$ is given in (10). It is duly noted that this is a Resource Constrained Shortest Path Problem from s to h in D, with non-negative costs.

$$\text{(PP\_h)} \quad \min \left[\left(\sum_{(i,j)\in A} (c_{ij} - \pi_{hij})\ f_{hij}\right) - \eta_0^h\right] \tag{10a}$$

subject to

$$\sum_{j\in FS(i)} f_{hij} - \sum_{j\in BS(i)} f_{hji} = \begin{cases} 1 & \text{if } i = s, \\ -1 & \text{if } i = h, \\ 0, & \text{otherwise.} \end{cases} \qquad \forall i \in V, \tag{10b}$$

$$\sum_{(i,j)\in A} r_{ij} f_{hij} \leq R_h, \tag{10c}$$

$$f_{hij} \geq 0, \qquad \forall (i,j) \in A. \tag{10d}$$

4.3 | Solving the pricing problem

As noted in Section 4.2, the block diagonal structure of the RC-CluSPTP leads to $n-1$ Pricing Problems, each of which consists in the solution of a Resource Constrained Shortest Path Problem from s to $h \in V \setminus \{s\}$ in D. At this purpose, we resort to a Dynamic Programming (DP) algorithm, given its successful application in the solution of constrained shortest path problems [16, 23, 52].

According to this solution paradigm, the possible paths starting from s are associated to labels, recording both their cost and resource consumption. Doing so, the DP explores the solution space evaluating path labels via the concepts of *feasibility* and *dominance*.

In this context, denoted with $R(p)$ the total resource consumption along a generic path p, given two paths p_{si} and $\hat{p}_{si}$ from s to $i \in V \setminus \{s\}$, path p_{si} is said to be *dominated* by $\hat{p}_{si}$ if the following conditions hold: (i) $c(\hat{p}_{si}) \leq c(p_{si})$; (ii) $R(\hat{p}_{si}) \leq R(p_{si})$, and at least one of such conditions is strict. Consequently, throughout the exploration process of DP, only feasible nondominated paths can be held as candidate for extensions toward the target node h.

Let $L_i = (c_i, r_i, p_{si})$ be a label associated to p_{si}, where $c_i = c(p_{si})$, and r_i is the total resource consumption of p_{si}. Let $D(i)$ define the set of all the labels L_i associated with the different paths connecting s to i, and denote with $\langle p_{si}, [i,j]\rangle$ the concatenation between the path p_{si} and edge $[i,j] \in E$.

Algorithm 1 explicitly reports the operations that define the DP approach herein adopted. Firstly, the shortest paths from node h to each node $i \in V \setminus \{h\}$ are computed to prune labels that do not lead to optimal solutions. The next steps performed—Lines from 3 to 5—consist of the initialization of the first label, the list of labels L, the lists $D(j)$, $\forall\, j \in V$, and the incumbent cost Λ.

While L is not empty, DP extracts a feasible nondominated label $L_i \in L$ (Line 7). If the total objective function value related to L_i improves Λ, either the incumbent is updated—when i coincides with the target h – or the procedure generates and adds to L the labels of each node $j \in V$ such that $[i,j] \in E$ (Line 15). As detailed in Algorithm 2, a certain label L_j is added to L and $D(j)$ only if it is feasible and nondominated. The final output of DP is an optimal resource-constrained shortest path. While several different strategies are available in the scientific literature to carry out Line 7, in this implementation we use an A* approach as label-extraction policy, for its balance between simplicity and good performance. Therefore, we implemented L as a binary heap and the comparison value of a label L_i is equal to $c_i + d_i$, where d_i is the minimum distance from i to h.

4.4 | Columns initialization and branching strategy

Two of the main components of a column generation approach are column initialization and branching strategy. The former allows to build an initial set of columns to yield to first RMP, while the latter is meant to partition the solution space in smaller regions.

Function DP$(V, A, c, R, s, h, \pi, n_0^h)$
  Compute the minimum distance d_i from each node $i \in V$ to h
  $L_s \leftarrow (0, \emptyset, \langle s \rangle)$
  $L \leftarrow \{L_s\}$; $D(s) \leftarrow \{L_s\}$; $D(j) \leftarrow \emptyset, \forall j \in V, j \neq s$
  $best \leftarrow$ **Nil**; $\Lambda \leftarrow +\infty$
  while $L \neq \emptyset$ **do**
    $L_i \leftarrow$ Extract(L); $L \leftarrow L \setminus \{L_i\}$
    if $c_i + d_i < \Lambda$ **then**
      if $i = h$ **then**
        $\Lambda = c_i$
        $best \leftarrow L_i$
      else
        foreach $j \in V : (i,j) \in A$ **do**
          $L_j \leftarrow (c_i + (c_{ij} - \pi_{hij})), r_i + r_{ij}, \langle p_{si}, (i,j) \rangle)$
          AddLabel (L, D, L_j)
  $c_{best} \leftarrow c_{best} - n_0^h$
  return *best*

