The *k*-Color Shortest Path Problem

Daniele Ferone, Paola Festa, and Tommaso Pastore

Abstract This paper proposes a mathematical model and an exact algorithm for a novel problem, the k-Color Shortest Path Problem. This problem is defined on a edge-colored weighted graph, and its aim is to find a shortest path that uses at most k different edge-colors. The main support and motivation for this problem arise in the field of transmission networks design, where two crucial matters, reliability and cost, can be addressed using both colors and arc distances in the solution of a constrained shortest path problem. In this work, we describe a first mathematical formulation of the problem of interest and present an exact solution approach based on a branch and bound technique.

Keywords Constrained path problem · Edge-colored graph · Branch and bound

1 Introduction

Shortest Path Problems (SPPs) represent one of the most significant and investigated family of problems in Operations Research. The formulations that describe members of this class are often intuitive and easily relatable to real-world scenarios, while their broad applicability implies that often SPPs need to be solved as sub-tasks

D. Ferone
Department of Mechanical, Energy and Management Engineering, University of Calabria, Rende, Italy
e-mail: daniele.ferone@unical.it

P. Festa
Department of Mathematics and Applications, University of Naples Federico II, Naples, Italy
e-mail: paola.festa@unina.it

T. Pastore (✉)
Department of Structures for Engineering and Architecture, University of Naples Federico II, Naples, Italy
e-mail: tommaso.pastore@unina.it

M. Paolucci et al. (eds.), *Advances in Optimization and Decision Science for Society, Services and Enterprises*, AIRO Springer Series 3,
https://doi.org/10.1007/978-3-030-34960-8_32

in many other combinatorial problems, such as Maximum-Flow Minimum-Cost Problems or Vehicle Routing Problems.

While notably the classical SPPs can be optimally solved using polynomial algorithms, a more recent stream of research focused on the solution of either dynamic [12, 13] or constrained-SPPs [11, 14, 20], studying both exact and heuristic techniques. In this work, we define and formally state a novel problem: the k-Color Shortest Path Problem (k-CSPP), whose objective is to find a shortest path in a weighted edge-colored network, with a constraint on the maximum number of different colors that can be used. In the following we will use interchangeably the terms "colors" and "labels".

Edge-colored networks received a fair share of attention in the scientific literature, given their aptness for the depiction of complex and diverse relations among nodes. This feature proved to be beneficial in a wide variety of application fields, such as: computational biology [9], telecommunications [23], as well as in the analysis of transportation networks [1], and conflicts resolution [21].

In [4], it is proven that to find a path from a source node s to a destination node t with maximum k colors is a **NP**-complete problem, by reduction from the 3-SAT problem. We observe how any instance of the decisional problem of finding an $s-t$ path with at most k colors can be represented as an instance of k-CSPP, where each edge has null cost. Therefore, a polynomial algorithm for the k-CSPP would efficiently solve the decisional problem described in [4]. Consequently, k-CSPP is **NP**-hard.

In the study of edge-colored graphs, many works—of both theoretical and experimental interest—are concerned with the investigation of specific properly-colored edge structures, where a coloring is said to be proper whenever any two adjacent edges differ in color. These structures include for example: paths, trails, trees and cycles; see for example [17]. On the other hand, some classical optimization problems—such as the Minimum Spanning Tree (MST), the Traveling Salesman Problem (TSP), and the Longest Path Problem (LPP)—have all been extended to the case of edge-colored graphs, taking labels into account either in the objective function or in their constraints.

For example, the Minimum Label Spanning Tree is defined in [8] as a variant of the classical MST in which the cost of the spanning tree is given by the number of different edge-labels used. The authors of [8] describe a heuristic technique and an exact method based on the A* algorithm. The problem is further investigated in [6] which present a logarithmic approximation algorithm and a comparative study of several metaheuristic techniques, respectively. A strictly related generalization, the k-labeled Spanning Forest Problem, is studied in [7].

In [19], Jozefowiez et al. present an in-depth analysis of the Minimum Label Hamiltonian Cycle Problem (MLHCP). The MLHCP consists in the determination of an Hamiltonian cycle that presents the minimum total number of different edge-labels used. Moreover, in [19] two variations are introduced: the MLHP with length constraints and the Traveling Salesman Problem with label constraints (LCTSP). Aim of the LCTSP is to minimize the length of the tour—as in the classical TSP—while constraining the maximum number of different colors that can be used. The authors propose mathematical models and prove valid inequalities that are then included in branch-and-cut algorithms for the MLHCP and its two variants.

