[33] S. Zajac. On a two-phase solution approach for the bi-objective k-dissimilar vehicle routing problem. *Journal Heuristics*, 24(3):515–550, 2018.

Ali Moghanni
University of Coimbra, CMUC, Department of Mathematics, 3001-501 Coimbra, Portugal
E-mail address: ali.moghanni@mat.uc.pt
URL: https://sites.google.com/view/alimoghanni/

Marta Pascoal
University of Coimbra, CMUC, Department of Mathematics, 3001-501 Coimbra, Portugal, Institute for Systems Engineering and Computers – Coimbra; and Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Piazza Leonardo da Vinci, 32, Milan, 20133, Italy
E-mail address: marta@mat.uc.pt

Maria Teresa Godinho
Department of Mathematics and Physical Sciences, Polytechnic Institute of Beja, Campus do Instituto Politécnico de Beja, rua Pedro Soares, 7800-295 Beja, Portugal, and CMAFcIO, Faculdade de Ciências da Universidade de Lisboa, Campo Grande, 1749-016 Lisboa
E-mail address: mtgodinho@ipbeja.pt

Pré-Publicações do Departamento de Matemática
Universidade de Coimbra
Preprint Number 21–33

FINDING K DISSIMILAR PATHS USING INTEGER LINEAR FORMULATIONS

ALI MOGHANNI, MARTA PASCOAL AND MARIA TERESA GODINHO

Abstract: While finding a path between two nodes is the basis for several applications, the need for alternative paths also may have various motivations. For instance, this can be of interest for ensuring reliability in a telecommunications network, for reducing the consequences of possible accidents in the transportation of hazardous materials, or to decrease the risk of robberies in money distribution. Each of these applications has particular characteristics, but they all have the common purpose of searching for a set of paths which are as dissimilar as possible with respect to the nodes/arcs that compose them.

In this work we present linear integer programming formulations for finding K dissimilar paths, with the main goal of preventing the overlap of arcs in the paths for a given integer K. The different formulations are tested for randomly generated general networks and for grid networks. The obtained results are compared in terms of the solutions' dissimilarity and of the run time. Two of the new formulations are able to find 10 paths with better average and minimum dissimilarity values than an iterative approach in the literature, in less than 20 seconds, for random networks with 500 nodes and 5000 arcs.

Keywords: K alternative paths, Dissimilarity, Integer linear programming formulations.
Math. Subject Classification (2020): 90C05, 90C10, 90C90.

1. Introduction

Let (N, A) denote a given directed network consisting of a set $N = \{1, \ldots, n\}$ of nodes and a set $A \subseteq N \times N$ of m arcs. Let s and t be two different nodes of N, called the source node and the target node, respectively. Finding a path in the network (N, A) between the nodes s and t is one of the most classical and widely used network optimization problems, and the basis for several applications in operations research. Studying the determination of alternative paths, on the other hand, is an interesting problem by itself that

Received September 19, 2021.

This work was partially financially supported by the Portuguese Foundation for Science and Technology (FCT) under project grants UID/MAT/04561/2019, UID/MAT/00324/2020 and UID/MULTI/00308/2020. The work was also partially supported by project P2020 SAICT-PAC/0011/2015, co-financed by COMPETE 2020, Portugal 2020 – Operational Program for Competitiveness and Internationalization (POCI), European Union's European Regional Development Fund, and FCT, and by FEDER Funds and National Funds under project CENTRO-01-0145-FEDER-029312.

stems from different real-life problems but has been considerably less studied than the former. For instance, in a modern and industrialized society, routing hazardous materials like poisonous gases or radioactive materials is an important issue, so the need for alternative safe routes is crucial for reducing the risk of disaster in case of accidents or if the best route becomes infeasible due to road construction [1, 4, 9, 16]. Repeating paths is also avoided in money collection, where having alternative paths/routes decreases the risk of robberies and can be used in case of danger of robberies [3, 7]. Additionally, in telecommunications, a backup path is often replaced by a primary one if a failure occurs along it or if it can be used simultaneously to spread information transmitted at a specific time [14, 15].

Let $K \in \mathbb{N}$ be a given number of alternative paths to be found. The definition of alternative paths may vary depending on the application, the common denominator being that the paths in the solution should share the least possible network resources. Several works use dissimilarity measures between two paths as the metric for achieving this purpose, nevertheless, also this notion is not uniquely defined, nor would that be desirable provided that the metrics are often tailored to the application. For instance, [12] developed four indices for measuring the similarity between two paths, defined as follows:

$$\text{Index 1: } S_1(p,q) = \frac{1}{2}\left(\frac{L(p \cap q)}{L(p)} + \frac{L(p \cap q)}{L(q)}\right)$$

$$\text{Index 2: } S_2(p,q) = \sqrt{\frac{L^2(p \cap q)}{L(p)L(q)}}$$

$$\text{Index 3: } S_3(p,q) = \frac{L(p \cap q)}{\max\{L(p), L(q)\}}$$

$$\text{Index 4: } S_4(p,q) = \frac{L(p \cap q)}{L(p \cup q)}$$

where p and q are two paths and $L(p)$ denotes the length of path p, that is, its number of arcs. The dissimilarity between p and q is then given by $D_i(p,q) = 1 - S_i(p,q)$, for $i = 1, 2, 3, 4$. The dissimilarities vary between 0 and 1, the first when the two paths coincide and the latter when they are arc disjoint. The authors also showed that there is a strong correlation between these indices. Other works have extended these concepts by including information about the underlying area affected by the paths or the distance between them, once again depending on the problem [9, 25].

[11] E. Erkut. The discrete p-dispersion problem. *European Journal of Operational Research*, 46:48–60, 1990.

[12] E. Erkut and V. Verter. Modeling of transport risk for hazardous materials. *Operations Research*, 46:625–642, 1998.

[13] A. Frank. Edge-disjoint paths in planar graphs. *Journal of Combinatorial Theory, Series B*, 39(2):164–178, 1985.

[14] T. Gomes and J. Craveirinha. An algorithm for enumerating srlg diverse path pairs. *Journal of Telecommunications and Information Technology*, pages 5–12, 2010.

[15] T. Gomes, L. Jorge, P. Melo, and R. Girão-Silva. Maximally node and srlg-disjoint path pair of min-sum cost in gmpls networks: a lexicographic approach. *Photonic Network Communications*, 31(1):11–22, 2016.

[16] R. Gopalan, K. Kolluri, R. Batta, and M. Karwan. Modeling equity of risk in the transportation of hazardous materials. *Operations research*, 38(6):961–973, 1990.

[17] V. Guruswami, S. Khanna, R. Rajaraman, B. Shepherd, and M. Yannakakis. Near-optimal hardness results and approximation algorithms for edge-disjoint paths and related problems. *Journal of Computer and System Sciences*, 67(3):473–496, 2003.

[18] W. Hoffman and R. Pavley. A method for the solution of the n-th best path problem. *Journal of the ACM*, 6(4):506–514, oct 1959.

[19] F. Iqbal and F. Kuipers. *Wiley Encyclopedia of Electrical and Electronics Engineering*, chapter Disjoint paths in networks, pages 1–14. Wiley, New York, 2015.

[20] V. Jiménez and A. Marzal. Computing the k shortest paths: A new algorithm and an experimental comparison. In *International Workshop on Algorithm Engineering*, pages 15–29. Springer, 1999.

[21] P.E. Johnson, D.S. Joy, and D. Clarke. Highway 3.01, an enhancement routing model: program, description, methodology and revised user's manual. *Arbeitsbericht, Oak Ridge National Laboratories, Washington, DC*, 1992.

[22] N. Katoh, T. Ibaraki, and H. Mine. An efficient algorithm for k shortest simple paths. *Networks*, 12(4):411–427, 1982.

[23] M. Kuby, X. Zhongyi, and X. Xiaodong. A minimax method for finding the k best "differentiated" paths. *Geographical Analysis*, 29:298–313, 1997.

[24] K. Lombard and R. Church. The gateway shortest path problem: Generating alternative routes for a corridor location problem. *Geographical Systems*, 1(1):25–45, 1 1993.

[25] R. Martí, J. González-Velarde, and A. Duarte. Heuristics for the bi-objective path dissimilarity problem. *Computers & Operations Research*, 36(11):2905–2912, 2009.

[26] E. Martins. An algorithm for ranking paths that may contain cycles. *European Journal of Operational Research*, 18(1):123–130, 1984.

[27] E. Martins and M. Pascoal. A new implementation of Yen's ranking loopless paths algorithm. *Quarterly Journal of the Belgian, French and Italian Operations Research Societies*, 1(2):121–133, 2003.

[28] E. Martins, M. Pascoal, and J. Santos. Deviation algorithms for ranking shortest paths. *International Journal of Foundations of Computer Science*, 10(3):247–261, 1999.

[29] J. Suurballe. Disjoint paths in a network. *Networks*, 4(2):125–145, 1974.

[30] J. Suurballe and R. Tarjan. A quick method for finding shortest pairs of disjoint paths. *Networks*, 14(2):325–336, 1984.

[31] J. Vygen. NP-completeness of some edge-disjoint paths problems. *Discrete Applied Mathematics*, 61(1):83–90, 1995.

[32] J.Y. Yen. Finding the k shortest loopless paths in a network. *Management Science*, 17(11):712–716, 1971.

TABLE 29. Run times of IPM (seconds)

	K								
$R_{n,m}$	3	4	5	6	7	8	9	10	
$R_{100,500}$	0.021	0.028	0.034	0.040	0.046	0.057	0.063	0.070	
$R_{100,1000}$	–	0.053	0.065	0.076	0.089	0.101	0.113	0.126	
$R_{300,1500}$	0.064	0.087	0.110	0.129	0.150	0.176	0.197	0.219	
$R_{300,3000}$	0.116	0.151	0.181	0.210	0.244	0.279	0.311	0.344	
$R_{500,2500}$	0.109	0.139	0.174	0.207	0.245	0.285	0.317	0.351	
$R_{500,5000}$	0.191	0.251	0.322	0.395	0.469	0.545	0.599	0.691	
Average	0.100	0.118	0.147	0.176	0.207	0.240	0.266	0.300	0.194

	K								
$G_{p,q}$	3	4	5	6	7	8	9	10	
$G_{3,12}$	0.000	0.002	0.003	0.004	0.005	0.003	0.005	0.005	
$G_{4,36}$	0.008	0.013	0.016	0.012	0.021	0.026	0.030	0.033	
$G_{6,6}$	0.003	0.003	0.001	0.002	0.004	0.000	0.007	0.008	
$G_{12,12}$	0.005	0.007	0.010	0.013	0.011	0.018	0.017	0.014	
Average	0.004	0.006	0.007	0.007	0.010	0.011	0.014	0.015	0.009

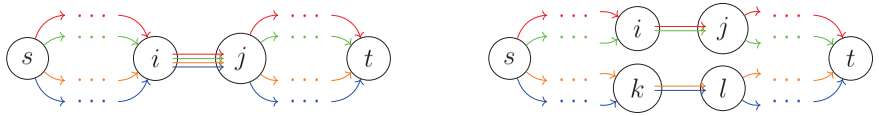
TABLE 30. Average AvDi and MiDi of IPM in grid networks

AvDi	K								
$G_{p,q}$	3	4	5	6	7	8	9	10	
$G_{3,12}$	0.948	0.833	0.800	0.769	0.758	0.741	0.732	0.728	
$G_{4,36}$	0.892	0.912	0.982	0.859	0.830	0.817	0.805	0.798	
$G_{6,6}$	0.933	0.933	0.900	0.893	0.866	0.857	0.844	0.840	
$G_{12,12}$	0.969	0.969	0.954	0.951	0.939	0.935	0.929	0.926	
Average	0.958	0.911	0.886	0.868	0.848	0.837	0.827	0.823	0.870

MiDi	K								
$G_{p,q}$	3	4	5	6	7	8	9	10	
$G_{3,12}$	0.846	0.153	0.153	0.153	0.153	0.000	0.000	0.000	
$G_{4,36}$	0.947	0.526	0.526	0.500	0.500	0.131	0.078	0.078	
$G_{6,6}$	0.900	0.900	0.600	0.600	0.600	0.600	0.500	0.500	
$G_{12,12}$	0.954	0.954	0.818	0.818	0.681	0.681	0.681	0.681	
Average	0.911	0.633	0.524	0.517	0.483	0.353	0.314	0.314	0.506

[9] P. Dell'Olmo, M. Gentili, and A. Scozzari. On finding dissimilar Pareto-optimal paths. *European Journal of Operational Research*, 162:70–82, 2005.
[10] D. Eppstein. Finding the k shortest paths. *SIAM Journal on Computing*, 28(2):652–673, 1998.

Additionally, in concrete applications the problem has frequently been handled from a bi-objective point of view, having the goals of optimizing both the total paths length/cost as well as the dissimilarity of the set of paths. Now, while the shortest path problem or the ranking of K shortest paths problem are well-known and well-studied problems, the same is not true when the objective function represents paths dissimilarity. Many of these bi-objective problems have been addressed from an algorithmic approach whose primary goal is not to optimize the dissimilarity and, to our knowledge, there are no published studies considering the paths dissimilarity problem from an integer programming point of view.



(A) Solution with one repeated arc, shared by several paths

(B) Solution with many repeated arcs, shared by few paths

FIGURE 1. Different sets of $K = 4$ paths

Filling this gap and deepening the understanding of such problems are the main motivations for the present work, also because this may be a relevant contribution for an efficient treatment of a bi-objective problem involving paths dissimilarity.

As mentioned earlier, it is not uncommon to find different understandings of the term "dissimilarity" in the literature. In this work, we focus on the conception of dissimilarity as defined by D_1. Thus, the presented models aim at producing sets of K paths with good scores in terms of D_1.

However, modeling D_1 as the objective function of an integer linear programming (ILP) model, presents some difficulties as this is a non linear metric involving an underlying combinatorial problem. On the other hand, it is intuitive that minimizing the number of arcs shared by the K paths or minimizing the number of paths that share a common arc in the K paths, favor the dissimilarity of the solution. Furthermore, these problems can be modeled quite easily, obviating the above mentioned difficulties. Thus these alternative ways of looking into the K dissimilar paths problem seem promising and are worth exploring.

In the present work we introduce and compare three families of ILP formulations, each one addressing one of the strategies mentioned before. Figure 1 illustrates their differences and emphasizes some of their limitations.

For simplicity, assume that all the paths represented in Figure 1 have the same length. There are 6 arc overlaps for every pair of paths in Figure 1a

(that is, the sum of $L(p \cap q)$ for every pair of paths p, q in the solution is equal to 6), while in the solution depicted in Figure 1b there are only 2. Therefore, the second is a better solution than the first with regard to their dissimilarity. However, if one chooses to count the arcs shared by more than one path, there is only 1 in the solution in Figure 1a and there are 2 in the solution in Figure 1b. Thus, the first solution is the best with regard to this metric. This situation illustrates the major drawback of the strategy devised by this approach – the lack of control over the number of overlaps associated to arcs that are used by more than one path. By contrast, if counting the number of arc repetitions in the solution (given by the number of times an arc is present in the solution, besides its first use), there are 3 in the solution depicted in Figure 1a and 2 in the solution depicted in Figure 1b, favoring the solution that seems to be the most dissimilar one. Still, this third approach is not exempted of drawbacks, as it does not distinguish between alternative solutions with different dissimilarities.

To (partially) overcome the drawbacks of the two approaches a set of additional constraints is proposed, which imposes a bound on the number of paths that use each arc in the solution. Formal definitions of each of these problems and concepts are introduced later on.

The remainder of this work is structured as follows. Section 2 gives a literature overview of problems related with finding alternative paths. In Section 3 the problem of finding K dissimilar paths between two nodes is presented, and in the next three sections ILP formulations are introduced to provide solutions based on the approaches described above. Computational results for different variants of each formulation are presented at the end of each section. A set of constraints which help the new formulations to obtain more dissimilar solutions is introduced in Section 7. Overall computational experiments for all approaches are presented in Section 8. The performance of the proposed formulations is analyzed and this is compared to an intuitive and classical approach in the literature of finding alternative paths, the iterative penalty method (IPM) [21]. The formulations are compared both in terms of the run time and of the dissimilarity of the output solutions, derived from D_1. Some conclusions and future directions for research are outlined in Section 9.

2. Literature review

Several problems focus on finding a single path between two nodes in a network, which optimizes either a certain criterion or several criteria simultaneously, while others aim at finding a set with a given number of paths, again with respect to one, or several, criteria. The single path problems have practical interest by themselves, but finding a set of paths may still be

Appendix C. Dissimilarities and run times for the IPM

TABLE 28. Average AvDi and MiDi of IPM in random networks

AvDi	K								
$R_{n,m}$	3	4	5	6	7	8	9	10	
$R_{100,500}$	0.712	0.823	0.841	0.847	0.839	0.846	0.830	0.830	
$R_{100,1000}$	–	0.857	0.910	0.918	0.941	0.943	0.947	0.941	
$R_{300,1500}$	0.774	0.868	0.892	0.895	0.890	0.886	0.888	0.885	
$R_{300,3000}$	0.917	0.913	0.937	0.919	0.929	0.942	0.942	0.941	
$R_{500,2500}$	0.896	0.911	0.904	0.907	0.911	0.911	0.906	0.904	
$R_{500,5000}$	0.764	0.853	0.867	0.894	0.909	0.917	0.927	0.927	
Average	0.812	0.870	0.891	0.896	0.903	0.907	0.906	0.904	0.886

MiDi	K								
$R_{n,m}$	3	4	5	6	7	8	9	10	
$R_{100,500}$	0.429	0.485	0.484	0.428	0.373	0.301	0.212	0.158	
$R_{100,1000}$	–	0.145	0.312	0.316	0.430	0.419	0.423	0.369	
$R_{300,1500}$	0.613	0.642	0.621	0.621	0.491	0.347	0.338	0.251	
$R_{300,3000}$	0.750	0.531	0.550	0.541	0.503	0.521	0.500	0.464	
$R_{500,2500}$	0.736	0.640	0.634	0.576	0.569	0.526	0.498	0.448	
$R_{500,5000}$	0.525	0.588	0.539	0.530	0.564	0.564	0.580	0.580	
Average	0.610	0.505	0.523	0.502	0.488	0.446	0.425	0.378	0.484

References

[1] V. Akgün, E. Erkut, and R. Batta. On finding dissimilar paths. *European Journal of Operational Research*, 121:232–246, 2000.