Algorithm 1: Pseudo-code of the dynamic programming algorithm to solve the resource constrained shortest path from s to h

Procedure AddLabel (L, D, L_j)
  if L_j *is feasible* **then**
    if L_j *is not dominated by any label* L'_j *belonging to* $D(j)$ **then**
      Remove from $D(j)$ and L all labels L'_j that are dominated by L_j
      $L \leftarrow L \cup \{L_j\}$
      $D(j) \leftarrow D(j) \cup \{L_j\}$

Algorithm 2: Pseudo-code of label-adding procedure

In our approach, the following strategy is adopted to obtain the starting columns. For each cluster C_k, let $s_k \in C_k$ be such that

$$s_k = \arg\min_{i \in C_k} \left(r_{\hat{p}_{si}} + r_{\hat{T}_i} \right), \tag{11}$$

where $\hat{p}_{si}$ and $\hat{T}_i$ are respectively the minimum path from s to i and the shortest path tree from i to the other nodes of C_k, both obtained using r as cost function. The starting columns are related to the paths of $\bigcup_{k=1}^{K} \left(\hat{p}_{ss_k} \cup \hat{T}_{s_k} \right)$. Moreover, for each $i \in C_k \setminus \{s_k\}$, we include a dummy column with a high cost to ensure at any time feasibility with respect to reachability constraints.

For what concerns the branching strategy, this work considers three distinct branching rules: *random*, *most fractional*, and *Ryan-Foster*.

The core idea of the first two rules, is to generate branches starting from an edge that is partially included in the solution of the RMP. Namely, whenever the set $B = \{[i,j] \in E : 0 < x_{ij} < 1\}$ is not empty, that is, a noninteger solution of the RMP has been found, an edge in B is selected and two corresponding branches are obtained. The former forces the edge to be included in the solution; conversely, in the latter branch this edge is not included in any solution.

The selection of the edge $[i,j]$ is performed uniformly at random in the *random branching*, and by choosing the edge $[i,j]$ corresponding to the variable x_{ij} closer to 0.5 in the *most fractional branching rule* (ties are broken at random).

In the branch where x_{ij} must not be selected, we remove all the columns that use the edge $[i,j]$ and the constraint $x_{ij} = 0$ is added to the RMP. Conversely, in the other branch, the constraint $x_{ij} = 1$ is inserted in the model of the RMP.

Differently, the starting assumption of the *Ryan-Foster rule* [54] is that whenever the solution of the Main problem is fractional, there exist two edges e_1 and e_2 such that the sum of the λ_h values of the paths sending flows through both e_1 and e_2 is fractional. Therefore, the Ryan-Foster rule generates two branches, the first one imposes that both edges are used by the same path, while the second requires that the paths cannot traverse both edges.

5 | COMPUTATIONAL RESULTS

This Section presents and discusses the computational experiments we have designed to appraise the performance of the proposed Branch&Price algorithm (BP), and its comparison to the results achieved by the direct solution of the mathematical model (5) performed with the ILOG CPLEX 12.9 solver. More specifically, Section 5.1 presents the data-set used in our numerical experiments, Section 5.3 discusses the direct comparison of the proposed methodology and the CPLEX solver, while Section 5.4 delves into the analysis of the results of BP when related to instance properties.

The approaches here compared were coded in C++ using the flags `-std=c++17 -O3` and compiled with `g++ 8.2`. The experiments were run on the computer cluster of the SCoPE supercomputing center at the University of Naples "Federico II". A time limit of 5400 s has been used for all solution methods.

5.1 | Instance generation

The network topologies featured in the computational experiments of the present Section are the small instances of Types 1, 5, and 6 used by [61] for the Clustered Shortest Path Tree Problem.

In particular, as described in [44]: Type 1 networks are obtained from the TSPLIB benchmark, using a k-means algorithm to define the clusters; Type 5 instances derive from those used in the literature of the TSP, clustered by grouping the vertices in geometric centers; and, lastly, graphs of Type 6 are adapted from the TSPLIB library, by partitioning the networks into quadrilaterals, each of which corresponds to a cluster. Using these topologies, a set of feasible instances for the RC-CluSPTP is generated through the procedure detailed in the next paragraphs.