As an additional example, a special case of longest path on edge-colored graphs, the Orderly Colored Longest Path Problem, was recently studied in [5] by Carrabs et al. Suitably adapting existing formulations, the authors obtain several mathematical descriptions for the problem, that are then compared over a broad set of instances.

The main ground of interest for the k-CSPP arises in the field of telecommunications. While designing transmission networks, reliability is a crucial matter to ensure good performances and prevent data loss. As deeply discussed in [22] and [23], the robustness of a path-routed communication network can be achieved by means of path protection schemes, which make use of backup paths to ensure reachability in the case of single link failures. The backup path and the primary path are link-disjoint, and share the same source and destination. To prevent traffic loss, the backup path is activated whenever the primary path fails. On the other hand, often a single happening can cause the simultaneous failure of several links in the network. For instance, in WDM networks it is customary to bundle multiple fiber links in the same conduit. Consequently, even if these links are disjoint in the network layer, a damage to the conduit will cause the failure of all the links there bundled. The fibers sharing a common risk factor are said to be in the same Shared Risk Link Group, and are modeled with arcs of the same color. In [23], Yuan et al. address the failure minimization problem as a minimum-color path problem, in which each path is associated with one or more colors and each color is related to a given failure event. The authors support their approach arguing that by minimizing the number of colors involved in the path, the failure probability of the path can consequently be minimized. At the same time, while this argument handles different edge-labels, it does not include lengths in the comparison of different paths. Accordingly, a path with at most k different colors is connected if and only if failures do not occur in any of the k colors traversed in the path. If we make the assumption that color failures are mutually independent, and equiprobable—with probability $p \in [0, 1]$—, then the reliability of the path can be computed as $(1 - p)^k$. Consequently, an upper bound on the number of different colors allows to have a probabilistic estimate on the reliability of the network. A similar argument can be repeated in the case of independent failure events with different probabilities.

With in mind a similar network reliability scenario, the k-CSPP handles risk adversity as a strict requirement, while optimizing path length. Hence, the mathematical models introduced in the present paper include distances in the objective function, while encompassing the use of few colors in a problem constraint.

The main contributions of this work are: (1) a first formal description of the k-CSPP and (2) the design of a branch and bound algorithm. The work is organized as follows: in Sect. 2 some related works on Constrained Shortest Path Problems are presented. The problem is formally introduced in Sect. 3, where a mathematical model is proposed. Section 4 describes the Branch and Bound implementation used to optimally solve the problem. Computational results and a comparative analysis of the performances of our Branch and Bound with respect to the CPLEX solver are presented in Sect. 5. Concluding remarks are given in Sect. 6.

2 A Brief Taxonomy of Shortest Path Problems with Edge Constraints

The goal of this section is to provide a brief classification of some Shortest Path Problem variants that include edge constraints, with the aim of pointing out their differences with the k-CSPP.

One of the most broad and notable classes of edge-constrained SPPs is given by Resource Constrained Shortest Path Problems (RCSPPs) [3, 20].

In RCSPPs, in addition to the customary directed graph $G = (V, E)$ and edge-distance function $d_{ij} : E \to \mathbb{R}_0^+$, a L-dimensional vector of resources R is defined. Essentially, each resource is related to relevant link attributes that needs to be accounted in the planning of the path. Indeed, for each $l = 1, \dots, L$, to each $(i, j) \in E$ is associated a resource attribute r_{ij}^l. Accordingly, a path P^* is optimal whenever it is minimal w.r.t. the distance function d, and satisfies the restrictions enforced on the resources r_{ij}^l.

As argued in [2], the resources and the subsequent constraints can be of multifold nature. In fact, resources can model both numerical (either cumulative or non-cumulative) and categorical (or index) attributes. Straightforward examples of cumulative numerical attributes are travel time or fuel consumption, whose total use in a path P is obtained adding up all the travel times or consumption along the edges belonging to P, as in [10].

On the other hand, road width is an example of numerical non-cumulative parameter; in cases like this the feasibility of a path P can be subject to average or bottleneck restrictions, in which either the average of the resource or its minimum (respectively maximum) has to respect some bounds. Finally, RCSPPs can include categorical attributes, that can be used to specify the type of connection among two nodes. Whenever the problem considers arcs with categorical attributes, the formulation can feature constraints such as: a feasible path cannot contain links whose attribute is equal to a specific value (or a set of specific values). For a more detailed discussion see [2, 18].