[2] R. Bhandari. *Survivable Networks: Algorithms for Diverse Routing*. Kluwer Academic Publishers, USA, 1998.

[3] R. Calvo and R. Cordone. A heuristic approach to the overnight security service problem. *Computers & Operations Research*, 30(9):1269–1287, 2003.

[4] M. Caramia, S. Giordani, and A. Iovanella. On the selection of k routes in multiobjective hazmat route planning. *IMA Journal of Management Mathematics*, 21(3):239–251, 2010.

[5] P. Carotenuto, S. Giordani, and S. Ricciardelli. Finding minimum and equitable risk routes for hazmat shipments. *Computers & Operations Research*, 34(5):1304–1327, 2007.

[6] P. Carotenuto, S. Giordani, S. Ricciardelli, and S. Rismondo. A tabu search approach for scheduling hazmat shipments. *Computers & Operations Research*, 34(5):1328–1350, 2007.

[7] M. Constantino, M. C. Mourão, and L. Pinto. Dissimilar arc routing problems. *Networks*, 70(3):233–245, 2017.

[8] T. Cormen, C. Leiserson, R. Rivest, and C. Stein. *Introduction to algorithms*. The MIT press, 2009.

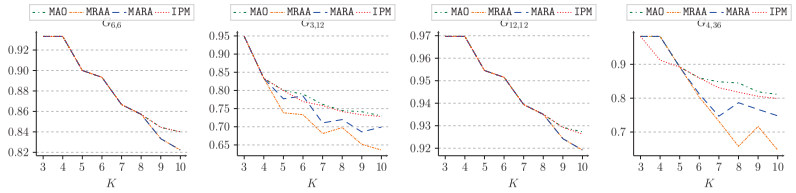


FIGURE 31. Average dissimilarity in grid networks

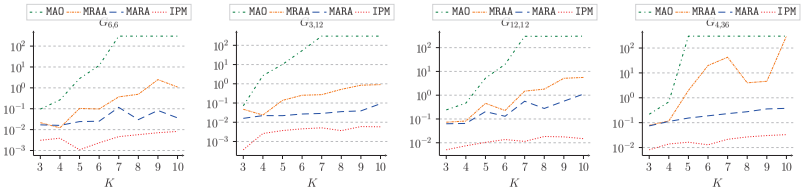


FIGURE 32. Run times in grid networks (seconds – log scale)

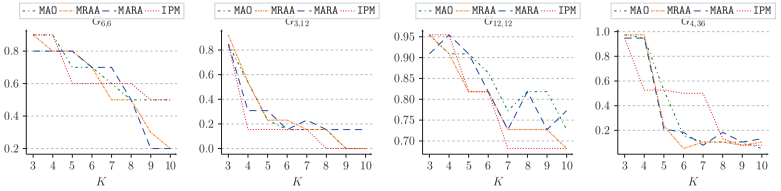


FIGURE 33. Minimum dissimilarity in grid networks

relevant, for instance to ensure reliability and having a replacement path in case of failure in the primary one, or simply if several alternatives should avoid sharing resources with other paths. In this case ranking paths provides a pre-defined number of paths from the source node to the target node by increasing order of the objective function.

Several network optimization problems search for a path between two nodes which optimizes a certain criterion. In many cases it is of interest to extend this problem by searching not only for the best solution but also for the second best, the third best and so on, that is, to rank paths by increasing order of the objective function. In practice this is useful, for instance, when the paths need to satisfy additional constraints, which can be checked as new solutions are found. This problem, known as the K shortest paths problem, was first proposed in 1959, by Hoffman [18], and is usually classified into two variants, one that aims at the determination of unconstrained paths and another one for which the nodes in each solution cannot appear more than once. Despite the first being easier to solve than the second, both can be solved in polynomial space and time, depending on the number K of paths to be found and on the size of the network. See for example [10, 20, 26, 28] for works on ranking unconstrained paths and [22, 27, 32] for works on ranking loopless paths.

The search for solutions when ranking paths is guided by the objective function, therefore, very often the solutions which are close in terms of the cost are also similar in terms of their composition. In the K disjoint paths problem, a cost objective function of K paths is minimized, while the overlaps between them are forbidden. The problem can be classified into arc disjoint or node disjoint, the second one being a particular case of the first (for instance, if every node is split into two nodes that are linked by an arc). The disjointness of the paths is often a requirement in telecommunications, in order to ensure the reliability of communications. In practice this is managed by the computation of a pair of paths connecting two given nodes, a primary path to be used as a first option and a backup path to replace the first one if there is a failure along its arcs or nodes. The determination of K disjoint simple paths has been studied by [2, 29, 30]. Their approaches consist of formulating the problem as a minimum cost flow problem and propose the application of a labeling algorithm, changing the given network. The arc disjoint version of the problem has been studied in [13, 17, 31]. A review on disjoint path problems can be found in [19].

A handicap of the K disjoint paths problem is that the disjointness condition may be too demanding for some instances and no solution is returned in those cases. The dissimilar paths problem has been studied in the context of hazardous materials transportation, where the alternative paths should not share a large number of arcs and they should be relatively short in length.

Different methods have been proposed for approaching this problem. The IPM [21] is one of the most intuitive methods, based on the iterative application of shortest path algorithms. At each iteration, a cost penalty is associated to each selected arc to discourage them of appearing in the forthcoming iteration; hence, generating dissimilar paths. Another proposed method is the Gateway Shortest Path, [24]. In this case, the generated shortest paths should go through a given set of nodes called a "gateway". Additionally, the concept of "area under a path" is used to evaluate the similarity between two paths. The minimax method, by Kuby et al. [23], selects paths starting from K assigned paths using some dissimilarity indices. Akgün et al. [1] review the three mentioned methods for generating dissimilar paths, and proposes another dissimilar path model that makes use of a p-dispersion location model, [11]. Erkut [12] presents four indices to measure the dissimilarity among two paths, one of which will be used later. In [5], the authors introduce a model for generating dissimilar paths that takes into account also the risk induced on the arcs in the neighborhood of a selected path. Another work, [6], also considers the need to distribute the risk of the paths in an equitable way with respect to both the space and the time, avoiding as much as possible the presence of more than one hazardous vehicle at the same time on the same zone. Later on Dell'Olmo et al. [9] study the problem from a multi-objective perspective. They introduce the concept of "buffer zone" in the measure of similarity. Martí et al. [25] choose approaches different from the previous and consider a spatial point of view in their dissimilarity index. More recently, Zajac [33] works on the bi-objective K dissimilar vehicle routing problem (kd-VRP). The work considers two dissimilarity indices: the "grid metric", which treats spatial dissimilarity, as well as the "edge metric", which defines dissimilarity via shared arcs between different routes.

3. The K dissimilar paths problem

Let (N, A) be a directed graph with $|N| = n$ nodes and $|A| = m$ arcs, where s denotes a given source node and t denotes a given target node, $s, t \in N$. Let also P denote the set of paths in (N, A) from node s to node t and K be a given positive integer. The goal of the K dissimilar paths problem from s to t is to find a set of K paths in P, such that the paths in the set are fairly distributed throughout the network. Considering dissimilarity based in the index S_1 introduced in Section 1, the problem can be defined as

$$\begin{aligned} \max \quad & \sum_{i=1}^{K-1} \sum_{j=i+1}^{K} D_1(p_i, p_j) / \binom{K}{2} \\ \text{subject to} \quad & p_1, p_2, \ldots, p_K \in P \end{aligned}$$

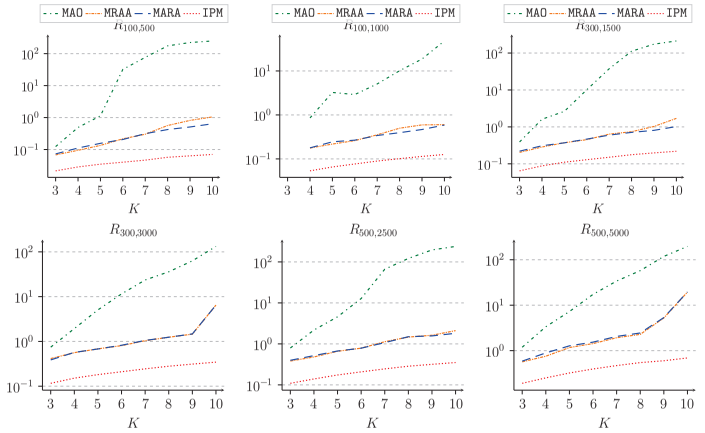


FIGURE 29. Run times in random networks (seconds – log scale)

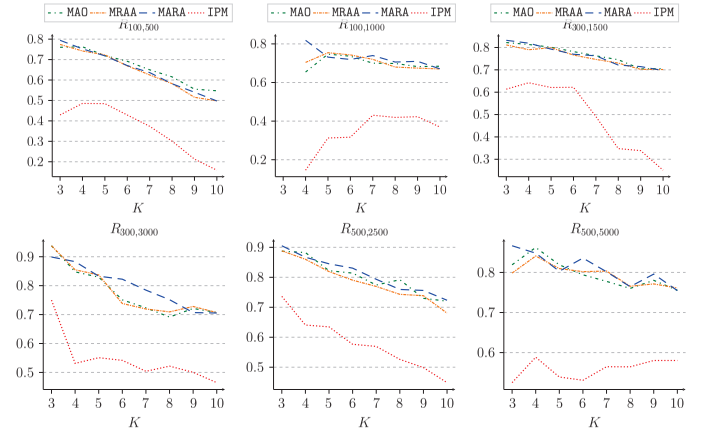


FIGURE 30. Minimum dissimilarity in random networks

TABLE 26. Average AvDi and MiDi of MRAA in grid networks

AvDi	K								
$G_{p,q}$	3	4	5	6	7	8	9	10	
$G_{3,12}$	0.949	0.833	0.738	0.703	0.714	0.681	0.656	0.641	
$G_{4,36}$	0.982	0.982	0.892	0.804	0.732	0.759	0.702	0.718	
$G_{6,6}$	0.933	0.933	0.900	0.893	0.867	0.857	0.833	0.822	
$G_{12,12}$	0.970	0.970	0.955	0.952	0.939	0.935	0.924	0.919	
Average	0.959	0.930	0.871	0.838	0.813	0.808	0.779	0.775	0.847

MiDi	K								
$G_{p,q}$	3	4	5	6	7	8	9	10	
$G_{3,12}$	0.923	0.538	0.231	0.231	0.154	0.154	0.000	0.000	
$G_{4,36}$	0.974	0.947	0.237	0.053	0.105	0.105	0.079	0.105	
$G_{6,6}$	0.900	0.800	0.800	0.700	0.500	0.500	0.300	0.200	
$G_{12,12}$	0.955	0.909	0.818	0.818	0.727	0.727	0.727	0.682	
Average	0.938	0.805	0.521	0.450	0.372	0.372	0.277	0.247	0.498

TABLE 27. Average AvDi and MiDi of MARA in grid networks

AvDi	K								
$G_{p,q}$	3	4	5	6	7	8	9	10	
$G_{3,12}$	0.949	0.833	0.777	0.785	0.711	0.720	0.686	0.699	
$G_{4,36}$	0.982	0.982	0.982	0.812	0.747	0.787	0.767	0.748	
$G_{6,6}$	0.933	0.933	0.900	0.893	0.867	0.857	0.833	0.822	
$G_{12,12}$	0.970	0.970	0.955	0.952	0.939	0.935	0.924	0.919	
Average	0.959	0.930	0.881	0.860	0.816	0.825	0.803	0.797	0.859

MiDi	K								
$G_{p,q}$	3	4	5	6	7	8	9	10	
$G_{3,12}$	0.846	0.308	0.308	0.154	0.231	0.154	0.154	0.154	
$G_{4,36}$	0.974	0.947	0.211	0.184	0.079	0.184	0.105	0.132	
$G_{6,6}$	0.800	0.800	0.800	0.700	0.700	0.500	0.200	0.200	
$G_{12,12}$	0.909	0.955	0.909	0.818	0.727	0.818	0.727	0.773	
Average	0.876	0.752	0.557	0.464	0.434	0.414	0.297	0.315	0.514

where

$$D_1(p,q) = 1 - S_1(p,q) = 1 - \frac{1}{2}\left(\frac{L(p \cap q)}{L(p)} + \frac{L(p \cap q)}{L(q)}\right),$$

for any paths $p, q \in P$, and this is equivalent to minimizing the similarity of the set of K paths,

$$\sum_{i=1}^{K-1} \sum_{j=i+1}^{K} \left(\frac{L(p_i \cap p_j)}{L(p_i)} + \frac{L(p_i \cap p_j)}{L(p_j)}\right).$$

This objective function is fractional and difficult to handle directly. Therefore, we will consider simplifications of the problem.

The next sections introduce three formulations, simpler than this one, but which try to capture the main characteristic of this problem based on different assumptions.

4. Minimization of the number of arc overlaps for each pair of paths

The length of the paths overlap is a common term to the several similarity indices described in Section 1. For now we focus on that length and, by doing so, the objective function becomes linear and simpler to handle. Thus, the current goal is to find solutions for

$$\begin{aligned} \min \quad & \sum_{i=1}^{K-1} \sum_{j=i+1}^{K} L(p_i \cap p_j) \\ \text{subject to} \quad & p_1, p_2, \ldots, p_K \in P \end{aligned} \tag{1}$$

Given two paths $p, q \in P$, it is said that there is an overlap whenever there is an arc $(i,j) \in A$ that belongs to both p and q, that is, if $(i,j) \in p$ and $(i,j) \in q$. Thus, the number of overlaps between those paths is the number of arcs that appear in both, $OL(p,q) = |\{(i,j) \in A : (i,j) \in p \wedge (i,j) \in q\}|$, which coincides with $L(p \cap q)$. The number of overlaps in a given set of K paths is the total number of overlaps for each pair of paths, that is,

$$OL(\{p_1, p_2, \ldots, p_K\}) = \sum_{i=1}^{K-1} \sum_{j=i+1}^{K} OL(p_i, p_j),$$

which is the objective function in problem (1). To illustrate these concepts, we recall the example in Figure 1. The arc (i,j) belongs to all four paths in Figure 1a. Therefore, the number of overlaps for those paths is 6. In Figure 1b the arc (i,j) appears in the paths in red and green, while the arc (k,l)

appears in the blue and orange paths. So, there are 2 arc overlaps in the 4 paths. This latter solution is better than the first with respect to problem (1).

The first formulation intends to model problem (1) as an integer linear program. Considering the decision variables x_{ij}^k equal to 1 if the arc (i,j) lies in the k-th path from s to t, or 0 otherwise, for any $(i,j) \in A$ and $k = 1, \ldots, K$, the problem is formulated as follows:

$$\min \quad f_1(x,z,v) = \sum_{(i,j)\in A} v_{ij} \tag{2a}$$

$$\text{subject to} \quad \sum_{j\in N:(i,j)\in A} x_{ij}^k - \sum_{j\in N:(j,i)\in A} x_{ji}^k = \begin{cases} 1 & i = s \\ 0 & i \neq s,t \\ -1 & i = t \end{cases}, k = 1, \ldots, K \tag{2b}$$

$$z_{ij}^{kl} \leq x_{ij}^k, \ (i,j) \in A, \ k,l = 1, \ldots, K, \ l > k \tag{2c}$$

$$z_{ij}^{kl} \leq x_{ij}^l, \quad (i,j) \in A, \ k,l = 1, \ldots, K, \ l > k \tag{2d}$$

$$z_{ij}^{kl} \geq x_{ij}^k + x_{ij}^l - 1, \quad (i,j) \in A, \quad k,l = 1, \ldots, K, \quad l > k \tag{2e}$$

$$v_{ij} = \sum_{k=1}^{K-1} \sum_{l=k+1}^{K} z_{ij}^{kl}, \quad (i,j) \in A \tag{2f}$$

$$x_{ij}^k, z_{ij}^{kl} \in \{0,1\}, \ (i,j) \in A, \ k,l = 1, \ldots, K, \ l > k \tag{2g}$$

This formulation may considerably big for problems of modest size, given that the variables z are related with any pair of paths from s to t. Thus, it has $O(mK^2)$ variables and $O(mK^2 + nK)$ constraints.

The flow conservation constraints (2b) ensure the existence of K paths from node s to node t; the constraints (2c) – (2e) guarantee that variables $z_{ij}^{kl} \in \{0,1\}$ are equal to 1 if and only if the arc (i,j) is used both in the paths defined by the variables x_{ij}^k and x_{ij}^l, for $(i,j) \in A$, $k \in \{1, \ldots, K-1\}$, $l \in \{k+1, \ldots, K\}$. In fact:

- If $x_{ij}^k = 0$ or $x_{ij}^l = 0$, then (2c) or (2d) imply that $z_{ij}^{kl} = 0$, for any $(i,j) \in A$, $k \in \{1, \ldots, K-1\}$, $l \in \{k+1, \ldots, K\}$.
- If $x_{ij}^k = x_{ij}^l = 1$, then both (2c) and (2d) imply that $z_{ij}^{kl} \leq 1$, whereas (2e) imply that $z_{ij}^{kl} \geq 1$. Thus, $z_{ij}^{kl} = 1$ for $(i,j) \in A$, $k \in \{1, \ldots, K-1\}$, $l \in \{k+1, \ldots, K\}$.