The aim of this generating procedure is to build a broadly applicable data-set for resource constrained clustered problems, without assuming specific information regarding the nature of the resources therein modeled. Therefore, in the following procedure, we emphasize only two properties: randomness and the triangle inequality.

Given a generic clustered undirected graph $G = (V, E)$, a resource consumption function $r : E \rightarrow \mathbb{R}^{+}$ satisfying the triangle inequality can be associated to the edges of G according to the following steps:

1. randomly assign temporary resources $r_{tmp,e}$ to each $e \in E$;
2. solve an all-pairs shortest path tree problem using $r_{tmp,e}$ as costs;
3. impose r_{ij} equal to the cost of the shortest path from i to j found in step 2.

Once the non-negative resource consumption function $r : E \rightarrow \mathbb{R}^{+}$ is determined, a RC-CluSPTP instance is specified through its resource constraints.

Let T_s^* be the optimal solution of the classic CluSPTP with r as cost function, and let $p_{s,h}^*$ be the path of T_s^* connecting the source s and the node $h \in V \setminus \{s\}$. Moreover, let $R(p_{s,h}^*)$ indicate its cost in T_s^*, that is, the total resource consumption of $p_{s,h}^*$. Then, using a generation scheme similar the one described in [18], it is possible to define three subclasses characterized by: *low-resource* (Low-R), *medium-resource* (Medium-R), and *high-resource* (High-R), by considering convex combinations of $R(p_{s,h}^*)$ and 2 $R(p_{s,h}^*)$.

$$\text{(Low-R)} \qquad R_h = (1-\alpha)R(p_{s,h}^*) + \alpha 2\, R(p_{s,h}^*), \qquad \forall h \in V \setminus \{s\}, \tag{12}$$

$$\text{(Medium-R)} \qquad R_h = 0.5\, R(p_{s,h}^*) + R(p_{s,h}^*), \qquad \forall h \in V \setminus \{s\}, \tag{13}$$

$$\text{(High-R)} \qquad R_h = \alpha R(p_{s,h}^*) + (1-\alpha)2\, R(p_{s,h}^*). \qquad \forall h \in V \setminus \{s\}. \tag{14}$$

As [18], we used $\alpha = 0.05$ for all graph topologies. It is duly noted how in all three cases T_s^* is a feasible (possibly suboptimal) solution. As a result of this instance generation process we obtained 255 feasible instances for the RC-CluSPTP, whose characterizing features are summarized in Table 1.[1]

5.2 | Analyzing branching rules

Branching rules represent one of the main components of tree-based approaches, as they determine how the subproblems are generated and the exploration criterion of the branching tree. As reported in Section 4.4, this work considers the three following branching rules: *random*, *most fractional*, and *Ryan-Foster*.

Table 2 reports the results obtained on all the instances presenting more than a single branching node. For each rule, the results are organized presenting the averages for the three instance types (Type 1, 5, and 6) and global averages (All). Considering the total computation time, we can observe that most fractional and Ryan-Foster are generally better than the random branching

[1]The full data-set can be found at https://doi.org/10.6084/m9.figshare.14484480.

TABLE 1 Summary of instances properties

Type	Number	Nodes	Edges	Clusters
1	81	51–105	1275–5460	5–75
5	63	30–120	435–7140	5–10
6	111	51–105	1275–5460	2–42

TABLE 2 Comparison of branching rules

Type	Branching rule	Cost	Total time	Branching nodes	Total pricing time	Pricing time per node
Type_1	Random	137 646.72	1968.08	70.71	17.95	2.35
	Most fractional	137 646.72	1526.82	23.29	14.14	3.30
	Ryan-Foster	137 646.72	1667.41	5.29	20.98	4.82
Type_5	Random	38 610.19	1549.43	1206.00	47.64	0.70
	Most fractional	38 610.10	1694.89	1243.33	41.54	0.80
	Ryan-Foster	38 610.10	718.99	16.67	16.54	0.73
Type_6	Random	80 071.85	2813.36	355.44	35.86	1.22
	Most fractional	80 056.12	2502.31	350.33	31.63	1.50
	Ryan-Foster	800 55.95	2799.85	58.78	64.55	3.22
All	Random	87 083.40	2199.70	496.82	33.38	1.44
	Most fractional	87 076.94	1971.72	489.82	28.77	1.88
	Ryan-Foster	87 076.87	1872.02	30.27	37.59	3.05

rule. Moreover, while the number of branching nodes is significantly lower for Ryan-Foster, this criterion often results in harder pricing problems, as shown by the Pricing time per node.