The fundamental difference between the k-CSPP and RCSPPs lies in the fact that the k-CSPP does not restricts color-sets a priori and there is a strong interdependence among arcs. In our scenario, indeed, the cost of a color as a resource is not constant during the exploration of the solution space: once that an arc with a certain color is traversed, all other arcs sharing the same colors turn free and thus can be inserted in the solution without placing additional burden on the color constraint.

Another related idea studied in path problems on edge colored graphs consists in the use of *reload costs*. For each couples of colors (b, c) a reload cost $\rho_{b,c}$ is the amount to be paid if in the path P an arc of color c is traversed after an arc of color b. Gourvès et al. [16] studied the problem of minimizing reload costs for walks, trails and paths, deriving the resulting computational complexities. On the other hand, [1] consider a general form of objective function that includes both distances and reload cost. Aside from their presence in the objective function rather than in the constraints, the main difference between reload costs and the modeling paradigm of the k-CSPP is that reload costs are fixed, and have to be taken into account any given time there is a change from a color to another. On the contrary, bounding the maximum number of different colors, as required in the k-CSPP, means to count just once a transition to a specific color, whatever the preceding color (if any) may be.

3 Mathematical Model

Let $G = (V, E)$ be an undirected graph, with n nodes and m edges. Let the functions $C : E \longrightarrow \mathbb{N}$ and $d : E \longrightarrow \mathbb{R}_0^+$ be an edge-coloring and a non-negative distance function defined on E, respectively. The positive integer $C(e)$, $\forall e \in E$, is said to be the *color* of edge e.

Let $c(G)$ be the number of colors used to label the edges of G, for each color $h \in \{1, \dots c(G)\}$ all edges labeled with h are collected in color class C_h, in a way that $E = \bigcup_{h=1}^{c(G)} C_h$.

The k-CSPP consists in finding a shortest path $P^* = (v_1, v_2, \dots, v_h)$ from a source node $v_1 = s$ to a destination node $v_h = t$, with $s, t \in V$, such that the number of different colors traversed in the path does not exceed k.

Introducing a Boolean decision variable x_{ij}, for each edge $[i, j] \in E$, such that:

$$x_{ij} = \begin{cases} 1, & \text{if } [i, j] \text{ belongs to } P^*, \\ 0, & \text{otherwise,} \end{cases}$$

and for each possible color h, a Boolean decision variable y_h such that: $y_h = 1$, if color h is traversed in P^*, $y_h = 0$ otherwise. Then, the problem can be formulated as follows:

$$z = \min \sum_{[i,j] \in E} d_{ij} x_{ij} \tag{1a}$$

subject to:

$$\sum_{j \in V \setminus \{i\}} x_{ji} - \sum_{j \in V \setminus \{i\}} x_{ij} = b_i, \qquad \forall i \in V \tag{1b}$$

$$x_{ij} \le y_h, \quad \forall [i, j] \in C_h, h = 1, \dots, c(G) \tag{1c}$$

$$\sum_{h=1}^{c(G)} y_h \le k \tag{1d}$$

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

$$y_h \in \{0, 1\}, \qquad \forall h = 1, \ldots, c(G) \tag{1f}$$

with $b_i = -1$ for $i = s$, $b_i = 1$ for $i = t$, and $b_i = 0$ otherwise.

The objective function (1a) minimizes the total cost $d(p^*)$ of the solution path p^*. The constraints (1b) represent the flow balance constraint at each node. Constraints (1c) correlate arc traversal to color selection, and constraints (1d) impose the maximum number of colors that can be used in the path. Finally, the binary constraints are given in Eqs. (1e) and (1f).

4 Branch and Bound

The basic idea of the branch and bound here proposed consists in the observation that relaxing the color constraints (1d), the problem can be solved very efficiently by a classical shortest path algorithm. Consequently, at each node of the branching tree, a shortest path problem is solved on a given edge-colored graph $G' = (V, A')$. Then, if $p^*_{G'}$ is the optimal solution obtained, and $c(p^*_{G'})$ is the number of different colors used in $p^*_{G'}$, four cases can occur:

- $d(p^*_{G'}) = +\infty$: there is no path from s to d, the feasible region is empty, and the branching node becomes a leaf;
- $d(p^*_{G'}) \geq d(\hat{p})$: where $\hat{p}$ is the incumbent solution. In this case, the branching node is fathomed due to the bounding criterion;
- $c(p^*_{G'}) \leq k$: the solution is feasible for the original problem, and the incumbent is updated if necessary;
- $c(p^*_{G'}) = l > k$: the solution is not feasible for the original problem.