The constraints (2f) state that the auxiliary variables v_{ij} correspond to the number of paths that use arc $(i,j) \in A$ and constraints (2g) define the variables. Observe that because the x_{ij}^k are binary variables, then by (2c)

TABLE 24. Run times of MRAA (seconds)

	K								
$R_{n,m}$	3	4	5	6	7	8	9	10	
$R_{100,500}$	0.068	0.095	0.135	0.216	0.301	0.564	0.815	1.048	
$R_{100,1000}$	–	0.177	0.218	0.261	0.353	0.499	0.594	0.601	
$R_{300,1500}$	0.204	0.285	0.372	0.451	0.637	0.741	1.028	1.702	
$R_{300,3000}$	0.414	0.561	0.682	0.805	1.039	1.241	1.455	6.490	
$R_{500,2500}$	0.382	0.480	0.661	0.787	1.131	1.468	1.625	2.109	
$R_{500,5000}$	0.567	0.755	1.171	1.432	1.924	2.283	5.247	19.440	
Average	0.327	0.329	0.540	0.659	0.898	1.133	1.794	5.320	1.394
	K								
$G_{p,q}$	3	4	5	6	7	8	9	10	
$G_{3,12}$	0.047	0.024	0.137	0.253	0.272	0.514	0.842	0.905	
$G_{4,36}$	0.078	0.117	0.197	19.864	42.651	3.999	4.611	300	
$G_{6,6}$	0.022	0.012	0.103	0.097	0.370	0.493	2.487	1.093	
$G_{12,12}$	0.073	0.082	0.456	0.227	1.484	1.802	5.142	5.582	
Average	0.055	0.059	0.667	5.111	11.194	1.702	3.270	76.915	12.371

TABLE 25. Run times of MARA (seconds)

	K								
$R_{n,m}$	3	4	5	6	7	8	9	10	
$R_{100,500}$	0.073	0.112	0.156	0.206	0.313	0.422	0.508	0.639	
$R_{100,1000}$	–	0.178	0.245	0.267	0.341	0.395	0.466	0.588	
$R_{300,1500}$	0.220	0.309	0.370	0.466	0.599	0.710	0.806	1.021	
$R_{300,3000}$	0.398	0.517	0.671	0.787	1.075	1.496	1.567	1.814	
$R_{500,2500}$	0.398	0.517	0.671	0.787	1.075	1.496	1.567	1.814	
$R_{500,5000}$	0.588	0.891	1.276	1.540	2.055	2.447	5.342	19.563	
Average	0.333	0.429	0.566	0.681	0.906	1.119	1.693	5.014	1.364
	K								
$G_{p,q}$	3	4	5	6	7	8	9	10	
$G_{3,12}$	0.016	0.022	0.022	0.027	0.029	0.036	0.039	0.092	
$G_{4,36}$	0.074	0.112	0.153	0.188	0.231	0.276	0.357	0.377	
$G_{6,6}$	0.017	0.016	0.024	0.025	0.116	0.030	0.083	0.036	
$G_{12,12}$	0.063	0.065	0.204	0.130	0.558	0.277	0.561	1.111	
Average	0.043	0.054	0.101	0.093	0.233	0.155	0.260	0.404	1.364

TABLE 23. Average `AvDi` and `MiDi` of `MARA` in random networks

`AvDi`	K								
$R_{n,m}$	3	4	5	6	7	8	9	10	
$R_{100,500}$	0.867	0.911	0.904	0.900	0.885	0.873	0.864	0.862	
$R_{100,1000}$	–	0.970	0.962	0.962	0.969	0.969	0.969	0.965	
$R_{300,1500}$	0.929	0.951	0.949	0.951	0.944	0.941	0.935	0.933	
$R_{300,3000}$	0.966	0.977	0.976	0.979	0.976	0.976	0.972	0.973	
$R_{500,2500}$	0.960	0.967	0.961	0.963	0.955	0.955	0.950	0.947	
$R_{500,5000}$	0.922	0.948	0.948	0.960	0.957	0.958	0.966	0.961	
Average	0.929	0.954	0.950	0.952	0.948	0.945	0.943	0.940	0.945

`MiDi`	K								
$R_{n,m}$	3	4	5	6	7	8	9	10	
$R_{100,500}$	0.794	0.752	0.718	0.671	0.636	0.582	0.540	0.497	
$R_{100,1000}$	–	0.819	0.731	0.719	0.740	0.706	0.710	0.671	
$R_{300,1500}$	0.832	0.819	0.793	0.770	0.763	0.721	0.714	0.697	
$R_{300,3000}$	0.899	0.883	0.833	0.822	0.785	0.752	0.707	0.705	
$R_{500,2500}$	0.905	0.867	0.845	0.831	0.793	0.759	0.756	0.723	
$R_{500,5000}$	0.867	0.848	0.803	0.836	0.802	0.764	0.796	0.755	
Average	0.859	0.831	0.787	0.775	0.753	0.714	0.714	0.675	0.760

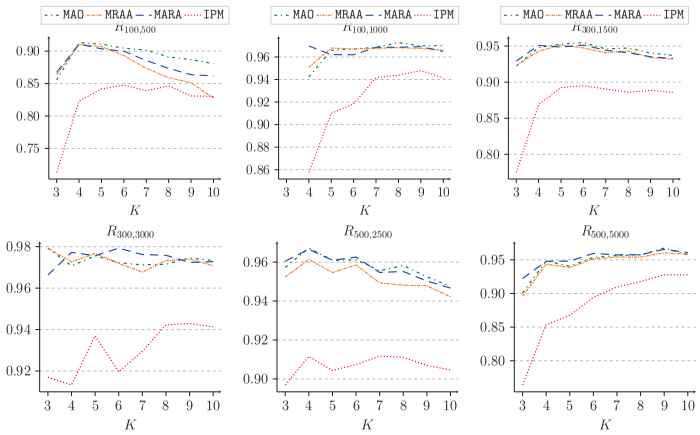
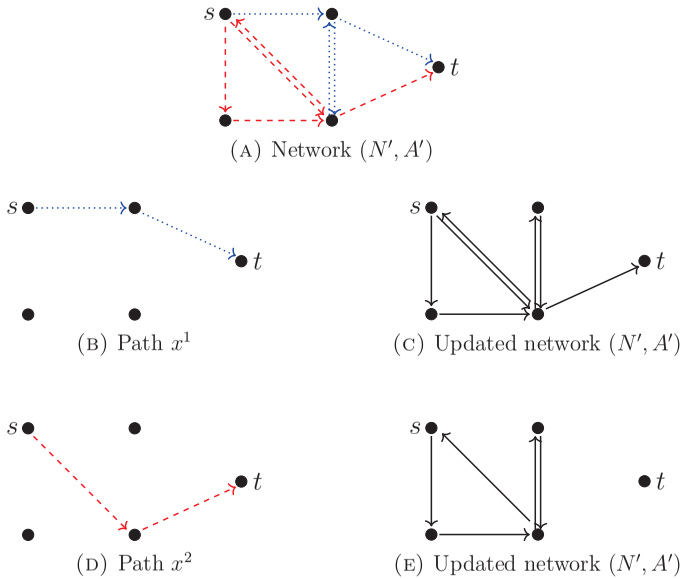


FIGURE 28. Average dissimilarity in random networks

– (2e), the variables z_{ij}^{kl} are binary as well and, consequently, by (2f), the variables v_{ij} are implicitly defined as non-negative integers, for any $(i,j) \in A$, $k \in \{1,\dots,K-1\}$, $l \in \{k+1,\dots,K\}$. The objective function (2a) corresponds to $OL(\{p_1,p_2,\dots,p_K\})$, the total number of arcs that are shared by at least two paths from s to t.



(A) Network (N', A')

(B) Path x^1

(C) Updated network (N', A')

(D) Path x^2

(E) Updated network (N', A')

FIGURE 2. Application of Algorithm 1 to a solution with loops

Formulation (2) may admit optimal solutions that contain subtours. For instance, the network depicted in Figure 2a shows a possible solution for finding $K = 2$ paths from node s to node t. In this case, the solution has objective value 0, given that the paths represented in red and in blue do not have any arcs in common. However, each of those paths contains one subtour. Nevertheless, if the problem is feasible, there always exists a loopless optimal solution to the problem, and this is easy to compute after a first optimal solution has been found. Solutions with loops can be avoided by adding subtour elimination constraints to the formulation or by adding a term to the objective function that penalizes the utilization of arcs. Another alternative is to apply a post processing algorithm that allows to extract one loopless optimal solution from a given optimal solution. Algorithm 1 outlines the procedure to obtain such a loopless solution, when given a solution x.

Algorithm 1: Algorithm for removing loops from a given solution x

1 $N' \leftarrow \{i \in N : x_{ij}^k = 1 \text{ for some } j \in N \wedge k = 1, \ldots, K\} \cup \{t\}$
2 $A' \leftarrow \{(i,j) \in A : x_{ij}^k = 1 \text{ for some } k = 1, \ldots, K\}$
3 **for** $(i,j) \in A'$ **do** $u_{ij} \leftarrow \sum_{k=1}^{K} x_{ij}^k$
4 **for** $k = 1, \ldots, K$ **do**
5 **for** $(i,j) \in A$ **do** $\bar{x}_{ij} \leftarrow 0$
6 $\bar{x}^k \leftarrow$ shortest path from s to t in terms of the number of arcs in network (N', A')
7 **for** $(i,j) \in A'$ *such that* $\bar{x}_{ij}^k = 1$ **do**
8 $u_{ij} \leftarrow u_{ij} - 1$
9 **if** $u_{ij} = 0$ **then** Delete arc (i,j) from A'

In lines 1 and 2 of Algorithm 1 the network corresponding to the arcs in the given solution x is built. Line 3 assigns each arc $(i,j) \in A'$ with the number of times it appears in x, u_{ij}. This value works like the arc (i,j)'s capacity. Then, in each iteration of the loop in lines 4 to 9, one path is determined and the capacity of an arc is updated every time it is used. The loop runs K times exactly, so that K paths are found. Moreover, the paths determined on line 5 are always loopless. These paths can be found by means of breadth-first search, [8], and therefore the algorithm runs in $O(Km)$ time. Also, the empirical tests reported later showed that its run time is very small when compared to solving any of the formulations here presented.

Proposition 4.1. *Let (x, z, v) be a feasible solution for formulation (2) and $\bar{x}$ be the corresponding vector output by Algorithm 1. Then, $\bar{x}$ defines K loopless paths from s to t.*

Proof: The remarks above show that the result holds. Shortly, every path defined by $\bar{x}^k$ is loopless, for any $k = 1, \ldots, K$, because it is the solution of a shortest path problem with all costs positive (unitary). Additionally, x is the characteristic vector of K paths from s to t, given that it satisfies the constraints (4b). Therefore, $\bar{x}^k$ defines K loopless paths from s to t. ■

Proposition 4.2. *Let (x, z, v) be an optimal solution for formulation (2). Let $\bar{x}$ be the vector output by Algorithm 1 when applied to x, and $\bar{z} \in \{0,1\}^{mK^2}$ and $\bar{v} \in \mathbb{N}_0^m$ be vectors which satisfy the constraints (2c) – (2f). Then, $(\bar{x}, \bar{z}, \bar{v})$ is a loopless optimal solution for problem (2).*

Appendix B. Dissimilarities and run times for the constrained formulations

TABLE 22. Average `AvDi` and `MiDi` of `MRAA` in random networks

`AvDi`	K								
$R_{n,m}$	3	4	5	6	7	8	9	10	
$R_{100,500}$	0.864	0.910	0.907	0.893	0.873	0.859	0.851	0.828	
$R_{100,1000}$	–	0.951	0.968	0.967	0.968	0.968	0.968	0.966	
$R_{300,1500}$	0.923	0.943	0.952	0.947	0.941	0.943	0.934	0.932	
$R_{300,3000}$	0.979	0.973	0.977	0.972	0.968	0.973	0.974	0.971	
$R_{500,2500}$	0.952	0.961	0.955	0.959	0.949	0.948	0.948	0.942	
$R_{500,5000}$	0.896	0.944	0.939	0.952	0.954	0.954	0.960	0.958	
Average	0.923	0.947	0.949	0.948	0.942	0.941	0.939	0.933	0.941

`MiDi`	K								
$R_{n,m}$	3	4	5	6	7	8	9	10	
$R_{100,500}$	0.774	0.742	0.723	0.669	0.625	0.582	0.515	0.498	
$R_{100,1000}$	–	0.704	0.754	0.743	0.720	0.680	0.675	0.670	
$R_{300,1500}$	0.811	0.790	0.799	0.766	0.747	0.730	0.702	0.701	
$R_{300,3000}$	0.938	0.855	0.837	0.738	0.719	0.709	0.728	0.707	
$R_{500,2500}$	0.888	0.860	0.819	0.790	0.769	0.743	0.739	0.680	
$R_{500,5000}$	0.798	0.841	0.810	0.802	0.803	0.766	0.771	0.761	
Average	0.842	0.799	0.790	0.752	0.731	0.702	0.688	0.670	0.745

TABLE 21. Average AvDi and MiDi of MAR in grid networks

AvDi	K								
$G_{p,q}$	3	4	5	6	7	8	9	10	
$G_{3,12}$	0.949	0.821	0.769	0.708	0.670	0.654	0.607	0.610	
$G_{4,36}$	0.982	0.982	0.892	0.814	0.759	0.690	0.643	0.652	
$G_{6,6}$	0.933	0.933	0.900	0.893	0.867	0.857	0.833	0.822	
$G_{12,12}$	0.970	0.970	0.955	0.952	0.939	0.935	0.924	0.919	
Average	0.959	0.926	0.879	0.842	0.809	0.784	0.752	0.751	0.839

MiDi	K								
$G_{p,q}$	3	4	5	6	7	8	9	10	
$G_{3,12}$	0.923	0.538	0.231	0.385	0.231	0.308	0.308	0.231	
$G_{4,36}$	0.947	0.974	0.237	0.132	0.158	0.132	0.158	0.132	
$G_{6,6}$	0.800	0.800	0.600	0.600	0.600	0.500	0.200	0.200	
$G_{12,12}$	0.955	0.909	0.864	0.864	0.864	0.818	0.727	0.727	
Average	0.906	0.805	0.533	0.495	0.463	0.439	0.348	0.322	0.539

Proof: Suppose $(\bar{x}, \bar{z}, \bar{v})$ is obtained from (x, z, v) according to Algorithm 1 and the directions above. Then, $x_{ij}^k, \bar{x}_{ij}^k \in \{0, 1\}$ and

$$x_{ij}^k \geq \bar{x}_{ij}^k, \quad (i,j) \in A, \quad k = 1, \ldots, K. \tag{3}$$

Two aspects need to be considered:

(1) According to Proposition 4.1, $\bar{x}$ corresponds to K paths, so the constraints (2b) hold. Moreover, $\bar{z}$ and $\bar{v}$ satisfy (2c) and (2f), therefore $(\bar{x}, \bar{z}, \bar{v})$ is a feasible solution of (2).

(2) Because of condition (3), it also holds that

$$z_{ij}^{kl} \geq \bar{z}_{ij}^{kl}, \quad (i,j) \in A, \quad k = 1, \ldots, K-1, \quad l = k+1, \ldots, K,$$

$$v_{ij} \geq \bar{v}_{ij}, \quad (i,j) \in A,$$

therefore $f_1(x, z, v) \geq f_1(\bar{x}, \bar{z}, \bar{v})$, which shows that the new solution is optimal.

It can then be concluded that $(\bar{x}, \bar{z}, \bar{v})$ is an optimal loopless solution of (2). ■

In the next section we describe some computational experiments performed on formulation (2). Results will show that formulation (2) outputs solutions with very good dissimilarity scores, indicating that problem (1) might be a good approach to the K dissimilar paths problem, even though it neglects the role of the length of the paths in the dissimilarity of the K paths. However, the high run times required to solve even problems of modest dimension compromise its usability in practical applications. These results are not at all unexpected, due to the combinatorial nature of the model, but they reinforce the need for recurring to alternative models, as mentioned in Section 1.

4.1. Computational experiments. The purpose of the tests presented in the following is to study the behavior of formulation (2) in terms of the solutions it outputs, its run times and its integer programming gaps.

The code designated by MAO, standing for the implementation for minimizing the number of arcs overlaps for each pair of paths, formulation (2), was implemented in C. This code uses IBM ILOG CPLEX version 12.7 as the ILP solver. The Algorithm 1, also coded in C, was applied to the result of this implementation, in order to remove the loops from the obtained solutions. The tests were carried out on a 64-bit PC with an Intel®Core™ i7-6700 Quad core at 3.40GHz with 64GB of RAM.