Ultimately, given the slight time advantage that Ryan-Foster presents on the global averages over the most fractional branching rule, in the following comparison with CPLEX, we only consider this branching strategy.

5.3 | Branch&Price versus CPLEX

The first set of computational analyses is devoted to the comparison of BP with the performance achieved by CPLEX on the entire data-set.

Table 3 summarizes the comparative numerical analysis, averaging the results according to three distinct categories: instances solved by both algorithms ("avg both"); instances solved by the corresponding method—that is, all the instances solved by BP (resp. CPLEX) in the corresponding column—("avg single"); and all instances ("avg all"). For each of such categories, Table 3 reports the average objective function value ("Cost"), the total time and time to best, and the optimality gap, computed as

$$Gap = \frac{\hat{C} - LB}{LB}, \tag{15}$$

where $\hat{C}$ is the cost of the incumbent solution, and LB is the best lower bound found by the algorithm. Since BP and CPLEX derive lower bounds using different strategies, different solutions may yield the same optimality gap. Finally, the last two rows of Table 3 indicate, for each method, the number of instances for which at least a feasible solution has been found, and the number of instances solved to optimality[2].

Observing the results of Table 3 it emerges that the proposed Branch&Price found at least feasible solutions for approximately 88% of the instances (225/255); in comparison CPLEX was able to find solutions to 69% of the data-set (177/255). Moreover, when taking into account the number of optimal solutions found, the edge of the proposed algorithm with respect to CPLEX is evident, with a total of 217 instances solved to optimality against 150.

The difference in terms of the number of optimal solutions found is also reflected in the average solution cost in the category *avg both*, that evidences how BP achieves objective function values lower than those obtained by CPLEX.

This first analysis of the quality characterizing the solutions achieved by both approaches evidences that the proposed Branch&Price significantly improves the performance of CPLEX. However, this behavior is not highlighted by the Gap values,

[2]The full set of results can be found in the Supplementary Material of the article.

TABLE 3 BP versus CPLEX on the entire data-set

	BP				CPLEX			
	Cost	Total time	Gap	Time-to-best	Cost	Total time	Gap	Time-to-best
avg both	128 231.08	424.96	0.00	338.95	128 767.78	1886.97	0.04	1260.64
avg single	136 200.33	694.05	0.01	601.46	129 115.19	1986.44	0.05	1278.91
avg all		1247.69				3030.58		
Feasible	225/255				177/255			
Optimal	217/255				150/255			

TABLE 4 Average times and cost values

	Low-R		Medium-R		High-R		Avg	
Type	Cost	Time	Cost	Time	Cost	Time	Cost	Time
Type_1	152 065.46	73.99	153 341.91	765.76	161 223.54	971.80	154 827.34	539.26
Type_5	63 391.99	45.44	62 600.26	324.42	60 891.83	1889.10	62 294.69	752.99
Type_6	170 951.92	35.98	175 124.80	870.23	168 971.03	1815.07	172 074.49	764.12
Average	138 379.06	50.39	139 324.53	696.92	128 948.19	1602.06	136 200.33	694.05

Note: Instances grouped according to network types and resource allowance.

TABLE 5 Fraction feasible/optimal solution found

	Low-R		Medium-R		High-R		Total	
	Feasible	Optimal	Feasible	Optimal	Feasible	Optimal	Feasible	Optimal
Type_1	1	1	0.85	0.81	0.63	0.59	0.83	0.80
Type_5	1	1	1.00	1.00	1.00	0.90	1.00	0.97
Type_6	1	1	0.97	0.95	0.59	0.51	0.86	0.82
Total	1	1	0.94	0.92	0.71	0.64	0.88	0.85

Note: Instances grouped according to network types and resource allowance.

that on average are approximately the same for both algorithms; actually, this outcome can be related to the different strategies used to derive lower bounds.

Concerning the computation times, BP employs approximately 22% of the time needed by CPLEX on instances solved by both methods (424.96 s against 1886.97 s) and around 41% on all instances (1247.69 versus 3030.58 s).

5.4 | Sensitivity analysis

In the second set of analyses we evaluate the performance of the proposed Branch& Price relating the results to the type and distinctive features of the instances considered.

Table 4 reports the averages of the total times of computations and objective function values achieved, dividing the benchmark according to network types (Type 1, Type 5, and Type 6) and resource allowance (namely, Low-R, Medium-R, High-R). Analogously, Table 5 for each group reports the fraction of instances for which at least a feasible solution has been found, and the instances that were solved to optimality. Lastly, Table 6 collects a detailed description of the behavior of the proposed solution approach, reporting, for each group, the time spent solving reduced RMP ("MP time") and pricing subproblems ("PR time"), the number of total RMPs solved ("#MP") and call to the pricing procedures ("#PR"), the number of columns added ("#Cols"), and the total number of branching nodes explored ("#Bnodes").