In the last case, a branching operation is performed. Let $C^* = \{c_1, \ldots, c_l\}$ be the colors used by the path $p^*_{G'}$, for each $i = 1, \ldots, l$ a new branching node is generated on the graph $G'' = (V, E'')$, where $E'' = E' \setminus \{e_{vw} \in E' \colon C(e_{vw}) = c_i\}$.

Moreover, the strategy guiding the exploration of the branching tree is a depth-first mechanism, and the nodes of the tree are generated excluding colors according to their absolute frequencies in $p^*_{G'}$. The lesser used the color, the earlier it is excluded from G'. The main target of this strategy is to obtain a feasible solution in the quickest way possible, in order exploit the bounding operation as much as possible. This aim is reflected by both the choice of the depth first strategy, and in the exclusion criterion considered for colors. The latter, indeed, tries to define a sub-problem favoring the constriction of colors that are less used in the computed path $p^*_{G'}$.

5 Experimental Results

This section presents some computational experiments designed to compare the performances of the two exact solution strategies, the Branch and Bound technique described in Sect. 4, and the model presented in Sect. 3 and solved with ILOG CPLEX 12.9. Both the algorithms were coded in C++ using the flags `-std=c++17 -O3` and compiled with `g++ 8.2`. The experiments were run on a `INTEL i5-6400@2.70 GHz` processor with 8GB of RAM. A time limit of 10 min has been used for both the algorithms.

The instances used in the experiments can be divided in two separate classes, both obtained with the a generator adapted from [15] to suitably introduce edge-colors. More specifically, the networks considered are either grid graphs or fully random networks. The total number of colors introduced in each graph amounts to either the 15% or 20% of the total number of edges.

Table 1 reports a summary of the instances included in the experimental phase. For each type, ten instances have been generated with different seeds.

The computational results obtained by our Branch and Bound (B&B), and CPLEX—executed either with depth first (df) or breadth first (bf) strategy—are reported in Tables 2 and 3 for fully random and grid graphs, respectively. More specifically, for each instance type we report: the average time spent by the algorithms solving instances of that type (avg. time), and the number of instances

Table 1 Instance parameters

Fully random graphs				Grid graphs		
Problem	Nodes	Arcs	Colors	Problem	Size	Colors
R1	75,000	750,000	112,500	G1	100 × 100	5940
R2	75,000	750,000	150,000	G2	100 × 100	7920
R3	75,000	112,500	168,750	G3	100 × 200	11,910
R4	75,000	112,500	225,000	G4	100 × 200	15,880
R5	75,000	150,000	225,000	G5	250 × 250	37,350
R6	75,000	150,000	300,000	G6	250 × 250	49,800
R7	100,000	1,000,000	150,000	G7	250 × 500	74,775
R8	100,000	1,000,000	200,000	G8	250 × 500	99,700
R9	100,000	1,500,000	225,000	G9	500 × 500	149,700
R10	100,000	1,500,000	300,000	G10	500 × 500	199,600
R11	100,000	2,000,000	300,000	G11	500 × 1000	299,550
R12	100,000	2,000,000	400,000	G12	500 × 1000	399,400
R13	125,000	1,250,000	187,500			
R14	125,000	1,250,000	250,000			
R15	125,000	1,875,000	281,250			
R16	125,000	1,875,000	375,000			
R17	125,000	2,500,000	375,000			
R18	125,000	2,500,000	375,000			