In the performed tests $K = 3, 4, \ldots, 10$ paths were computed in directed networks. Two types of instances were considered:

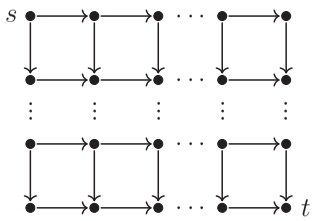


FIGURE 3. Grid network

- Random networks, denoted by $R_{n,m}$, with $n = 100, 300, 500$ nodes, and $m = dn$ arcs, for average degrees $d = 5, 10$. A Hamiltonian cycle is created for all the nodes in the network, and afterwards the remaining arcs are generated randomly. This Hamiltonian cycle is directed, therefore the strong connectivity of the graph is not fully ensured. There are no parallel arcs between pairs of nodes. For each size of these networks, 30 instances were generated based on different seeds. The reported results are based only on the 22, out of the 30, that originated feasible problems.
- Grid networks, denoted by $G_{p,q}$, with $p = 3, 4, 6, 12$ rows, $q = 6, 12, 36$ columns and $n = pq = 36, 144$ nodes, arranged in a planar grid, numbered consecutively from left to right and top to bottom – as shown in Figure 3. Any pair of adjacent nodes is connected by an arc, thus $m = 2pq - p - q = 57, 60, 248, 264$.

In both cases the source and the target nodes are $s = 1$ and $t = n$, respectively, with no loss of generality. It is worth noting that the considered grid networks are acyclic, therefore Algorithm 1 was not applied for such instances. Moreover, the run times of Algorithm 1 were small (at most 4% of the total run times, that is, up to 50 milliseconds), therefore they are not included in the results reported in the following.

We begin by observing that in a non negligible number of the random instances, K disjoint paths can be found. This happens for 274 of the 1056 instances. Table 1 exhibits the distribution of those instances.

TABLE 19. Average `AvDi` and `MiDi` of `MRA` in grid networks

`AvDi`	K								
$G_{p,q}$	3	4	5	6	7	8	9	10	
$G_{3,12}$	0.949	0.808	0.538	0.308	0.264	0.231	0.205	0.185	
$G_{4,36}$	0.982	0.982	0.718	0.584	0.560	0.570	0.414	0.506	
$G_{6,6}$	0.933	0.867	0.800	0.893	0.867	0.521	0.519	0.511	
$G_{12,12}$	0.970	0.970	0.955	0.952	0.939	0.929	0.924	0.758	
Average	0.959	0.907	0.753	0.684	0.657	0.563	0.516	0.490	0.691
`MiDi`	K								
$G_{p,q}$	3	4	5	6	7	8	9	10	
$G_{3,12}$	0.923	0.462	0.308	0.000	0.000	0.000	0.000	0.000	
$G_{4,36}$	0.974	0.947	0.132	0.158	0.132	0.000	0.000	0.000	
$G_{6,6}$	0.900	0.700	0.400	0.600	0.000	0.000	0.000	0.000	
$G_{12,12}$	0.955	0.955	0.909	0.818	0.773	0.727	0.773	0.318	
Average	0.938	0.766	0.437	0.419	0.376	0.182	0.193	0.080	0.424

TABLE 20. Average `AvDi` and `MiDi` of `MRO` in grid networks

`AvDi`	K								
$G_{p,q}$	3	4	5	6	7	8	9	10	
$G_{3,12}$	0.949	0.808	0.708	0.641	0.597	0.569	0.551	0.542	
$G_{4,36}$	0.982	0.982	0.892	0.804	0.729	0.650	0.621	0.581	
$G_{6,6}$	0.933	0.933	0.900	0.893	0.867	0.857	0.833	0.822	
$G_{12,12}$	0.970	0.970	0.955	0.952	0.939	0.935	0.924	0.919	
Average	0.959	0.927	0.864	0.822	0.783	0.753	0.732	0.716	0.819
`MiDi`	K								
$G_{p,q}$	3	4	5	6	7	8	9	10	
$G_{3,12}$	0.923	0.538	0.462	0.308	0.308	0.308	0.308	0.231	
$G_{4,36}$	0.947	0.974	0.132	0.132	0.132	0.158	0.132	0.132	
$G_{6,6}$	0.900	0.800	0.700	0.700	0.600	0.400	0.200	0.200	
$G_{12,12}$	0.909	0.909	0.864	0.864	0.818	0.773	0.727	0.773	
Average	0.920	0.805	0.539	0.501	0.464	0.410	0.342	0.334	0.539

TABLE 17. Average AvDi and MiDi of MAR in random networks

AvDi	K								
$R_{n,m}$	3	4	5	6	7	8	9	10	
$R_{100,500}$	0.876	0.903	0.895	0.878	0.871	0.858	0.849	0.845	
$R_{100,1000}$	–	0.962	0.957	0.958	0.964	0.957	0.955	0.955	
$R_{300,1500}$	0.950	0.949	0.950	0.940	0.936	0.930	0.920	0.913	
$R_{300,3000}$	0.977	0.969	0.970	0.973	0.968	0.965	0.965	0.960	
$R_{500,2500}$	0.956	0.962	0.957	0.954	0.951	0.945	0.942	0.934	
$R_{500,5000}$	0.925	0.948	0.938	0.949	0.954	0.948	0.957	0.952	
Average	0.937	0.949	0.945	0.942	0.941	0.934	0.931	0.926	0.938
MiDi	K								
$R_{n,m}$	3	4	5	6	7	8	9	10	
$R_{100,500}$	0.820	0.751	0.720	0.657	0.635	0.567	0.537	0.503	
$R_{100,1000}$	–	0.773	0.739	0.725	0.727	0.717	0.732	0.685	
$R_{300,1500}$	0.895	0.827	0.818	0.780	0.762	0.722	0.713	0.678	
$R_{300,3000}$	0.930	0.868	0.833	0.842	0.808	0.748	0.730	0.712	
$R_{500,2500}$	0.891	0.869	0.841	0.836	0.806	0.774	0.779	0.716	
$R_{500,5000}$	0.875	0.866	0.820	0.813	0.806	0.777	0.809	0.754	
Average	0.882	0.825	0.795	0.775	0.757	0.717	0.717	0.675	0.766

TABLE 18. Average AvDi and MiDi of MAO in grid networks

AvDi	K								
$G_{p,q}$	3	4	5	6	7	8	9	10	
$G_{3,12}$	0.949	0.833	0.800	0.787	0.762	0.745	0.741	0.730	
$G_{4,36}$	0.982	0.982	0.892	0.859	0.848	0.845	0.819	0.811	
$G_{6,6}$	0.933	0.933	0.900	0.893	0.867	0.857	0.844	0.840	
$G_{12,12}$	0.970	0.970	0.955	0.952	0.939	0.935	0.929	0.927	
Average	0.959	0.930	0.887	0.874	0.854	0.854	0.833	0.827	0.875
MiDi	K								
$G_{p,q}$	3	4	5	6	7	8	9	10	
$G_{3,12}$	0.846	0.538	0.231	0.154	0.154	0.154	0.000	0.000	
$G_{4,36}$	0.974	0.947	0.526	0.158	0.105	0.105	0.105	0.053	
$G_{6,6}$	0.900	0.900	0.700	0.700	0.600	0.500	0.500	0.500	
$G_{12,12}$	0.955	0.909	0.909	0.864	0.773	0.818	0.818	0.727	
Average	0.919	0.824	0.592	0.469	0.408	0.394	0.356	0.320	0.535

TABLE 1. Number of instances with K disjoint paths in random networks

	K							
$R_{n,m}$	3	4	5	6	7	8	9	10
$R_{100,500}$	12	2	1	0	0	0	0	0
$R_{100,1000}$	22	18	14	12	7	4	2	1
$R_{300,1500}$	14	7	1	0	0	0	0	0
$R_{300,3000}$	21	18	13	9	7	4	2	1
$R_{500,2500}$	14	4	3	1	1	0	0	0
$R_{500,5000}$	17	14	12	8	5	3	0	0

Unless otherwise stated, hereafter only the instances with no disjoint solutions are considered. There are two reasons for this:

- the main goal of this work is to find good methods to obtain dissimilar paths in networks where finding those paths is not easy (finding disjoint paths can be formulated as a minimum cost flow problem [29, 30]);
- such instances are not evenly distributed throughout the set of instances, thus their presence in different numbers in each group could skew the results.

The drawback of this decision is that each set $R_{n,m}$ now has a smaller and different number of instances. Table 2 indicates the final number for each type of instances. It should be remarked that there are no disjoint solutions in the grid instances.

TABLE 2. Final number of random network instances

	K							
$R_{n,m}$	3	4	5	6	7	8	9	10
$R_{100,500}$	10	20	21	22	22	22	22	22
$R_{100,1000}$	0	4	8	10	15	18	20	21
$R_{300,1500}$	8	15	21	22	22	22	22	22
$R_{300,3000}$	1	4	9	13	15	18	20	21
$R_{500,2500}$	8	18	19	21	21	22	22	22
$R_{500,5000}$	5	8	10	14	17	19	22	22

An upper bound of 300 seconds was set for the elapsed time when running the solver. The results presented in the following for random networks are average values for solving the instances with the same characteristics except for a seed, and with no disjoint solutions. The results shown for grid networks are based on a single instance, again with no disjoint solutions.

Approximately 84% of the random instances were solved to optimality (corresponding to 659 out of 782 instances) and 44% of the grid instances (corresponding to 14 out of 32 instances).

TABLE 3. Number of instances solved to optimality by `MAO` (%)

	K							
$R_{n,m}$	3	4	5	6	7	8	9	10
$R_{100,500}$	100	100	100	96	77	50	27	23
$R_{100,1000}$	–	100	100	100	100	100	100	95
$R_{300,1500}$	100	100	100	100	95	80	59	41
$R_{300,3000}$	100	100	100	100	100	100	100	86
$R_{500,2500}$	100	100	100	100	86	73	55	32
$R_{500,5000}$	100	100	100	100	100	100	86	82

	K							
$G_{p,q}$	3	4	5	6	7	8	9	10
$G_{3,12}$	100	100	100	100	0	0	0	0
$G_{4,36}$	100	100	0	0	0	0	0	0
$G_{6,6}$	100	100	100	100	0	0	0	0
$G_{12,12}$	100	100	100	100	0	0	0	0

Due to the combinatorial nature of the model, it would be expected that the results of its application depended heavily on the size of the network. However, according to Tables 3* and 4 many of the instances that were not solved within the time limit are actually associated to the smaller networks. In fact, the results behaved as expected only for the first values of $K \leq 4$, for both the random and the grid networks. As K increases, other features seem to have a more decisive influence. A finer analysis of the results allows to enumerate three possible explanations for that situation: the layout of the network; the sparsity of the network; and the relation between the number of paths K and the size of the network.

The differences in the layout of the network account for the fact that grid instances are, in general, harder to solve than random instances, even though

*There is only one grid instance of each kind, therefore, the listed values are either 0% or 100%.

TABLE 16. Average `AvDi` and `MiDi` of `MRO` in random networks

`AvDi`	K								
$R_{n,m}$	3	4	5	6	7	8	9	10	
$R_{100,500}$	0.866	0.906	0.885	0.865	0.849	0.828	0.815	0.808	
$R_{100,1000}$	–	0.978	0.957	0.944	0.951	0.943	0.932	0.931	
$R_{300,1500}$	0.931	0.947	0.938	0.929	0.916	0.912	0.907	0.904	
$R_{300,3000}$	0.963	0.974	0.971	0.968	0.956	0.951	0.941	0.931	
$R_{500,2500}$	0.942	0.965	0.952	0.942	0.932	0.929	0.929	0.923	
$R_{500,5000}$	0.919	0.941	0.935	0.950	0.948	0.944	0.942	0.938	
Average	0.924	0.952	0.940	0.933	0.925	0.918	0.911	0.906	0.926

`MiDi`	K								
$R_{n,m}$	3	4	5	6	7	8	9	10	
$R_{100,500}$	0.763	0.758	0.719	0.674	0.632	0.577	0.533	0.525	
$R_{100,1000}$	–	0.869	0.757	0.749	0.755	0.711	0.703	0.710	
$R_{300,1500}$	0.835	0.836	0.792	0.789	0.746	0.740	0.707	0.687	
$R_{300,3000}$	0.889	0.899	0.858	0.831	0.759	0.763	0.736	0.707	
$R_{500,2500}$	0.851	0.893	0.862	0.837	0.814	0.787	0.780	0.710	
$R_{500,5000}$	0.841	0.849	0.832	0.833	0.805	0.809	0.777	0.793	
Average	0.836	0.851	0.803	0.785	0.752	0.731	0.706	0.689	0.768

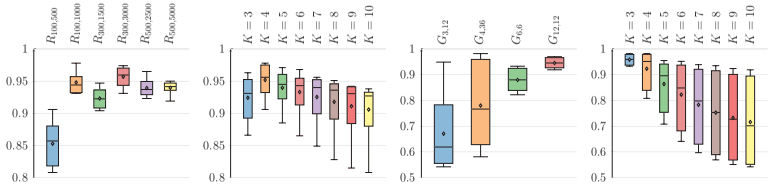


FIGURE 26. `AvDi` dispersion of `MRO`

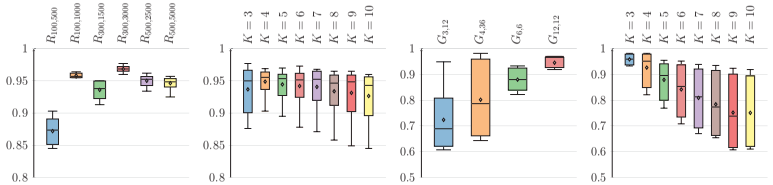


FIGURE 27. `AvDi` dispersion of `MAR`

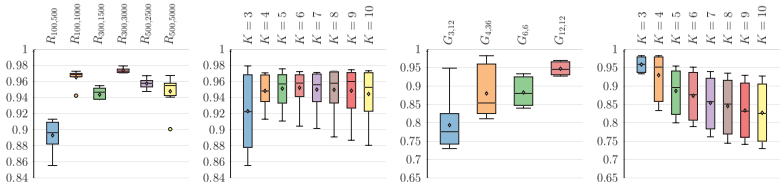


FIGURE 24. `AvDi` dispersion of `MAO`

TABLE 15. Average `AvDi` and `MiDi` of `MRA` in random networks

`AvDi`	K								
$R_{n,m}$	3	4	5	6	7	8	9	10	
$R_{100,500}$	0.852	0.895	0.885	0.862	0.826	0.803	0.794	0.741	
$R_{100,1000}$	–	0.933	0.951	0.939	0.944	0.940	0.937	0.877	
$R_{300,1500}$	0.915	0.942	0.933	0.925	0.913	0.904	0.901	0.884	
$R_{300,3000}$	0.979	0.924	0.971	0.886	0.891	0.892	0.897	0.896	
$R_{500,2500}$	0.952	0.950	0.945	0.935	0.929	0.907	0.904	0.889	
$R_{500,5000}$	0.879	0.929	0.927	0.938	0.940	0.934	0.933	0.922	
Average	0.915	0.929	0.935	0.914	0.907	0.897	0.894	0.868	0.909

`MiDi`	K								
$R_{n,m}$	3	4	5	6	7	8	9	10	
$R_{100,500}$	0.772	0.735	0.720	0.666	0.603	0.532	0.513	0.416	
$R_{100,1000}$	–	0.726	0.746	0.753	0.743	0.727	0.723	0.652	
$R_{300,1500}$	0.817	0.824	0.815	0.766	0.750	0.729	0.695	0.642	
$R_{300,3000}$	0.938	0.833	0.850	0.744	0.741	0.704	0.722	0.710	
$R_{500,2500}$	0.888	0.866	0.848	0.803	0.804	0.742	0.738	0.704	
$R_{500,5000}$	0.798	0.831	0.811	0.806	0.795	0.791	0.766	0.734	
Average	0.843	0.802	0.798	0.756	0.739	0.704	0.693	0.643	0.751

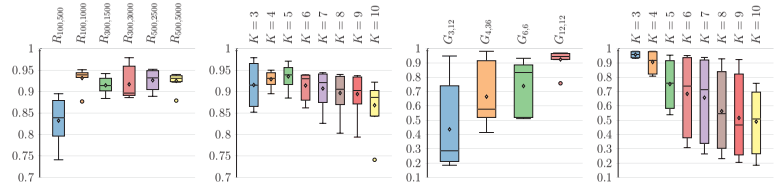


FIGURE 25. `AvDi` dispersion of `MRA`

they are much smaller (recall that $36 \leq n \leq 144$ and $57 \leq m \leq 264$ for the grid instances, whereas $100 \leq n \leq 500$ and $500 \leq m \leq 5000$ for the random instances). Furthermore, even among grid networks, there are differences related to the layout, as the problem seems to be more difficult to solve in rectangular grids rather than in square grids – see Table 4 and Figure 4.

TABLE 4. Run times of `MAO` (seconds)

	K								
$R_{n,m}$	3	4	5	6	7	8	9	10	
$R_{100,500}$	0.124	0.490	1.135	32.149	76.488	175.405	222.937	249.626	
$R_{100,1000}$	–	0.854	3.245	2.910	5.067	10.116	18.692	48.099	
$R_{300,1500}$	0.391	1.611	2.637	10.127	37.499	111.854	174.106	212.429	
$R_{300,3000}$	0.745	1.934	5.063	11.508	23.537	35.912	63.022	132.342	
$R_{500,2500}$	0.793	2.179	4.579	12.781	66.327	121.559	196.307	238.856	
$R_{500,5000}$	1.888	1.491	7.253	17.309	34.013	57.627	118.488	196.436	
Average	0.788	1.426	3.985	14.464	40.488	85.412	132.258	179.631	57.306

	K								
$G_{p,q}$	3	4	5	6	7	8	9	10	
$G_{3,12}$	0.072	2.665	10.150	52.917	300	300	300	300	
$G_{4,36}$	0.218	0.684	300	300	300	300	300	300	
$G_{6,6}$	0.096	0.265	2.776	11.783	300	300	300	300	
$G_{12,12}$	0.240	0.464	5.212	19.772	300	300	300	300	
Average	0.156	1.019	79.534	96.118	300	300	300	300	172.103

Moreover, the effect of the sparseness of the network is particularly clear in the results of the random instances. According to Figure 4, the run times decrease as the density of the network† increases when $K \geq 5$, in all sets of instances but one, and that this factor overcomes the one of the size of the network. The only exception is the smallest of the instances. In spite of its density, second only to $R_{100,1000}$, $R_{100,500}$ instances have proven to be, in average, the hardest of the random instances to solve. Another interesting conclusion is that the impact of sparseness in grid networks is not so significant as it is in random networks. This finding becomes evident when comparing the run times for the $G_{4,36}$ and $G_{12,12}$ networks, which have the lowest of the densities of the grid networks.