Analyzing the summaries of Tables 4 and 5, it is evident how a lower resource allowance is related to reduced computational times and thus a higher number of optimal solutions found. This particular behavior can be justified observing that a lower resource allowance leads to a smaller solution space, that allows BP to prove optimality by just exploring an extremely limited number of branching nodes. Table 6 confirms this evidence, reporting that, for the totality of instances characterized by a low resource allowance, a single branching node is sufficient to prove optimality. As a direct consequence, this behavior is also reflected in lower number of calls to the pricing subproblems and the number of columns added to the RMPs.

TABLE 6 Detailed behavior of the proposed Branch&Price approach

Type	Resource	Time	MP time	#MP	PR time	#PR	#Cols	#Bnodes
Type_1	Low-R	73.99	67.95 (91.84%)	45.04	2.21 (2.99%)	3800.74	930.59	1.00
	Medium-R	1452.40	1375.45 (94.70%)	354.81	25.05 (1.73%)	32685.52	3761.07	18.70
	High-R	2612.45	2385.44 (91.31%)	1543.78	41.94 (1.61%)	111 822.52	11 571.44	127.37
Type_5	Low-R	45.44	41.11 (90.46%)	42.71	2.12 (4.67%)	3046.10	745.67	1.00
	Medium-R	324.42	314.93 (97.07%)	118.43	4.29 (1.32%)	8806.33	2625.19	1.00
	High-R	1889.10	1777.65 (94.10%)	1438.33	44.17 (2.34%)	89 491.05	16 520.71	10.24
Type_6	Low-R	35.98	32.80 (91.16%)	42.70	1.32 (3.66%)	3345.43	860.92	1.00
	Medium-R	1163.66	1113.63 (95.70%)	392.78	23.04 (1.98%)	33 992.46	4530.68	4.27
	High-R	3268.63	2939.29 (89.92%)	3188.89	61.86 (1.89%)	199 815.97	18 680.27	358.24
Average		1207.34	1116.47 (92.92%)	796.39	22.89 (2.46%)	54 089.57	6691.84	58.09

Moreover, as the resource allowance grows, a set of more permissive resource constraints allows the algorithm to explore a wider range of solutions, thus yielding average costs that are generally lower. This is of course expected, since a feasible solution for low resource allowance is still feasible in the case of medium and high allowance. At times, though, even in presence of higher resource allowance it is possible to note higher average costs—see for example Type 1 Low-R and High-R instances in Table 4. In such cases, the average cost values are affected by the lower number of optimal solutions found in High-R instances, that leads to suboptimal solutions possibly worse than the optima of Low-R instances of the same networks.

This last observation suggests that: (i) given the lower computational times implied by the solution of instances with low resource allowance, and (ii) given that for a fixed network topology the feasible region of Low-R instances is a subset of Medium-R and High-R instances, the solution of a surrogate problem with lower resource allowance could be used to yield good-quality feasible solutions in the case of the harder High-R instances.

Lastly, Instances of Type 5 appear to be significantly easier to solve than Type 1 and 6. This observation is equally reflected in the average computational time required and the number of instances solved to optimality. This behavior can be related to the lower number of clusters that characterize Type 5 instances, as reported in Table 1.

6 | CONCLUSIONS

In this article, we present a new variant of Clustered Shortest Path Tree Problem (CluSPTP): the *Resource Constrained Clustered Shortest Path Tree Problem* (RC-CluSPTP). Since it could be considered as the clustered version of the Resource Constrained Shortest Path Tree problem, RC-CluSPTP is an NP-hard problem too.

A mathematical model along with its Dantzig-Wolfe decomposition are devised; additionally, we propose a Branch&Price for the resolution of RC-CluSPTP, which features a Column Generation approach with Multiple Pricing Scheme. The performance of this method are compared to those attained by CPLEX solver in the solution of the mathematical model, on a data-set obtained by adapting instances of the CluSPTP from literature. The results evidence that the proposed Branch&Price significantly improves the performance of CPLEX, solving to optimality a greater number of instances and taking less than half the computation time of CPLEX.