Table 2 Results on random graphs

	B&B		CPLEX (df)		CPLEX (bf)	
Instance type	Avg. time	O + F	Avg. time	O + F	Avg. time	O + F
R1	306.72	5 + 4	126.18	10 + 0	**92.43**	10 + 0
R2	308.49	5 + 4	118.66	9 + 0	**89.12**	9 + 0
R3	197.47	7 + 2	159.84	9 + 0	**158.54**	9 + 0
R4	199.00	7 + 2	210.22	8 + 0	**163.05**	9 + 0
R5	**322.67**	5 + 5	600.00	0 + 0	521.92	2 + 0
R6	**324.29**	5 + 5	600.00	0 + 0	513.47	2 + 0
R7	300.54	5 + 4	214.57	8 + 0	**123.57**	10 + 0
R8	300.62	5 + 2	289.92	6 + 0	**218.52**	8 + 0
R9	**360.75**	4 + 6	600.00	0 + 0	600.00	0 + 0
R10	**361.01**	4 + 6	600.00	0 + 0	600.00	0 + 0
R11	**121.40**	8 + 2	600.00	0 + 0	600.00	0 + 0
R12	**121.75**	8 + 2	600.00	0 + 0	600.00	0 + 0
R13	**383.90**	4 + 5	564.25	1 + 0	600.00	0 + 0
R14	**384.42**	4 + 5	557.50	1 + 0	512.84	2 + 0
R15	**420.33**	3 + 6	600.00	0 + 0	600.00	0 + 0
R16	**420.31**	3 + 6	600.00	0 + 0	600.00	0 + 0
R17	**305.85**	5 + 5	600.00	0 + 0	600.00	0 + 0
R18	**305.84**	5 + 5	600.00	0 + 0	600.00	0 + 0
Average	**302.52**		457.84		432.97	

Bold values indicate the algorithm with the shortest computational time average

of that type for which either an optimal (O) or feasible (F) solution has been found. This last information is collected in the column (O + F). Note that since each class is made up by 10 different instances, we have $O + F \leq 10$; whenever the preceding inequality its strict, then for some of the instances not even a feasible solution could be found within the time limit.

The results show how CPLEX is performing well on the smaller instances of the dataset, while the Branch and Bound approach is able to tackle larger instances, where the model becomes too large to be managed by CPLEX. For example, referring to graph types R15–R18, it is worthy to note that CPLEX can obtain just two feasible solutions—that happen to be optimal as well—within the required time limit. On the other hand, the branch and bound approach is able to obtain at least a feasible solution in 56 out of 60 total cases, guaranteeing optimality in 24.

Similarly, we can observe in Table 3 that even in the case of grid graphs our Branch and Bound outperforms CPLEX as the size of the network grows. Additionally, the results evidence higher computational times and a lower number of optimal solution found with respect to those reported in Table 2. For example, comparing the results obtained on instance classes G8 and R13—that have the same number of nodes—we can observe how the time spent by our Branch and Bound increased by 40%, and the number of optimal solution found within the time limit dropped by 50%. This behaviour is probably due to the greater sparsity of grid

Table 3 Results on grid graphs

	B&B		CPLEX (df)		CPLEX (bf)	
Instance type	Avg. time	O + F	Avg. time	O + F	Avg. time	O + F
G1	404.78	4 + 6	41.69	10 + 0	**41.05**	10 + 0
G2	403.93	4 + 6	57.15	10 + 0	**42.16**	10 + 0
G3	485.14	2 + 6	230.77	10 + 0	**200.71**	10 + 0
G4	543.64	1 + 7	230.45	10 + 0	**225.43**	9 + 0
G5	**578.72**	1 + 9	597.25	1 + 8	596.76	1 + 7
G6	**583.95**	1 + 9	587.59	2 + 7	587.55	2 + 7
G7	**532.55**	3 + 7	601.45	0 + 10	602.28	0 + 10
G8	**540.59**	2 + 8	601.28	0 + 10	602.36	0 + 10
G9	**540.24**	1 + 9	601.26	0 + 5	607.63	0 + 9
G10	**510.76**	2 + 7	600.00	0 + 0	607.07	0 + 10
G11	**540.36**	1 + 9	600.00	0 + 0	600.00	0 + 0
G12	**540.46**	1 + 9	600.00	0 + 0	600.00	0 + 0
Average	517.09		445.74		**442.75**	

Bold values indicate the algorithm with the shortest computational time average

graphs, that implies increased difficulties in the construction of a feasible k-color path.

6 Conclusions and Future Work

This paper presented a new variant of constrained shortest path problem, the k-Color Shortest Path (k-CSPP). The problem is formally described and a Branch and Bound is proposed for its solution. The performances of the exact method here described are then compared to those achieved by CPLEX in the solution of the integer programming model. The results evidence how our Branch and Bound can manage larger instances with respect to CPLEX, achieving good performances in terms of both optimal and feasible solutions found.

As future research perspectives—inspired by classical constrained shortest path literature—we plan to investigate the use of an exact dynamic programming algorithm. Additionally, given the complexity of the problem, future investigation will exploit metaheuristic techniques to quickly obtain good sub-optimal solutions. Moreover, a larger set of instances will be generated to properly assess a comparison between the methods, and to establish a shareable benchmark for future works.