†The density of a network is defined as $m/(n(n-1))$.

As for the third of the reasons presented: the proportion between the value of K and the size of the network, finding K dissimilar paths in a given network becomes more difficult as the value of K grows and this difficulty is increased in very small networks. This is a plausible reason why the results obtained for the $R_{100,500}$ cases do not follow the pattern of the remaining random instances. Moreover, it is also expected that this factor has an impact on the results for the grid instances. However, results for more grid networks would be recommended for a sound conclusion.

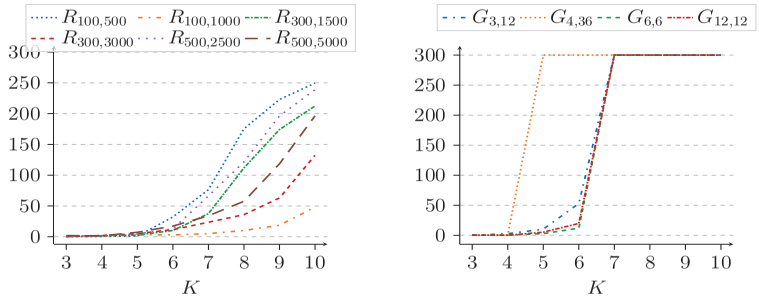


FIGURE 4. Run times of `MAO` (seconds)

For the sake of completeness, the lower bounds obtained by solving the linear programming relaxation of the model are also presented. Table 5 presents the average integer programming gaps, as well as the run times for solving the corresponding linear programming relaxations, for each group of instances. The integer programming gaps are computed as $100(f_1^* - f_{LR_1}^*)/|f_1^*|\%$, where f_1^* denotes the optimum value of (2) and $f_{LR_1}^*$ denotes the optimum value of its linear programming relaxation. Whenever the optimum value f_1^* is unknown, the best known integer is used to compute these gaps. The gaps associated with the grid instances were all 100%, therefore no such table is presented.

each pair of paths, given that this is more in accordance with the dissimilarity measure that was used for assessing the solutions.

Another important line of research, would be to clarify the relation between the dissimilarity and run time results of the formulations and some influencing characteristics of the networks, such as the density, average degree and topology. This aspect would allow a better use of the new models, particularly in the context of real world applications of the problem, for, in those cases, a wider variety of networks can arise. Finally, also the extension of the introduced formulations to bi-objective problems involving the minimization of a cost function besides the minimization of the arc repetitions seems to be of interest.

Appendix A. Dissimilarities for the unconstrained formulations

TABLE 14. Average `AvDi` and `MiDi` of `MAO` in random networks

`AvDi`	K								
$R_{n,m}$	3	4	5	6	7	8	9	10	
$R_{100,500}$	0.855	0.913	0.911	0.904	0.902	0.891	0.887	0.886	
$R_{100,1000}$	–	0.942	0.965	0.967	0.968	0.973	0.970	0.970	
$R_{300,1500}$	0.922	0.947	0.952	0.954	0.946	0.947	0.940	0.937	
$R_{300,3000}$	0.979	0.970	0.976	0.972	0.971	0.972	0.975	0.973	
$R_{500,2500}$	0.957	0.967	0.961	0.961	0.956	0.958	0.952	0.947	
$R_{500,5000}$	0.900	0.948	0.940	0.953	0.956	0.957	0.967	0.958	
Average	0.923	0.948	0.951	0.951	0.952	0.950	0.950	0.950	0.944

`MiDi`	K								
$R_{n,m}$	3	4	5	6	7	8	9	10	
$R_{100,500}$	0.855	0.763	0.720	0.691	0.650	0.615	0.555	0.547	
$R_{100,1000}$	–	0.654	0.750	0.740	0.700	0.698	0.682	0.684	
$R_{300,1500}$	0.822	0.813	0.801	0.782	0.763	0.745	0.704	0.705	
$R_{300,3000}$	0.938	0.850	0.829	0.751	0.723	0.691	0.721	0.707	
$R_{500,2500}$	0.887	0.882	0.822	0.814	0.774	0.792	0.730	0.720	
$R_{500,5000}$	0.819	0.862	0.819	0.794	0.778	0.760	0.780	0.756	
Average	0.864	0.804	0.790	0.761	0.731	0.731	0.717	0.696	0.687

of network. However, comparing to MARA, we realize that MAR offers worse dissimilarities in the same amount of time.

9. Concluding remarks

This manuscript addresses the search for K dissimilar paths connecting a given pair of nodes in a directed graph. Four integer formulations were introduced. The formulations have different motivations, but their general goal is to minimize the number of arcs that appear in more than one path or the total number of those overlaps, while searching for sets of paths with good dissimilarity. The inclusion of an additional set of constraints to the previous formulations, with the goal of improving the solutions dissimilarity, was also proposed. The performance of the new formulations and of a traditional method in the literature, the iterative penalty method, was tested over random and grid networks, assessing the required run time as well as the average and the minimum dissimilarities of the solutions. The dissimilarity of a given solution is the average of the pairwise dissimilarity between their paths, which is measured based on the ratios between the number of overlaps of the pair of paths and the length of the involved paths. Three classes can be considered with this respect:

- The code IPM, which not surprisingly is the fastest, given that it simply solves K shortest path problems. Besides being the fastest code, IPM provided solutions with good dissimilarity for the grid networks. On the other hand, the solutions produced by this method for the random networks are very poor as to the dissimilarity of the K paths.
- The code MAO, clearly the slowest of the five codes and often unable of finding an optimal solution within the established 300 seconds. In spite of this drawback, in general, MAO produced the solutions with the best dissimilarity, both for the random and the grid networks.
- And the final group, composed of codes MAR, MRAA and MARA, which also provide solutions of good quality with regard to the dissimilarity. The three codes have similar behaviors for the random networks, however, for the grid networks, the latter unequivocally outperforms the others. Unlike MRAA, both MAR and MARA were able to find the optimum within the time bound for all instances. Furthermore, the run times of MAR and MARA are much smaller than the run times of the remaining codes in this group.

It is worth noting that, the introduced formulations can still be extended to a case where the overlaps are penalized according to a given cost for the involved arcs, with no significant difference of the implementation.

In the future it would be interesting to investigate a simpler/smaller alternative formulation to MAO, accounting for the number of overlaps between

TABLE 5. Average integer programming gaps of $\texttt{MAO}_L$ (%)

	K							
$R_{n,m}$	3	4	5	6	7	8	9	10
$R_{100,500}$	43	73	75	77	78	78	79	79
$R_{100,1000}$	–	100	100	100	100	100	100	100
$R_{300,1500}$	75	89	92	93	93	93	93	93
$R_{300,3000}$	100	100	100	100	100	100	100	100
$R_{500,2500}$	91	96	96	97	97	97	97	97
$R_{500,5000}$	60	75	80	86	87	90	91	91

As it can be observed from Table 5, the linear programming relaxation of formulation (2) produces very weak lower bounds. This often happens in models that use the same type of linking constraints (2e) used in this formulation.

5. Minimization of the number of repeated arcs

While taking into account all the overlaps in the pairs of paths in the solutions, the formulation (2) is not easy to handle from a practical point of view, as shown by the experiments reported in Section 4.1. In the following an alternative, and simpler, approach to problem (1) is introduced.

A given arc $(i, j) \in A$ is said to be repeated in a set of K paths if it belongs to more than one of them. A way to ensure that the K paths are sufficiently different from each other is to consider a relaxed version of the K disjoint paths problem, where instead of forbidding the occurrence of repeated arcs, the number of arcs in those conditions is minimized. This concept was explored in the example illustrated in Figure 1, given in the introductory section. Because this approach simply privileges the number of repeated paths, it favors the solution depicted in Figure 1a, rather than the one in Figure 1b.

In order to model such a problem, let us consider, as before, the decision variables x_{ij}^k equal to 1 if the arc (i, j) lies in the k-th path from s to t, or 0 otherwise, for any $(i, j) \in A$ and $k = 1, \ldots, K$. The problem of minimizing

the number of repeated arcs can then be formulated as follows:

$$\min \quad f_2(x,y) = \sum_{(i,j)\in A} y_{ij} \tag{4a}$$

$$\text{subject to} \quad \sum_{j\in N:(i,j)\in A} x_{ij}^k - \sum_{j\in N:(j,i)\in A} x_{ji}^k = \begin{cases} 1 & i = s \\ 0 & i \neq s,t \\ -1 & i = t \end{cases}, k = 1,\ldots,K \tag{4b}$$

$$y_{ij} \leq \sum_{k=1}^{K} x_{ij}^k, \quad (i,j) \in A \tag{4c}$$

$$(K-1)y_{ij} \geq \sum_{k=1}^{K} x_{ij}^k - 1, \quad (i,j) \in A \tag{4d}$$

$$x_{ij}^k \in \{0,1\},\ y_{ij} \in \{0,1\}, \quad (i,j) \in A, \quad k = 1,\ldots,K \tag{4e}$$

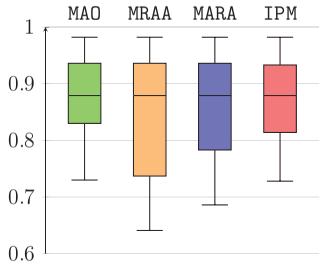
This formulation has $O(Km)$ binary variables and $O(Kn+m)$ constraints. The constraints (4b) are flow conservation constraints that model K paths from node s to node t. The constraints (4c) and (4d) relate the x and the y variables, in a way that y_{ij} is 1 if and only if the arc (i,j) is used in more than one path, that is, if this arc is repeated, and 0 otherwise. In fact, given the arc $(i,j) \in A$:

(1) If $x_{ij}^k = 0$ for every $k = 1,\ldots,K$, then by (4c) we have $y_{ij} = 0$, whereas (4d) has no implications on the value of y_{ij}.
(2) If $x_{ij}^k = 1$ for exactly one $k \in \{1,\ldots,K\}$, then neither (4c) nor (4d) have implications on the value of y_{ij}.
(3) If $x_{ij}^k = 1$ for more than one $k \in \{1,\ldots,K\}$, then (4c) has no implications on the value of y_{ij}, whereas by (4d) we have $y_{ij} = 1$.

When in situation 2., that is, if only one variable x_{ij}^k has value 1, the value of y_{ij} can be arbitrary. However, the objective function minimizes the sum of all these variables and this minimization is achieved if the arbitrary y_{ij}'s are equal to 0, as intended. Therefore, the objective function counts the number of repeated arcs, that is the number of arcs that are used more than once.

The same reasoning can be used to prove that the set of constraints (4c) can be dropped. In fact, because the goal of this formulation is to minimize $\sum_A y_{ij}$, the values of y_{ij} are 0 by default.

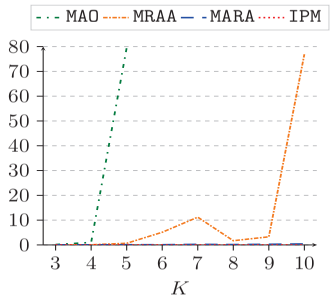
Just like for the formulation presented in the previous section, an optimal solution of formulation (4) may contain loops, as long as they do not include any arc that is common to several paths. However, given any such optimal solution, Algorithm 1 can be applied in order to remove its loops. The



	MAO	MRAA	MARA	IPM
Min. value	0.730	0.641	0.686	0.728
1^{st} quartile	0.830	0.737	0.783	0.814
Median	0.879	0.879	0.879	0.879
3^{rd} quartile	0.936	0.936	0.936	0.933
Max. value	0.982	0.982	0.982	0.982

FIGURE 22. `AvDi` dispersion of the constrained formulations in grid networks

for this type of network. Nevertheless, the best results are, again, associated to `MAO` and `IPM`.

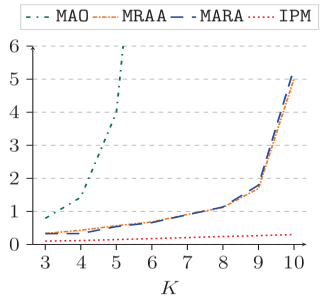


	K							
	3	4	5	6	7	8	9	10
MAO	0.158	1.019	79.534	96.118	300	300	300	300
MRAA	0.055	0.059	0.667	5.111	11.194	1.702	3.270	76.915
MARA	0.043	0.054	0.101	0.093	0.233	0.155	0.260	0.404
IPM	0.004	0.006	0.007	0.007	0.010	0.011	0.014	0.015

FIGURE 23. Run times for the constrained formulations in grid networks (seconds)

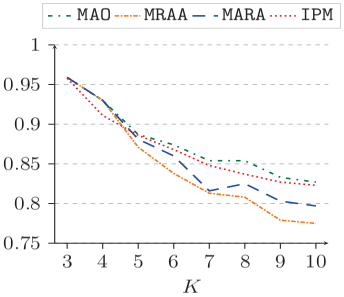
According to Figure 23, `MRAA` is much slower than `MARA` for this type of instances, which is, in turn, much slower than `IPM`. Thus, even though `MARA` has good run times it is clearly overtaken by `IPM`. As mentioned earlier, in general, the run times associated to `MAO` are very high. Tables 24 and 25 in Appendix B give detailed run times of the `MRAA` and `MARA` models.

The previous discussion confirms `IPM` as the most suitable method for solving instances in grid networks. `IPM` is the fastest of the codes being compared and produced the solutions with the highest overall average dissimilarity. On the other hand, the results associated to `MARA` are also to be considered. In average the difference towards `MAO` is 2%. This, together with the very reasonable run times, makes `MARA` also interesting for this type of network. In Section 8.1 we also pointed out `MAR` as an interesting option for this type



	K							
	3	4	5	6	7	8	9	10
MAO	0.079	1.426	3.985	14.464	40.488	85.412	132.258	179.631
MRAA	0.327	0.329	0.540	0.659	0.898	1.133	1.794	5.320
MARA	0.333	0.429	0.566	0.681	0.906	1.119	1.693	5.014
IPM	0.100	0.118	0.147	0.176	0.207	0.240	0.266	0.300

FIGURE 20. Run times of the constrained formulations in random networks (seconds)



	K							
	3	4	5	6	7	8	9	10
MAO	0.959	0.930	0.887	0.874	0.854	0.845	0.833	0.827
MRAA	0.959	0.930	0.871	0.838	0.813	0.808	0.779	0.775
MARA	0.959	0.930	0.881	0.860	0.816	0.825	0.803	0.797
IPM	0.958	0.911	0.886	0.868	0.848	0.837	0.827	0.823

FIGURE 21. Average AvDi values of the constrained formulations in grid networks

earlier good dissimilarities and run times of MAR, it not is clear whether its constrained model is the best option.

Figure 21 presents the average AvDi for MRAA, MARA, IPM and MOA in grid networks. In spite of the significant improvement for the pair MRA–MRAA (see Figure 14), MARA still produced solutions with higher average dissimilarities. In addition, it solved all the instances to optimality, which did not happen for MRAA (it was not possible to find 10 paths in $G_{4,36}$ within 300 seconds). Nevertheless, MAO provides the solutions with the best average dissimilarities, now followed closely by IPM. On the other hand, there are reductions in the dispersion of the results for the pairs MRA–MRAA and MAR–MARA, when comparing Figures 16 and 22. Furthermore, as the most significant improvements occurred in the lower quartiles, it can be concluded that both MRAA and MARA are less likely to produce solutions with low dissimilarities than MRA and MAR

arguments for proving Proposition 5.1 are similar to those used for Proposition 4.2, therefore its proof is omitted.

Proposition 5.1. *Let (x, y) be an optimal solution for problem (4). Let $\bar{x} \in \{0,1\}^{Km}$ be the corresponding vector output by Algorithm 1 when applied to x, and $\bar{y} \in \{0,1\}^m$ be such that the constraints (4c) and (4d) are satisfied. Then, $(\bar{x}, \bar{y})$ is a loopless optimal solution for problem (4).*

5.1. Computational experiments. In the following, formulation (4), which minimizes the number of repeated arcs, is analyzed empirically. The code that implements this formulation is designated by MRA. It was written in C and uses IBM ILOG CPLEX version 12.7 to solve the integer programs. The variant of the same formulation obtained by removing the constraints (4c) was also implemented. Note that both models are valid formulations of the problem, the later being a weaker (in terms of its linear programming bound) but smaller variant of the first. Because the differences in the run times obtained with both variants were not significant, only the original one is presented below.

The experimental setup was as described in Section 4.1. Algorithm 1 was applied to the results of the code MRA to remove the loops from the obtained solutions in the random instances.