Due to the computational intractability of the RC-CluSPTP, as further investigation we will exploit meta-heuristic techniques to obtain suboptimal solutions of good quality in a reasonable computation time. Moreover, since only the case of local resources has been addressed in the present paper, as a future line of research we plan to formalize the RC-CluSPTP with global resource constraints, and to develop an ad hoc solution approach. Moreover, for both variants of the problem, we intend to explore different formulations (e.g., cut-based models) through a polyhedral study and a comparison of their relaxation quality.

ACKNOWLEDGMENT

The authors gratefully acknowledge the availability of the Calculation Centre SCoPE of the University of Naples "Federico II" and thanks are due to SCoPE academic staff for the given support. Open Access Funding provided by Universita degli Studi di Napoli Federico II within the CRUI-CARE Agreement.

REFERENCES

[1] C. Archetti, N. Bianchessi, and M. G. Speranza, *A column generation approach for the split delivery vehicle routing problem*, Networks **58** (2011), 241–254.

[2] P. Avella, M. Boccia, and A. Sforza, *Resource constrained shortest path problems in path planning for fleet management*, J. Math. Model. Algorithms **3** (2004), 1–17.
[3] X. Bao and Z. Liu, *An improved approximation algorithm for the clustered traveling salesman problem*, Inform. Process. Lett. **112** (2012), 908–910.
[4] H. T. T. Binh, P. D. Thanh, T. B. Trung, et al., *Effective multifactorial evolutionary algorithm for solving the cluster shortest path tree problem*, IEEE Congress Evol. Comput. **2018** (2018), 1–8.
[5] C. Boonmee, M. Arimura, and T. Asada, *Facility location optimization model for emergency humanitarian logistics*, Int. J. Disaster Risk Reduct. **24** (2017), 485–498.
[6] N. Cabrera, A. L. Medaglia, L. Lozano, and D. Duque, *An exact bidirectional pulse algorithm for the constrained shortest path*, Networks **76** (2020), 128–146.
[7] A. M. Caunhye, X. Nie, and S. Pokharel, *Optimization models in emergency logistics: A literature review*, Socio-Econ. Plan. Sci. **46** (2012), 4–13.
[8] S. Chen and Y. Shen, *An improved column generation algorithm for crew scheduling problems*, J. Info. Comput. Sci. **10** (2013), 175–183.
[9] O. Cosma, P. C. Pop, and I. Zelina, *A novel genetic algorithm for solving the clustered shortest-path tree problem*, Carpathian J. Math. **36** (2020), 401–414.
[10] O. Cosma, P. C. Pop, and I. Zelina, *An effective genetic algorithm for solving the clustered shortest-path tree problem*, IEEE Access **9** (2021), 15570–15591.
[11] G. B. Dantzig and P. Wolfe, *Decomposition principle for linear programs*, Oper. Res. **8** (1960), 101–111.
[12] M. D'Emidio, L. Forlizzi, D. Frigioni, S. Leucci, and G. Proietti, *On the clustered shortest-path tree problem*, ICTCS **1720** (2016), 263–268.
[13] M. Desrochers, *La fabrication d'horaires de travail pour les conducteurs d'autobus par Une méthode de génération de colonnes*, Université de Montréal, Centre de recherche sur les transports, 1986.
[14] M. Desrochers, An algorithm for the shortest path problem with resource constraints, Tech. Report G-88-27 (1988).
[15] J. Desrosiers, P. Pelletier, and F. Soumis, *Plus court chemin avec contraintes d'horaires*, RAIRO-Oper. Res. **17** (1983), 357–377.
[16] L. Di Puglia Pugliese, D. Ferone, P. Festa, and F. Guerriero, *Shortest path tour with time windows*, Eur. J. Oper. Res. **282** (2020), 334–344.
[17] X. Dong, W. Li, and L. Zheng, *Ant colony optimisation for a resource-constrained shortest path problem with applications in multimodal transport*, Int. J. Model, Identifi. Control **18** (2013), 268–275.
[18] I. Dumitrescu and N. Boland, *Improved preprocessing, labeling and scaling algorithms for the weight-constrained shortest path problem*, Networks **42** (2003), 135–153.
[19] M. D'Emidio, L. Forlizzi, D. Frigioni, S. Leucci, and G. Proietti, *Hardness, approximability, and fixed-parameter tractability of the clustered shortest-path tree problem*, J. Comb. Optim. **38** (2019), 165–184.
[20] D. Feillet, P. Dejax, M. Gendreau, and C. Gueguen, *An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems*, Networks **44** (2004), 216–229.