References

1. Amaldi, E., Galbiati, G., Maffioli, F.: On minimum reload cost paths, tours, and flows. Networks **57**(3), 254–260 (2011)

2. Avella, P., Boccia, M., Sforza, A.: Resource constrained shortest path problems in path planning for fleet management. J. Math. Model. Algorithms **3**(1), 1–17 (2004)
3. Beasley, J.E., Christofides, N.: An algorithm for the resource constrained shortest path problem. Networks **19**(4), 379–394 (1989)
4. Broersma, H., Li, X., Woeginger, G.J., Zhang, S.: Paths and cycles in colored graphs. Australas. J. Comb. **31**, 299–312 (2005)
5. Carrabs, F., Cerulli, R., Felici, G., Singh, G.: Exact approaches for the orderly colored longest path problem: Performance comparison. Comput. Oper. Res. **101**, 275–284 (2019)
6. Cerulli, R., Fink, A., Gentili, M., Voß, S.: Metaheuristics comparison for the minimum labelling spanning tree problem. In: The Next Wave in Computing, Optimization, and Decision Technologies, pp. 93–106. Springer, New York (2005)
7. Cerulli, R., Fink, A., Gentili, M., Raiconi, A.: The k-labeled spanning forest problem. Proc. Soc. Behav. Sci. **108**, 153–163 (2014)
8. Chang, R.S., Shing-Jiuan, L.: The minimum labeling spanning trees. Inf. Process. Lett. **63**(5), 277–282 (1997)
9. Dorninger, D.: Hamiltonian circuits determining the order of chromosomes. Discrete Appl. Math. **50**(2), 159–168 (1994)
10. Feillet, D., Dejax, P., Gendreau, M., Gueguen, C.: An exact algorithm for the elementary shortest path problem with resource constraints: application to some vehicle routing problems. Networks **44**(3), 216–229 (2004)
11. Ferone, D., Festa, P., Guerriero, F., Laganà, D.: The constrained shortest path tour problem. Comput. Oper. Res. **74**, 64–77 (2016)
12. Ferone, D., Festa, P., Napoletano, A., Pastore, T.: Reoptimizing shortest paths: From state of the art to new recent perspectives. In: 2016 18th International Conference on Transparent Optical Networks (ICTON), pp. 1–5. IEEE, Piscataway (2016)
13. Ferone, D., Festa, P., Napoletano, A., Pastore, T.: Shortest paths on dynamic graphs: a survey. Pesquisa Operacional **37**(3), 487–508 (2017)
14. Ferone, D., Festa, P., Guerriero, F.: An efficient exact approach for the constrained shortest path tour problem. Optim. Methods Softw., 1–20 (2019). https://doi.org/10.1080/10556788.2018.1548015
15. Festa, P., Pallottino, S.: A pseudo-random networks generator. Technical report, Department of Mathematics and Applications "R. Caccioppoli", University of Napoli FEDERICO II, Italy (2003)
16. Gourvès, L., Lyra, A., Martinhon, C., Monnot, J.: The minimum reload s–t path, trail and walk problems. Discrete Appl. Math. **158**(13), 1404–1417 (2010)
17. Gourvès, L., Lyra, A., Martinhon, C.A., Monnot, J.: Complexity of trails, paths and circuits in arc-colored digraphs. Discrete Appl. Math. **161**(6), 819–828 (2013)
18. Irnich, S.: Resource extension functions: properties, inversion, and generalization to segments. OR Spectr. **30**(1), 113–148 (2008)
19. Jozefowiez, N., Laporte, G., Semet, F.: A branch-and-cut algorithm for the minimum labeling hamiltonian cycle problem and two variants. Comput. Oper. Res. **38**(11), 1534–1542 (2011)
20. Pugliese, L.D.P., Guerriero, F.: A survey of resource constrained shortest path problems: exact solution approaches. Networks **62**(3), 183–200 (2013)
21. Xu, H., Kilgour, D.M., Hipel, K.W., Kemkes, G.: Using matrices to link conflict evolution and resolution in a graph model. Eur. J. Oper. Res. **207**(1), 318–329 (2010)
22. Yuan, S., Jue, J.P., et al.: Shared protection routing algorithm for optical network. Opt. Netw. Mag. **3**(3), 32–39 (2002)
23. Yuan, S., Varma, S., Jue, J.P.: Minimum-color path problems for reliability in mesh networks. In: IEEE INFOCOM, vol. 4, p. 2658 (2005)