TABLE 6. Number of instances solved to optimality by MRA (%)

	K							
$G_{p,q}$	3	4	5	6	7	8	9	10
$G_{3,12}$	100	100	100	100	100	100	100	100
$G_{4,36}$	100	100	100	100	100	0	0	0
$G_{6,6}$	100	100	100	100	100	100	100	100
$G_{12,12}$	100	100	100	100	100	0	0	0

MRA was able to solve to optimality all the random network instances, within the 300 seconds time window. Additionally, Table 6 shows the same number but for grids. It was possible to solve 81% of these instances, corresponding to 26 out of the 32 instances. The interruptions after 300 seconds occurred in problems of finding 8 or more paths in the 144 node grids. Finally, it is worth noting that even though formulations (2) and (4) model different problems it is still of interest to compare them, since they both are relaxations of the K dissimilar paths problem, and that far more instances were solved by MRA than by MAO.

TABLE 7. Run times of MRA (seconds)

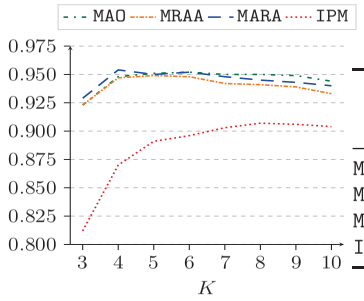
	K								
$R_{n,m}$	3	4	5	6	7	8	9	10	
$R_{100,500}$	0.051	0.061	0.098	0.173	0.254	0.657	0.981	2.092	
$R_{100,1000}$	–	0.108	0.145	0.189	0.206	0.351	0.373	0.385	
$R_{300,1500}$	0.138	0.204	0.279	0.312	0.432	0.539	1.057	2.144	
$R_{300,3000}$	0.280	0.347	0.411	0.485	0.614	0.723	0.855	0.949	
$R_{500,2500}$	0.256	0.323	0.457	0.557	0.723	0.964	1.458	1.586	
$R_{500,5000}$	0.394	0.542	0.770	0.944	1.074	1.271	1.488	1.836	
Average	0.223	0.261	0.360	0.443	0.550	0.750	1.053	1.498	0.640

	K								
$G_{p,q}$	3	4	5	6	7	8	9	10	
$G_{3,12}$	0.054	0.154	0.455	0.859	0.696	0.902	1.078	1.235	
$G_{4,36}$	0.103	0.107	1.833	5.047	221.466	300	300	300	
$G_{6,6}$	0.019	0.093	0.327	1.200	11.392	3.075	2.176	1.441	
$G_{12,12}$	0.166	0.239	1.070	4.359	47.101	300	300	300	
Average	0.085	0.148	0.921	2.866	70.163	150.994	150.813	150.669	65.832

The run times of code MRA are summarized in Table 7 and depicted in Figure 5. The results were particularly sensitive to the number of paths. However, other features have negative repercussions as well.

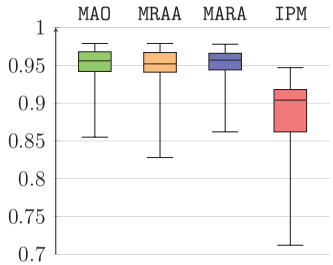
First, the results depend greatly on the layout of the network. In fact, unsolved instances were only found solo in the case of grids. Furthermore, the magnitude of the run times associated to the solved grid instances is several times higher than the run times for solving the random instances (up to 221.5 seconds in the former case and to 2.2 seconds in the latter). Another evidence of the difficulty associated to solving the grid instances, is the fact that between 4 and 256 arcs have to be repeated in these instances, against between 1 and 24 for random networks. In this way, both formulations (2) and (4) are very sensitive to the layout of the network and both work worse in the case of grid networks. In the case of the grid networks, the run times of MRA vary both with the shape of the grid and with K. Further conclusions would require more exhaustive tests.

Second, as made evident by analyzing Table 7 and Figure 5, the patter behaviour of the MRA run times for random networks changes for $K \geq 7$. Prior to that value, the run times vary with the size of the networks; afterwards other factors become dominant. This type of behavior was identified



	K							
	3	4	5	6	7	8	9	10
MAO	0.923	0.948	0.951	0.952	0.950	0.950	0.949	0.944
MRAA	0.923	0.947	0.949	0.948	0.942	0.941	0.939	0.933
MARA	0.929	0.954	0.950	0.952	0.948	0.945	0.943	0.940
IPM	0.812	0.870	0.891	0.896	0.903	0.907	0.906	0.904

FIGURE 18. Average AvDi values of the constrained formulations in random networks



	MAO	MRAA	MARA	IPM
Min. value	0.855	0.828	0.862	0.712
1^{st} quartile	0.942	0.941	0.944	0.862
Median	0.956	0.952	0.957	0.904
3^{rd} quartile	0.968	0.967	0.966	0.918
Max. value	0.979	0.979	0.978	0.947

FIGURE 19. AvDi dispersion of the constrained formulations in random networks

Comparing Figures 12 and 19 reveals a reduction in the dispersion of the results of both MARA and MRAA. Nevertheless, MARA outperforms MRAA and performs even better than MAO regarding the worst cases. Again, IPM produces the poorest results.

Figure 20 reports the average run times of the models. As shown in Section 7.1, adding the constraints (10) affected MAR more than MRA. In fact, MARA and MRAA are now very close – MRAA runs faster for the smaller networks and MARA for the larger. It is worth mentioning the increase of the run times from 1 to 5 seconds for the biggest instances. As expected, IPM is faster than the other codes.

In short, both MRAA and MARA seem tailored for this type of network: the average difference in the dissimilarities towards MAO is of 0.50% for MRAA and of 0.08% for MARA; as to the run times, MRAA is in advantage for $K \leq 7$ and MARA for the remaining values of K. However, taking into account the

were smaller both in the cases of the random and of the grid networks (up to 6% in the first case and to 24% in the later). There were no changes in the dissimilarities obtained on the square grids. Finally, the smallest difference was found for MAR and MARA. Even so, the improvements on the rectangular grids were quite significant. Again, no differences were registered in the dissimilarity results for square grids. Detailed information about the three models can be found in Appendix B.

TABLE 13. Average dissimilarity variation for MRAA, MROA and MARA (%)

	MRAA								MROA								MARA							
	K								K								K							
$R_{n,m}$	3	4	5	6	7	8	9	10	3	4	5	6	7	8	9	10	3	4	5	6	7	8	9	10
$R_{100,500}$	1	1	2	4	6	7	7	12	0	1	2	4	3	6	6	5	−1	1	1	2	2	2	2	2
$R_{100,1000}$	–	2	2	3	2	3	3	10	–	0	1	2	2	3	4	4	–	1	1	0	0	1	1	1
$R_{300,1500}$	1	0	2	2	3	4	4	5	0	0	2	3	3	3	3	3	−2	0	0	1	1	1	2	2
$R_{300,3000}$	0	5	1	10	9	9	9	8	0	−1	1	1	2	3	3	4	−1	1	1	1	1	1	1	1
$R_{500,2500}$	0	1	1	3	2	4	5	6	1	0	0	2	2	3	2	2	1	0	0	1	0	1	1	1
$R_{500,5000}$	2	2	1	1	2	2	3	4	1	1	1	0	1	1	2	2	0	0	1	1	0	1	1	1

	MRAA								MROA								MARA							
	K								K								K							
$G_{p,q}$	3	4	5	6	7	8	9	10	3	4	5	6	7	8	9	10	3	4	5	6	7	8	9	10
$G_{3,12}$	0	3	37	138	158	202	218	244	0	3	4	16	18	23	22	24	0	2	1	11	6	10	13	15
$G_{4,36}$	0	0	24	38	31	16	73	28	0	0	0	0	0	16	14	24	0	0	0	0	−2	14	19	15
$G_{6,6}$	0	8	13	0	0	64	60	61	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$G_{12,12}$	0	0	0	0	0	1	0	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

In the following, we analyze the trade-off between the improvements in the dissimilarities and the variations in the run time, identified in Section 7.1, to assess to what extent the constrained models are worth considering. The new models are compared to IPM and MAO. The code MROA was ruled out of the study, because it follows closely MARA but always with some disadvantage.

Figure 18 depicts the variation of the average AvDi for MRAA, MRAA and IPM in the random networks. In spite of the improvement on the average dissimilarities associated to the MRAA when compared to MRA (see Figures 10 and 18), the constrained version of MAR is still the best, in average. The differences, however, are now very tenuous.

in Section 4.1, when analyzing MAO results. Then, the density of the network and the proportion between the value of K and the size of the network were identified as determinant factors of the observed deviation. The corresponding MRA results, seem to be affected in a similar way – the density of the network smooths the growth of the run times, whereas the increase of K causes abrupt increases from a certain threshold for the $R_{100,500}$ and $R_{300,1500}$ instances.

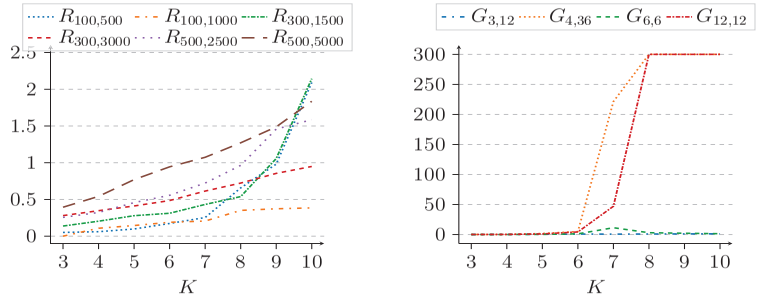


FIGURE 5. Run times of MRA (seconds)

TABLE 8. Average integer programming gaps of MRA_L (%)

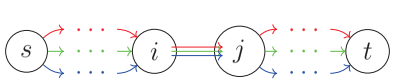
	K							
$R_{n,m}$	3	4	5	6	7	8	9	10
$R_{100,500}$	10	40	36	38	36	42	43	43
$R_{100,1000}$	–	12	23	25	40	46	48	49
$R_{300,1500}$	14	33	45	41	39	39	47	47
$R_{300,3000}$	2	11	25	35	38	44	49	47
$R_{500,2500}$	17	42	35	37	35	42	45	45
$R_{500,5000}$	7	14	17	30	38	41	48	42

	K							
$G_{p,q}$	3	4	5	6	7	8	9	10
$G_{3,12}$	50	61	50	46	41	37	35	32
$G_{4,36}$	50	67	72	59	51	46	41	37
$G_{6,6}$	50	67	67	70	70	66	60	56
$G_{12,12}$	50	67	67	70	70	71	71	72

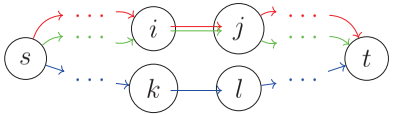
Finally, Table 8 presents the average integer programming gaps determined by the lower bound produced by the linear relaxation of the formulation (4). These gaps are computed as explained in Section 4.1. In the random instances the gap values are at most 49%. The gaps are even bigger for the grid networks, between 32% ad 72%.

6. Minimization of the number of arc repetitions

The goal of formulation (4), presented in the previous section, is to minimize the number of arcs which are repeated in the solutions. The undesired effect of this single objective may be that few arcs appear in many different paths. This situation is illustrated in Figure 1. Another example is depicted by the two sets of $K = 3$ paths in Figure 6. Both the solutions in Figure 6 have a single repeated arc. However, the paths in the solution in Figure 6b are more dissimilar than the paths in Figure 6a, the reason being that in the first case the arc (i, j), which is repeated, appears only twice, while it appears in all the three paths in the latter case. In the following two approaches are presented which intend to model a more complete understanding of how dissimilar paths should look like.



(A) Solution with one repeated arc, shared by several paths



(B) Solution with two repeated arcs, shared by few paths

FIGURE 6. Different sets of $K = 3$ paths

According to the example above, the way how paths spread in a network is affected by the number of repeated arcs as well as by the number of times that the repeated arcs appear in the paths. This issue will be addressed with two approaches in the following. First, by minimizing the number of times the repeated arcs appear in the set of the paths. Later, by minimizing the number of times they are repeated, which accounts also for the number of paths where they appear.

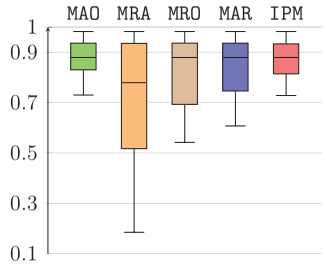
The number of occurrences of the arc $(i, j) \in A$ in a set of K paths P_K is defined as:

$$Occ(i, j; P_K) = \begin{cases} 0 & \text{if } |\{p \in P_k : (i, j) \in p\}| \leq 1 \\ |\{p \in P_k : (i, j) \in p\}| & \text{otherwise} \end{cases}$$

and the number of repeated arc occurrences in P_k is given by

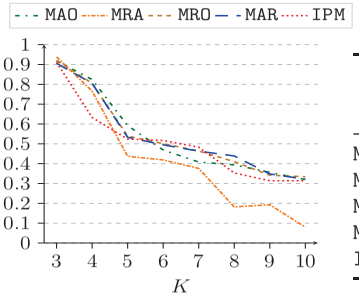
$$RO(P_K) = \sum_{(i,j)\in A} Occ(i, j; P_K).$$

This value is $RO(P_3) = 3$ in Figure 6a, and $RO(P_3') = 2$ in Figure 6b, which makes the solution in the second plot better than the first with respect to the number of repeated arc occurrences. The purpose of the next formulation



	MAO	MRA	MRO	MAR	IPM
Min. value	0.730	0.185	0.542	0.607	0.728
1^{st} quartile	0.830	0.517	0.693	0.746	0.814
Median	0.879	0.779	0.879	0.879	0.879
3^{rd} quartile	0.936	0.935	0.936	0.936	0.933
Max. value	0.982	0.982	0.982	0.982	0.982

FIGURE 16. AvDi dispersion of the unconstrained formulations in grid networks

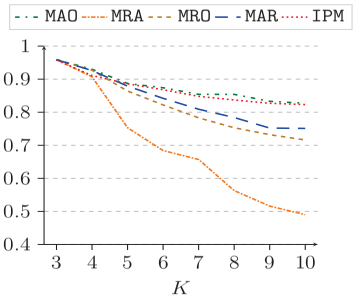


	K							
	3	4	5	6	7	8	9	10
MAO	0.919	0.824	0.592	0.469	0.408	0.394	0.356	0.320
MRA	0.938	0.766	0.437	0.419	0.376	0.182	0.193	0.080
MRO	0.920	0.805	0.539	0.501	0.464	0.410	0.342	0.334
MAR	0.906	0.805	0.533	0.495	0.463	0.439	0.348	0.322
IPM	0.911	0.633	0.524	0.517	0.483	0.353	0.314	0.314

FIGURE 17. Average MiDi of the unconstrained formulations in grid networks

Moreover, the MiDi results favor MAR. This information together with the good run times associated to this method, suggest that MAR is also worth considering in this context.

8.2. Constrained formulations. As shown in Table 13, the models obtained by adding the constraints (10) to formulations (4), (5) and (6) produce solutions with better dissimilarity results, with few exceptions. Furthermore, the increase in the dissimilarities is bigger when more paths need to be found and bigger in the grids than in the random networks. As expected, the most significant differences occurred to the pair MRA–MRAA, with dissimilarity improvements of up to 10% for the random networks and up to 244% for the grid networks. As to the pair MRO–MROA, Table 13 indicates also an improvement of the average dissimilarity of the solutions. However, the differences



	K							
	3	4	5	6	7	8	9	10
MAO	0.959	0.930	0.887	0.874	0.854	0.845	0.833	0.827
MRA	0.959	0.907	0.753	0.684	0.657	0.563	0.516	0.490
MRO	0.959	0.923	0.864	0.822	0.783	0.753	0.732	0.716
MAR	0.959	0.926	0.879	0.842	0.809	0.784	0.752	0.751
IPM	0.958	0.911	0.886	0.868	0.848	0.837	0.827	0.823

FIGURE 14. Average AvDi values of the unconstrained formulations in grid networks

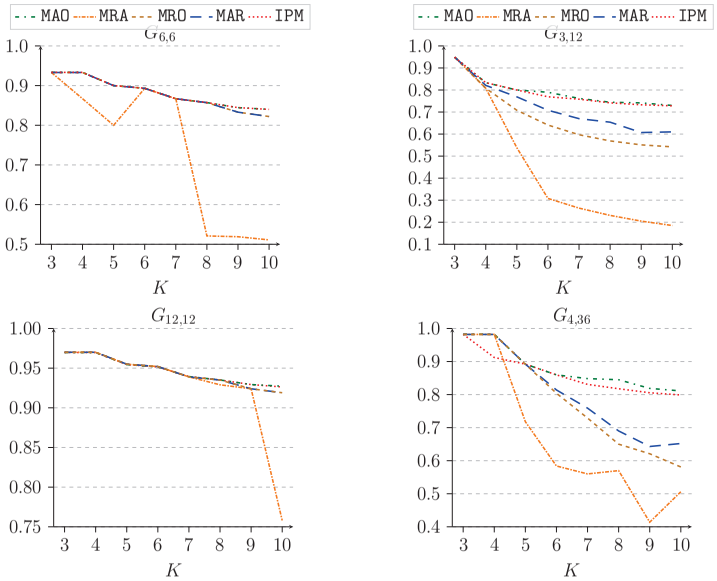


FIGURE 15. Average dissimilarity of the unconstrained formulations in grid networks

a sound approach for this specific type of networks. As to the remaining formulations, the AvDi values associated to MAR are smaller but close to the above mentioned methods (with an average difference of 5%, to both).

is to find a set of K paths which minimizes the number of repeated arc occurrences, $RO(P_K)$.