[21] C. Feremans, "*On generalized minimum spanning trees*," *Generalized spanning trees and extensions*, Universite Libre de Bruxelles, Bruxelles, Belgium, 2001.
[22] C. Feremans, M. Labbé, and G. Laporte, *Generalized network design problems*, Eur. J. Oper. Res. **148** (2003), 1–13.
[23] D. Ferone, P. Festa, S. Fugaro, and T. Pastore, *A dynamic programming algorithm for solving the k-color shortest path problem*, Optim. Lett. **15** (2020), 1973–1992.
[24] D. Ferone, P. Festa, S. Fugaro, and T. Pastore, On the shortest path problems with edge constraints. Proceedings of the 22nd International Conference on Transparent Optical Networks (ICTON), 2020, pp. 1–4.
[25] C. S. Ferreira, L. S. Ochi, V. Parada, and E. Uchoa, *A GRASP-based approach to the generalized minimum spanning tree problem*, Expert Syst. Appl. **39** (2012), 3526–3536.
[26] M. Fischetti, J. J. S. González, and P. Toth, *The symmetric generalized traveling salesman polytope*, Networks **26** (1995), 113–123.
[27] M. Fischetti, J. J. Salazar González, and P. Toth, *A branch-and-cut algorithm for the symmetric generalized traveling salesman problem*, Oper. Res. **45** (1997), 378–394.
[28] R. Garcia, *Resource constrained shortest paths and extensions*, Ph.D. thesis, Georgia Institute of Technology, Atlanta, US, 2009.
[29] G. Ghiani and G. Improta, *An efficient transformation of the generalized vehicle routing problem*, Eur. J. Oper. Res. **122** (2000), 11–17.
[30] N. Guttmann-Beck, R. Hassin, S. Khuller, and B. Raghavachari, *Approximation algorithms with bounded performance guarantees for the clustered traveling salesman problem*, Algorithmica **28** (2000), 422–437.
[31] G. Y. Handler and I. Zang, *A dual algorithm for the constrained shortest path problem*, Networks **10** (1980), 293–309.
[32] M. Horváth and T. Kis, *Solving resource constrained shortest path problems with LP-based methods*, Comput. Oper. Res. **73** (2016), 150–164.
[33] B. Hu, M. Leitner, and G. R. Raidl, *Combining variable neighborhood search with integer linear programming for the generalized minimum spanning tree problem*, J. Heuristics **14** (2008), 473–499.
[34] E. Ihler, Bounds on the quality of approximate solutions to the group Steiner problem, Int. Workshop on Graph-Theoretic Concepts in Computer Science (1990), 109–118.
[35] E. Ihler, G. Reich, and P. Widmayer, *Class Steiner trees and VLSI-design*, Discrete Appl. Math. **90** (1999), 173–194.
[36] S. Irnich, *Resource extension functions: Properties, inversion, and generalization to segments*, OR Spectr. **30** (2008), 113–148.
[37] S. Irnich and G. Desaulniers, "*Shortest path problems with resource constraints*," *Shortest path problems with resource constraints*, G. Column Generation, J. D. Desaulniers, and M. M. Solomon (eds.), Springer, US, Boston, MA, 2005, pp. 33–65.
[38] A. Koster, *Frequency assignment: Models and algorithms*, Ph.D. thesis, Maastricht University, Maastricht, Netherlands, 1999.
[39] W. J. Li, H. S. J. Tsao, and O. Ulular, *The shortest path with at most/nodes in each of the series/parallel clusters*, Networks **26** (1995), 263–271.
[40] C. W. Lin and B. Y. Wu, *On the minimum routing cost clustered tree problem*, J. Comb. Optim. **33** (2017), 1106–1121.
[41] L. Lozano and A. L. Medaglia, *On an exact method for the constrained shortest path problem*, Comput. Oper. Res. **40** (2013), 378–384.
[42] M. E. Lübbecke and J. Desrosiers, *Selected topics in column generation*, Oper. Res. **53** (2005), 1007–1023.
[43] Y. Marinakis, A. Migdalas, and A. Sifaleras, *A hybrid particle swarm optimization–variable neighborhood search algorithm for constrained shortest path problems*, Eur. J. Oper. Res. **261** (2017), 819–834.
[44] M. Mestria, *A hybrid heuristic algorithm for the clustered traveling salesman problem*, Pesqui. Oper. **36** (2016), 113–132.
[45] Y. S. Myung, C. H. Lee, and D. W. Tcha, *On the generalized minimum spanning tree problem*, Networks **26** (1995), 231–241.
[46] T. Öncan, J. F. Cordeau, and G. Laporte, *A tabu search heuristic for the generalized minimum spanning tree problem*, Eur. J. Oper. Res. **191** (2008), 306–319.
[47] M. M. Pascoal and A. Sedeño-Noda, *Enumerating K best paths in length order in dags*, Eur. J. Oper. Res. **221** (2012), 308–316.