Let x_{ij}^k be decision variables defined as before and let us consider the formulation:

$$\min \quad f_3(x,y,u) = \sum_{(i,j)\in A} u_{ij} \tag{5a}$$

$$\text{subject to} \quad \sum_{j\in N:(i,j)\in A} x_{ij}^k - \sum_{j\in N:(j,i)\in A} x_{ji}^k = \begin{cases} 1 & i = s \\ 0 & i \neq s,t \\ -1 & i = t \end{cases}, k = 1,\dots,K \tag{5b}$$

$$y_{ij} \leq \sum_{k=1}^{K} x_{ij}^k, \quad (i,j) \in A \tag{5c}$$

$$(K-1)y_{ij} \geq \sum_{k=1}^{K} x_{ij}^k - 1, \quad (i,j) \in A \tag{5d}$$

$$u_{ij} \leq K\, y_{ij}, \quad (i,j) \in A \tag{5e}$$

$$u_{ij} \leq \sum_{k=1}^{K} x_{ij}^k, \quad (i,j) \in A \tag{5f}$$

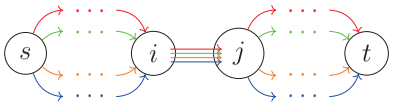
$$u_{ij} \geq y_{ij} + \sum_{k=1}^{K} x_{ij}^k - 1, \quad (i,j) \in A \tag{5g}$$

$$x_{ij}^k, y_{ij} \in \{0,1\}, u_{ij} \in \mathbb{N}_0, \quad (i,j) \in A, \quad k = 1,\dots,K \tag{5h}$$

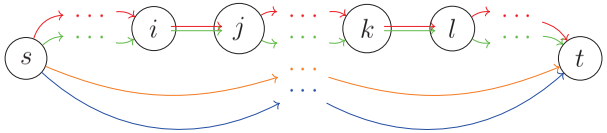
This formulation has $O(Km)$ variables and $O(Kn+m)$ constraints. For any $(i,j) \in A$, the variable y_{ij} is defined as in the previous formulation. Additionally, for a given $(i,j) \in A$:

- If $x_{ij}^k = 0$ for any $k = 1,\dots,K$, then the constraints (5f) imply that $u_{ij} = 0$.
- If $x_{ij}^k = 1$ for exactly one $k \in \{1,\dots,K\}$, then by constraints (5f), $u_{ij} \leq 1$, and by constraints (5g), $u_{ij} \geq 0$. Because the goal of the problem is to minimize $\sum_A u_{ij}$, then $u_{ij} = 0$.
- If $x_{ij}^k = 1$ for more than one $k \in \{1,\dots,K\}$, because in the last section we saw that then $y_{ij} = 1$, then by constraints (5f), $u_{ij} \leq \sum_{k=1}^{K} x_{ij}^k$, and by constraints (5g), $u_{ij} \geq \sum_{k=1}^{K} x_{ij}^k$. Combining the two conditions implies that $u_{ij} = \sum_{k=1}^{K} x_{ij}^k$.

Thus, the variable u_{ij} counts the number of times the arc (i,j) appears in the solution, or is equal to 0 if (i,j) is not repeated in that solution, for any $(i,j) \in A$. According to these points it can also be concluded that the variables u_{ij} can be relaxed as $u_{ij} \geq 0$, without changing the solution. The constraints (5e) and (5f) can be dropped, because by default the minimization of the objective function implies that $u_{ij} = 0$. Moreover, also the constraints (5c) can be skipped because the variables y_{ij} are only useful when the arc (i,j) appears in more than one path and this constraint is not affected in that case.



(A) Solution with one repeated arc, shared by several paths



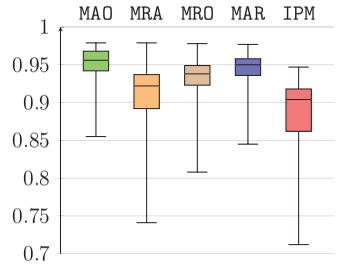
(B) Solution with two repeated arcs, shared by one path

FIGURE 7. Different sets of $K = 4$ paths

In the case shown in Figure 7, counting the number of times that the repeated arcs appear in the solution is not enough to distinguish between the two depicted solutions. In fact, Figures 7a and 7b we have $RO(P_4) = RO(P'_4) = 4$, where the repeated arcs are (i,j) in the first case, and (i,j) and (k,l) in the second. Nevertheless, the fact that the repetitions happen in different arcs should be valued, given that this is reflected in the dissimilarity of these two sets of paths. Therefore, a new concept will be introduced, the number of repetitions for any arc that appears in the solution more than once.

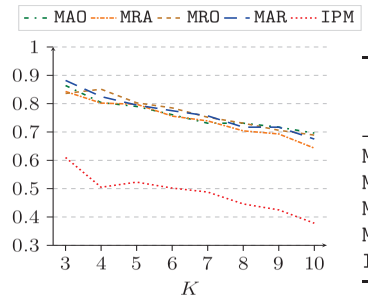
Let the number of repetitions of an arc $(i,j) \in A$ in the set P_K be the number of paths it belongs to, excluding its first utilization, that is,

$$Rep(i,j;P_K) = \begin{cases} 0 & \text{if } |\{p \in P_k : (i,j) \in p\}| \leq 1 \\ Occ(i,j;P_K) - 1 & \text{otherwise} \end{cases}$$



	MAO	MRA	MRO	MAR	IPM
Min. value	0.855	0.741	0.808	0.845	0.712
1st quartile	0.942	0.892	0.923	0.936	0.862
Median	0.956	0.922	0.938	0.950	0.904
3rd quartile	0.968	0.937	0.949	0.958	0.918
Max. value	0.979	0.979	0.978	0.977	0.947

FIGURE 12. AvDi dispersion of the unconstrained formulations in random networks



	K							
	3	4	5	6	7	8	9	10
MAO	0.864	0.804	0.790	0.761	0.731	0.717	0.696	0.687
MRA	0.843	0.802	0.790	0.761	0.731	0.717	0.696	0.687
MRO	0.836	0.851	0.803	0.785	0.752	0.731	0.706	0.689
MAR	0.882	0.825	0.795	0.775	0.757	0.717	0.717	0.675
IPM	0.610	0.505	0.523	0.502	0.488	0.446	0.425	0.378

FIGURE 13. Average MiDi values of the unconstrained formulations in random networks

chances of obtaining solutions with low values AvDi for this type of networks. The figures and the tables in Appendix A, show that the trends discussed above regarding Figure 15 are also present when studying the dissimilarity.

As shown in Figure 17, there is no clear dominance regarding the values of MiDi. Nevertheless, the worst results are associated to the IPM for $K = 3, 4$ and to MRA model for $K \geq 5$. It should be mentioned that, in all cases, the results follow heavily with the increase of K. In fact, the new formulations only look for good average dissimilarities between the pairs of paths in the solution, which may hide solutions with pairs of paths with very different dissimilarities.

In conclusion, the dissimilarity results suggest the use of MAO model when dealing with grid networks. However, once again, its run times limit its application. On the other hand, IPM was also able of finding good solutions and has the strong advantage of running in little time. Therefore, IPM seems

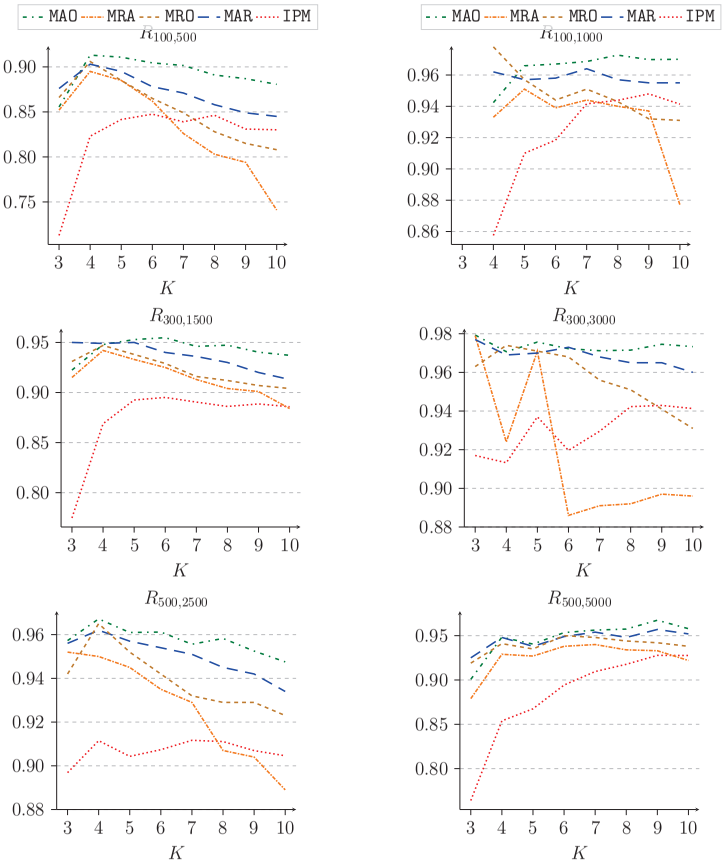


FIGURE 11. Average dissimilarity of the unconstrained formulations in random networks

Figure 15 allows a more comprehensive comparison of the `AvDi` values produced by each code. The results highlight the greater difficulty of solving the square instances, as pointed out in sections 4.1, 5.1 and 6.1, and also some inconsistency in the values associated to `MRA`. Figure 16 allows the analysis of the dispersion of these results. `MAO` is the formulation with the more uniform values, followed by `IPM`. On the other hand, the models `MAR`, `MRO` and `MRA` present the smallest dispersion (in this order). Overall, the differences between the models lay in the low quartiles, thus `MAO` and `IPM` offer less

Then, the number of arc repetitions in P_k is given by

$$Rep(P_K) = \sum_{(i,j)\in A} Rep(i,j;P_K).$$

The number of arc repetitions defined above reflects two aspects: the number of arcs shared by more than one paths as well as the number of paths that share them. Recalling the example in Figure 1, for the solution P_4 in Figure 1a we have $Rep(P_4) = 3$, because the arc (i,j) is repeated 3 times, whereas for the solution P_4' in Figure 1b we have $Rep(P_4') = 2$, because both the arcs (i,j) and (k,l) are repeated once. The next formulation aims at minimizing $Rep(P_K)$.

Like before, for modeling the problem of finding K paths from s to t which minimize the number of arc repetitions, the variables x_{ij}^k represent K paths and have value 1 if the arc (i,j) is in the k-th path from s to t or are 0 otherwise, for any $(i,j) \in A$. The formulation is as follows:

$$\min \quad f_4(x,w,u) = \sum_{(i,j)\in A} u_{ij} \tag{6a}$$

$$\text{subject to} \quad \sum_{j\in N:(i,j)\in A} x_{ij}^k - \sum_{j\in N:(j,i)\in A} x_{ji}^k = \begin{cases} 1 & i = s \\ 0 & i \neq s,t \\ -1 & i = t \end{cases}, k = 1,\ldots,K \tag{6b}$$

$$x_{ij}^k \leq w_{ij}, \quad (i,j) \in A, \quad k = 1,\ldots,K \tag{6c}$$

$$w_{ij} \leq \sum_{k=1}^{K} x_{ij}^k, \quad (i,j) \in A \tag{6d}$$

$$u_{ij} = \sum_{k=1}^{K} x_{ij}^k - w_{ij}, \quad (i,j) \in A \tag{6e}$$

$$x_{ij}^k,\ w_{ij} \in \{0,1\},\ u_{ij} \geq 0 \quad (i,j) \in A, \quad k = 1,\ldots,K \tag{6f}$$

This formulation has $O(Km)$ variables and $O(K(m+n))$ constraints. The constraints (6b) are flow conservation constraints that define a set of K paths from node s to node t. The constraints (6c) and (6d) are used to define the variables $w_{ij} \in \{0,1\}$, each one equal to 1 if and only if the arc (i,j) is used in at least one path, or 0 otherwise, for any $(i,j) \in A$. Additionally, the constraints (6e) define the auxiliary variables u_{ij}, which corresponds to the number of times that arc $(i,j) \in A$ is repeated in different paths. For a given $(i,j) \in A$:

- If $x_{ij}^k = 0$ for any $k = 1, \ldots, K$, then the constraints (6d) imply that $w_{ij} = 0$. Therefore the constraints (6e) imply that $u_{ij} = 0$.
- If $x_{ij}^k = 1$ for exactly one $k \in \{1, \ldots, K\}$, then by constraints (6c), $w_{ij} \geq 1$ which together with (6f) imply $w_{ij} = 1$, and by constraints (6e), $u_{ij} = 0$.
- If $x_{ij}^k = 1$ for more than one $k \in \{1, \ldots, K\}$, then by constraints (6c), $w_{ij} \geq 1$ which together with (6f) imply that $w_{ij} = 1$, and by constraints (6e), $u_{ij} = \sum_{k=1}^{K} x_{ij}^k - 1$.

Because both x_{ij}^k and w_{ij} are binary variables, the variables u_{ij} are implicitly defined as integers for any $(i, j) \in A$, $k = 1, \ldots, K$. Additionally, (6e) together with the non-negative constraints of the variables u_{ij} imply (6d). Therefore, the constraints (6d) can be skipped from the formulation.

Finally, we observe that constraints (6c) can be aggregated as

$$\sum_{k=1}^{K} x_{ij}^k \leq K\, w_{ij}, \quad (i, j) \in A. \tag{7}$$

Like for formulations (2) and (4), both formulations (5) and (6) admit optimal solutions with loops. However, in such cases Proposition 6.1 holds and the Algorithm 1 can be applied in order to obtain loopless optimal solutions.

Proposition 6.1. (1) *Let (x, y, u) be an optimal solution for problem (5). Let $\bar{x} \in \{0,1\}^{Km}$ be the vector output by Algorithm 1 when applied to x, $\bar{y} \in \{0,1\}^m$ and $\bar{u} \in \mathbb{N}_0^m$ be such that the constraints (5c) to (5h) are satisfied. Then, $(\bar{x}, \bar{y}, \bar{u})$ is a loopless optimal solution for problem (5).*

(2) *Let (x, w, u) be an optimal solution for problem (6). Let $\bar{x} \in \{0,1\}^{Km}$ be the vector output by Algorithm 1 when applied to x, and $\bar{w} \in \{0,1\}^m$ and $\bar{u}$ be such that the constraints (6c) to (6f) are satisfied. Then, $(\bar{x}, \bar{w}, \bar{u})$ is a loopless optimal solution for problem (6).*

6.1. Computational experiments. The tests setup presented in the following for experimentally assessing formulations (5) and (6) is similar to what was described in Section 4.1.

While a number of relaxations of formulation (5) were tested, only the results for the one with the best behavior with regard to the run times are reported. These correspond to the variant that omits the set of constraints (5c) from the original model, (5). For simplicity, we keep the same designation and in the following refer to the new model as formulation (5). Likewise, several variants of (6) were tested. The model obtained from (6) by replacing

dissimilarity obtained by `IPM` tends to improve when K grows, whereas it tends to worsen for all the formulations but `MAO`.

Figure 11 summarizes the average `AvDi` results for each random instance. The variation of the dissimilarities follows closely the pattern identified in Sections 4.1, 5.1 and 6.1 for the run times of the formulations: the instances recognised as harder to solve, namely $R_{100,500}$, are associated to the worst `AvDi` values.

Figure 12 allows a comparison of the dispersion of the results. The best scores are associated to `MAO`, followed by `MAR` and `MRO`, whereas the `IPM` is the code with more disperse values. In terms of the formulations, `MRA` was worse than the others. The box-plots in Figures 24 – 27, in Appendix A, allow a thorough analysis of the dispersion of the average `AvDi` values for each formulation. In general, the dispersion of the dissimilarities increases with K. It can also be concluded that, as expected, the hardest instances have smaller dissimilarities and a bigger dispersion of values.

The values of `MiDi` allow to study the worst case in terms of dissimilarity. Figure 13 depicts a summary of the average `MiDi` for random networks. In this case, the best results are found for the `MRO` and the `MAR` models. `IPM` is worse than all formulations with this regard, while all the latter show very similar performances (the inner differences do not exceed 5%). Also worthy of note, is the significant decrease of the `MiDi` values observed with the increase of K.

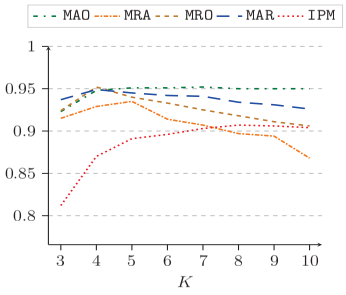
To summarize, the best and least disperse average `AvDi`'s are associated to the `MAO` model. However, its high run times undermine its application. In contrast, `MAR` produced solutions with good average and dispersion dissimilarities in less than 2 seconds, for all instances. Furthermore, the `MiDi` analysis indicates that `MAR` is less likely to produce solution with very poor dissimilarities.

Next, the results for the grid networks are analyzed. As already mentioned, for some grid networks it is possible to know the optimal value of D_1. In fact, the length of the paths in a grid network $G_{p,q}$ is constant, namely $q + p - 2$. Consequently, maximizing D_1 is equivalent to minimizing only the numerator of the fractions in its expression, which is precisely the goal of formulation (2). Thus, the values of `AvDi` for all the instances solved to optimality by `MAO` are optimal dissimilarities. Figure 14 illustrates the main differences between the five codes. Like what happened for the random networks, `MAO` produces the best results. However, this model is now followed by `IMP`, and the differences towards `MAR` have become wider and go as high as 10% whereas they do not exceed 2% for `IPM`. `MRA` still provides the worst results. Despite the good dissimilarities obtained by `MAO`, it is worth noting that these are affected by the fact that most of these instances were not solved to optimality within the time limit.

MiDi in all random instances, this was the value fixed in the rest of the experiments. Tables 28 and 29 in Appendix C present the dissimilarities and the run times of this implementation, which will be used for comparison with the introduced formulations. The dissimilarity results are far better for grids than for random networks. This is due to the fact that all paths in the used grids of a certain size have the same length. In effect, IPM naturally avoids the arcs previously used, unless the cost associated to the increase in the number of arcs exceeds the penalty. When applied to the random networks, the length of the paths varies and so does the dissimilarity of the solutions found by the method. Because this method essentially solves K shortest path problems, it has polynomial complexity of $O(Km + Kn \log n)$ and it run fast for any of the considered instances: in less than 0.70 seconds in random networks and in less than 0.05 seconds in grid networks.

The rest of the section is organized as follows: first, the results for the four initial formulations and the IPM are compared; then the effect of adding the constraints (10) to the original models is discussed.

8.1. Unconstrained formulations. The average and minimum dissimilarities of MAO, MRA, MRO, and MAR are compared in the following. The detailed results for each formulation can be found in Appendix A.



	K							
	3	4	5	6	7	8	9	10
MAO	0.923	0.948	0.951	0.952	0.950	0.950	0.949	0.944
MRA	0.915	0.929	0.935	0.914	0.907	0.897	0.894	0.860
MRO	0.924	0.952	0.940	0.933	0.925	0.918	0.911	0.906
MAR	0.937	0.949	0.945	0.942	0.941	0.934	0.931	0.926
IPM	0.812	0.870	0.891	0.896	0.903	0.907	0.906	0.904

FIGURE 10. Average AvDi values of the unconstrained formulations in random networks

We begin by analysing the results for the random networks. Figure 10 depicts the variation of the average AvDi for each model. In general, the highest values of the average AvDi are associated to the code MAO, which is followed closely by MAR (the difference between the values associated to the two models does not exceed 2%). Nevertheless, MAR surmounts MAO for $K = 3$ and $K = 4$. On the other hand, IPM has the worst performance with this regard, except for $K \geq 8$, where it outperforms MAR. In fact, the average

constraints (6c) with its aggregated version, constraints (7), was significantly faster than the others. Thus, only the results for this new model are presented. Again, for simplicity, we keep the same designation and hereafter refer to this new model as formulation (6).

The following codes were written in C, while using IBM ILOG CPLEX version 12.7 for solving the integer programs:

- MRO: Implementation for minimizing the number of repeated arc occurrences, formulation (5).
- MAR: Implementation for minimizing the number of arc repetitions, formulation (6).

TABLE 9. Run times of MRO (seconds)

	K								
$R_{n,m}$	3	4	5	6	7	8	9	10	
$R_{100,500}$	0.051	0.067	0.108	0.183	0.274	0.572	0.836	1.322	
$R_{100,1000}$	–	0.123	0.157	0.220	0.238	0.307	0.386	0.497	
$R_{300,1500}$	0.161	0.214	0.253	0.331	0.439	0.706	0.998	1.601	
$R_{300,3000}$	0.303	0.370	0.444	0.508	0.677	0.724	0.846	1.000	
$R_{500,2500}$	0.280	0.382	0.485	0.594	0.757	1.035	1.326	1.687	
$R_{500,5000}$	0.486	0.603	0.746	0.906	1.083	1.285	1.500	1.781	
Average	0.256	0.293	0.365	0.457	0.578	0.771	0.892	1.314	0.627

	K								
$G_{p,q}$	3	4	5	6	7	8	9	10	
$G_{3,12}$	0.037	0.251	0.815	1.041	5.496	15.547	8.673	11.894	
$G_{4,36}$	0.094	0.293	2.204	43.209	140.598	300	300	300	
$G_{6,6}$	0.165	0.009	0.276	0.752	2.409	3.930	23.715	14.426	
$G_{12,12}$	0.075	0.274	0.880	3.660	15.674	9.391	87.845	146.961	
Average	0.092	0.206	1.043	12.165	41.044	82.217	150.058	118.320	45.018

Unlike the code MAR, which was able to find the optimal solution for all the instances, the code MRO resumed after the 300 seconds limit for the 4×36 grids when seeking for more than 7 paths. This can be seen in Table 9, which reports their average run times, depicted in Figure 8.

The code MAR outperformed the previous in almost all cases in terms of run time. Figure 9 illustrates the results summarized in Table 10. The major differences are found in the results associated to the grid networks (where run times fall, in average, by 99.6%) and in the subset of the random

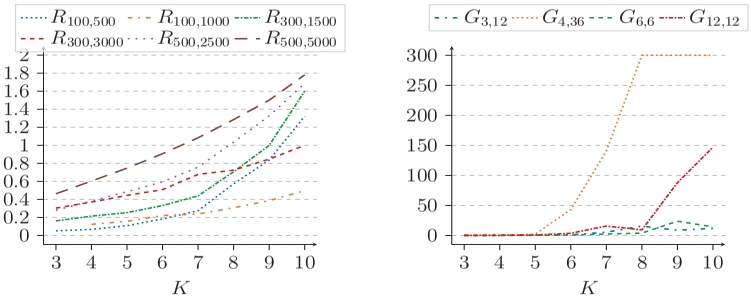


FIGURE 8. Run times of MRO (seconds)

networks previously identified as the harder to solve (which has reductions of 48% for the $R_{100,500}$ instances, 41% for the $R_{300,1500}$ instances and 25% for the $R_{300,1500}$ instances). Still, more than emphasizing the relative reductions towards MRO, it is important to point out that MAR solved all instances, in less than 2 seconds. Moreover, results suggest that MAR is far less susceptible to the effect of the variables that have been identified as inhibiting to other models (the layout of the network, its sparseness and the proportion between the value of K and the size of the network), indicating that this model can be used in a wider range of situations. As a final remark, there was some instability in the run time of the grid networks that should be clarified in a future work.

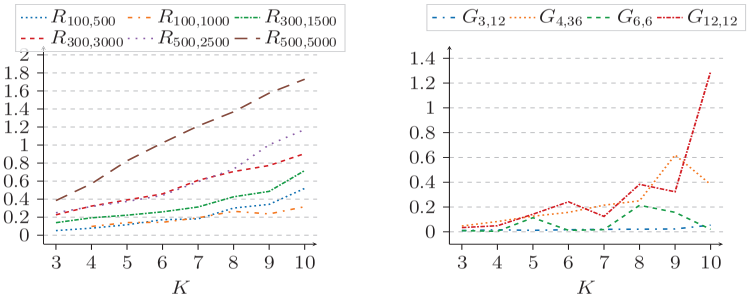


FIGURE 9. Run times of MAR (seconds)

The average integer programming gaps produced by the linear relaxation of (5), in Table 11, range between 1% ad 18% for the random instances and between 7% and 33% for the grid instances. The same values were all 0 for the linear relaxation of formulation (6).

The integer programming gap associated with MARA was equal to 0 in all the tested instances. In general these values increased for the remaining formulations after adding the new constraints, specially in the random instances and in the smaller grids and big K's. However, this variation is not very meaningful, as the unconstrained and the constrained problems are different.

8. Application to finding K dissimilar paths

In the previous sections, integer formulations were presented for the problems of finding K paths between a given pair of nodes, such that:

- the number of arc overlaps for each pair of paths is minimized;
- the number of repeated arcs is minimized;
- the number of occurrences of repeated arcs, or the number of arc repetitions, is minimized.

These problems were suggested with the purpose of capturing characteristics of sets of K dissimilar paths. Therefore in the current section the behavior of the presented approaches is discussed and compared from the perspective of the dissimilarity, based on the metric D_1 defined in Section 1. In addition, these approaches are also compared with the classical method known as the IPM and proposed in [21]. As mentioned in Section 2, the idea behind this method is to solve K shortest path problems and penalize the cost of the selected arcs every time one of those paths is computed, in order to prevent their overlap as much as possible. This approach has often been used for comparisons in the literature due to its flexibility to incorporate features of various problems as well as the simplicity of its implementation.

While a relative comparison of the dissimilarities produced by the different approaches is possible, an absolute assessment of the results requires the optimal value of D_1 to be known for each instance. As that is the case only for some of the grid instances, the analysis is based on a description of the results of each model followed by a relative overall comparison. For that purpose, both the average dissimilarity between the pairs of paths in the solution (AvDi) and their minimum dissimilarity (MiDi) are calculated for each instance. Then, the averages of AvDi and of MiDi for each set of instances are calculated.

The test bed used for this study and the testing conditions are the same described in Section 4. Furthermore, the IPM code implements the method with the same name, in C and calling the CPLEX solver for solving the shortest path problems. A set of preliminary tests was run, in order to decide how to parametrize the IPM. IPM works with unitary arc costs and was tested with the additive penalizations $\alpha = 0.25, 0.50, 0.75, 1.00$, applied to the cost of the arcs of the most recently found path. Since considering the penalization $\alpha = 1.00$ produced better results than the remaining penalties for AvDi and

TABLE 12. Run times variation for MRAA, MROA and MARA (%)

	MRAA							MROA							MARA						
	K							K							K						
$R_{n,m}$	3	4	5	6	7	8	9	10	3	4	5	6	7	8	9	10	3	4	5	6	7	8	9	10
$R_{100,500}$	32	56	38	25	19	-14	-17	-50	29	28	17	-14	2	-29	-30	-39	42	43	35	20	73	40	48	23
$R_{100,1000}$	–	64	50	38	71	42	59	56	–	38	22	8	37	15	10	9	–	83	75	87	77	49	97	86
$R_{300,1500}$	48	39	34	44	47	37	-3	-21	29	25	40	27	28	-2	-7	-23	58	60	67	80	92	67	66	42
$R_{300,3000}$	48	62	66	66	69	72	70	584	27	40	40	45	41	51	55	532	71	76	74	79	73	76	90	614
$R_{500,2500}$	50	49	45	41	56	52	11	33	31	22	25	24	35	43	10	9	60	63	79	78	81	104	56	55
$R_{500,5000}$	44	39	52	52	79	80	253	959	25	34	48	45	62	61	230	982	53	56	55	50	70	79	239	1031

	MRAA							MROA							MARA						
	K							K							K						
$G_{p,q}$	3	4	5	6	7	8	9	10	3	4	5	6	7	8	9	10	3	4	5	6	7	8	9	10
$G_{3,12}$	-13	-85	-70	-70	-61	-43	-22	-27	-43	-90	-81	-80	-95	-93	-90	-41	61	42	81	60	47	73	67	72
$G_{4,36}$	-24	9	7	294	-81	-99	-98	0	-36	-65	36	-90	-74	-97	-98	0	57	35	22	20	7	10	-43	-2
$G_{6,6}$	16	-87	-68	-92	-97	-84	14	-24	-90	79	-86	-94	-53	-91	-93	-93	78	273	-79	103	558	-86	-47	82
$G_{12,12}$	-56	-66	-57	-95	-97	-99	-98	-98	-44	-78	-77	-86	-95	-87	-95	-99	86	31	43	-46	348	-28	73	-14

TABLE 10. Run times of MAR (seconds)

	K								
$R_{n,m}$	3	4	5	6	7	8	9	10	
$R_{100,500}$	0.051	0.078	0.116	0.172	0.181	0.301	0.342	0.519	
$R_{100,1000}$	–	0.097	0.140	0.143	0.193	0.265	0.236	0.316	
$R_{300,1500}$	0.139	0.193	0.222	0.259	0.313	0.426	0.486	0.717	
$R_{300,3000}$	0.226	0.323	0.391	0.458	0.609	0.706	0.773	0.905	
$R_{500,2500}$	0.249	0.318	0.374	0.442	0.593	0.733	1.003	1.172	
$R_{500,5000}$	0.384	0.571	0.825	1.024	1.211	1.369	1.576	1.730	
Average	0.209	0.263	0.344	0.416	0.516	0.633	0.736	0.893	0.501

	K								
$G_{p,q}$	3	4	5	6	7	8	9	10	
$G_{3,12}$	0.010	0.016	0.012	0.017	0.020	0.021	0.023	0.054	
$G_{4,36}$	0.047	0.083	0.126	0.157	0.215	0.250	0.620	0.382	
$G_{6,6}$	0.010	0.004	0.114	0.012	0.018	0.214	0.157	0.020	
$G_{12,12}$	0.034	0.049	0.142	0.242	0.124	0.383	0.323	1.292	
Average	0.025	0.038	0.098	0.107	0.094	0.217	0.280	0.437	0.162

TABLE 11. Average integer programming gaps of $\texttt{MRO}_L$ (%)

	K								
$R_{n,m}$	3	4	5	6	7	8	9	10	
$R_{100,500}$	5	18	12	11	8	9	9	10	
$R_{100,1000}$	–	6	10	9	15	15	14	12	
$R_{300,1500}$	7	15	18	12	10	8	11	11	
$R_{300,3000}$	1	5	11	13	12	14	14	12	
$R_{500,2500}$	8	19	11	11	9	11	12	11	
$R_{500,5000}$	3	6	7	11	13	12	14	10	

	K								
$G_{p,q}$	3	4	5	6	7	8	9	10	
$G_{3,12}$	25	28	17	13	11	10	9	9	
$G_{4,36}$	25	33	36	21	14	11	8	7	
$G_{6,6}$	25	33	29	28	25	24	22	21	
$G_{12,12}$	25	33	29	28	25	24	22	21	

7. Bounding the number of arc presences

Formulation (4) aims at minimizing the number of arcs which appear in more than one path. The undesired consequence of the simplicity of this objective function may be that few arcs appear in many different paths, a situation which is illustrated in Figure 1. This was the motivation to consider the number of times that each repeated arc appears in the objective

function of formulations (5) and (6), presented in Section 6. In the following we propose an intermediate solution, consisting of overcoming this handicap by adding a constraint over the number of times that each arc is present in the solution. In addition, we found that this approach also had an interesting side effect in the paths dissimilarity of the solutions generated by formulations (5) and (6): when in the presence of multiple optimal solutions with respect to the number of arc repetitions, bounding the number of times that each arc appears gives an extra condition for untying those solutions, thus increasing their dissimilarity. Therefore, the new set of constraints will also be considered in the context of formulations (5) and (6). Naturally, on the downside, the new models may be more time consuming.

The new set of constraints are very similar to the set of constraints proposed by Constantino et al. [7] to prevent repetitions of the arcs over the time horizon. However, whereas in [7] the bound is an external parameter, in our case the bound is fixed by solving a simple problem that optimizes the worst case in terms of the number of times that each arc is present in the solution.

The new formulation aims at finding a set of K paths with the minimum maximum number of arc presences. It is as follows

$$\min \quad \max_{(i,j)\in A} \left\{ \sum_{k=1}^{K} x_{ij}^k \right\} \tag{8a}$$

$$\text{subject to} \quad \sum_{j\in N:(i,j)\in A} x_{ij}^k - \sum_{j\in N:(j,i)\in A} x_{ji}^k = \begin{cases} 1 & i = s \\ 0 & i \neq s,t \\ -1 & i = t \end{cases}, k = 1,\ldots,K \tag{8b}$$

$$x_{ij}^k \in \{0,1\}, \quad (i,j) \in A, \quad k = 1,\ldots,K \tag{8c}$$

where the decision variables are $x_{ij}^k \in \{0,1\}$ equal to 1 if and only if the arc (i,j) appears in path k, $(i,j) \in A$, $k = 1,\ldots,K$. This formulation can be linearized as

$$\min \quad r \tag{9a}$$

$$\text{subject to} \quad \sum_{j\in N:(i,j)\in A} x_{ij}^k - \sum_{j\in N:(j,i)\in A} x_{ji}^k = \begin{cases} 1 & i = s \\ 0 & i \neq s,t \\ -1 & i = t \end{cases}, k = 1,\ldots,K \tag{9b}$$

$$r \geq \sum_{k=1}^{K} x_{ij}^k, \quad (i,j) \in A \tag{9c}$$

$$x_{ij}^k \in \{0,1\}, \ (i,j) \in A, \ k = 1,\ldots,K \tag{9d}$$

which is equivalent to its linear programming relaxation.

Now, let R^* be the optimal value for problem (9) computed in advance. Then a new constraint can be added to formulations (4), (5) and (6) in order to prevent the number of arc presences from exceeding that value,

$$\sum_{k=1}^{K} x_{ij}^k \leq R^*, \quad (i,j) \in A. \tag{10}$$

The problems modeled by the resulting formulations are constrained and, therefore, different versions of the original ones. To assess whether the new problems produce better solutions to the K dissimilar paths problem, a set of computational experiments was performed. Results are discussed in Sections 7.1 and 8.2.

7.1. Computational experiments. In this section we analyze the impact of adding the constraints (10) to formulations (4), (5) and (6) on the run times. Considering the experimental setup described in Section 4, the following codes were tested:

- `MRAA`: implementation of formulation (4) including the constraints (10);
- `MROA`: implementation of formulation (5) including the constraints (10);
- `MARA`: implementation of formulation (6) including the constraints (10).

Like before, the codes were written in C, calling the integer programming solver IBM ILOG CPLEX version 12.7. The impact of adding the constraints (10) to the formulations introduced before on the several parameters is measured as $100 \times (\texttt{MA} - \texttt{M})/\texttt{M}\%$, where `M` stands for each of the codes listed above.

Of the new codes, `MARA` was the only one able of finding the optimal solution for all the instances within the time limit of 300 seconds. Both `MRAA` and `MROA` resumed after that limit for the 4×36 grids and $K = 10$.

According to Table 12, in most cases the constrained problems require more time to solve when the networks are denser, while the run times do not change much for the sparser instances. In some of the latter cases there is even a speed up. The speed up happens mostly for `MRA` and `MRO`, and is particularly relevant in grids, which are instances where finding solutions is difficult. It should be added that the run times required for solving the problem (9) are included in the values on Table 12. The results regarding the run times for the grid networks were uneven.