[48] P. C. Pop, *The generalized minimum spanning tree problem: An overview of formulations, solution procedures and latest advances*, Eur. J. Oper. Res. **283** (2020), 1–15.

[49] P. C. Pop, L. Fuksz, A. H. Marc, and C. Sabo, *A novel two-level optimization approach for clustered vehicle routing problem*, Comput. Indus. Eng. **115** (2018), 304–318.

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

[51] G. Reich and P. Widmayer, Beyond Steiner's problem: A VLSI oriented generalization. Int. Workshop on Graph-Theoretic Concepts in Computer Science (1989), 196–210.

[52] G. Righini and M. Salani, *New dynamic programming algorithms for the resource constrained elementary shortest path problem*, Networks **51** (2008), 155–170.

[53] E. Rolland, R. A. Patterson, K. Ward, and B. Dodin, *Decision support for disaster management*, Oper. Manage. Res. **3** (2010), 68–79.

[54] D. M. Ryan and B. A. Foster, "*Computer scheduling of public transport: Urban passenger vehicle and crew scheduling*," *An integer programming approach to scheduling*, P. I. Feder (ed.), North-Holland Publishing Company, North-Holland, Amsterdam, 1981, pp. 269–280.

[55] L. Santos, J. Coutinho-Rodrigues, and J. R. Current, *An improved solution algorithm for the constrained shortest path problem*, Transp. Res. B: Methodol. **41** (2007), 756–771.

[56] A. Sedeño-Noda and S. Alonso-Rodr, *An enhanced K-SP algorithm with pruning strategies to solve the constrained shortest path problem*, Appl. Math. Comput. **265** (2015), 602–618.

[57] H. D. Sherali, T. B. Carter, and A. G. Hobeika, *A location-allocation model and algorithm for evacuation planning under hurricane/flood conditions*, Transp. Res. B: Methodol. **25** (1991), 439–452.

[58] S. L. Smith and F. Imeson, *GLNS: An effective large neighborhood search heuristic for the generalized traveling salesman problem*, Comput. Oper. Res. **87** (2017), 1–19.

[59] S. Srivastava, S. Kumar, R. Garg, and P. Sen, *Generalized traveling salesman problem through n sets of nodes*, Cors J. **7** (1969), 97–101.

[60] P. D. Thanh, H. T. T. Binh, N. B. Long, et al., *A heuristic based on randomized greedy algorithms for the clustered shortest-path tree problem*, IEEE Congress Evol. Comput. **2019** (2019), 2915–2922.

[61] P. D. Thanh, H. T. T. Binh, and T. B. Trung, *An efficient strategy for using multifactorial optimization to solve the clustered shortest path tree problem*, Appl. Intell. **50** (2020), 1233–1258.

[62] P. D. Thanh, D. A. Dung, T. N. Tien, and H. T. T. Binh, *An effective representation scheme in multifactorial evolutionary algorithm for solving cluster shortest-path tree problem*, IEEE Congress Evol. Comput. **2018** (2018), 1–8.

[63] B. W. Thomas, T. Calogiuri, and M. Hewitt, *An exact bidirectional a* approach for solving resource-constrained shortest path problems*, Networks **73** (2019), 187–205.

[64] C. Tilk and S. Irnich, *Combined column-and-row-generation for the optimal communication spanning tree problem*, Comput. Oper. Res. **93** (2018), 113–122.

[65] C. Vogiatzis, J. L. Walteros, and P. M. Pardalos, "*Evacuation through clustering techniques*," *Models, Algorithms, and Technologies for Network Analysis*, Springer, Boston, MA, 2013, pp. 185–198.

[66] B. Y. Wu and C. W. Lin, *On the clustered Steiner tree problem*, J. Comb. Optim. **30** (2015), 370–386.

[67] B. Yang and P. Gillard, *The class Steiner minimal tree problem: A lower bound and test problem generation*, Acta Inform. **37** (2000), 193–211.

[68] X. Zhu and W. E. Wilhelm, *A three-stage approach for the resource-constrained shortest path as a sub-problem in column generation*, Comput. Oper. Res. **39** (2012), 164–178.

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

How to cite this article: D. Ferone, P. Festa, S. Fugaro, and T. Pastore, *The resource constrained clustered shortest path tree problem: Mathematical formulation and Branch&Price solution algorithm*, Networks. (2022), 1–16. https://doi.org/10.1002/net.22124