RAIRO-Oper. Res. 48 (2014) 25–51
DOI: 10.1051/ro/2013046

**RAIRO Operations Research**
www.rairo-ro.org

# THE ORDERLY COLORED LONGEST PATH PROBLEM – A SURVEY OF APPLICATIONS AND NEW ALGORITHMS

Marta Szachniuk[1,2,*], Maria Cristina De Cola[3,*], Giovanni Felici[4] and Jacek Blazewicz[1,2]

**Abstract.** A concept of an Orderly Colored Longest Path (OCLP) refers to the problem of finding the longest path in a graph whose edges are colored with a given number of colors, under the constraint that the path follows a predefined order of colors. The problem has not been widely studied in the previous literature, especially for more than two colors in the color arrangement sequence. The recent and relevant application of OCLP is related to the interpretation of Nuclear Magnetic Resonance experiments for RNA molecules. Besides, an employment of this specific graph model can be found in transportation, games, and grid graphs. OCLP models the relationships between consecutive edges of the path, thus it appears very useful in representing the real problems with specific ties between their components. In the paper, we show OCLP's correlation with similar issues known in graph theory. We describe the applications, three alternative models and new integer programming algorithms to solve OCLP. They are formulated by means of max flow problems in a directed graph with packing constraints over certain partitions of nodes. The methods are compared in a computational experiment run for a set of randomly generated instances.

**Keywords.** Edge colored graph, longest path problem, alternating path.

**Mathematics Subject Classification.** 90C35, 94C15, 97K30.

Received September 21, 2013. Accepted September 30, 2013.

[1] Institute of Bioorganic Chemistry, Polish Academy of Sciences, Noskowskiego 12/14, 61-704 Poznan, Poland. mszachniuk@cs.put.poznan.pl

[2] Institute of Computing Science, Poznan University of Technology, Piotrowo 2, 60-965 Poznan, Poland. mszachniuk@cs.put.poznan.pl; jblazewicz@cs.put.poznan.pl

[3] IRCCS Centro Neurolesi "Bonino Pulejo", S.S. 113 Via Palermo c/da Casazza, 98123 Messina, Italy. cristina.decola@gmail.com

[4] Institute of Systems Analysis and Computer Science "A. Ruberti", National Research Council, V.le Manzoni 30, 00185 Rome, Italy. giovanni.felici@iasi.cnr.it

*Co-first authors (collaborated on the writing of the manuscript equally).

Article published by EDP Sciences

## 1. Introduction

In the following paragraphs we assume basic knowledge of both, standard graph-theoretical terminology and network flow problems (see [3,13,15]). All graphs and digraphs considered in the paper are finite and simple.

Let us consider an undirected graph $G = (V, E)$, where $V$ is a set of vertices, $|V| = n$, $E$ is a set of edges, $|E| = m$. Let $C$ be a set of colors (labels), where $|C| = c$, $c \geq 2$. $G$ is called a *c-edge-colored graph* if its edges are colored in $c$ colors by means of function $f : E \to C$. If every two adjacent edges of $G$ have different colors, the graph is *properly colored* (or *properly edge-colored*, PEC) [24].

A sequence $P = \langle v_0, v_1, \ldots, v_k \rangle$ of vertices of graph $G = (V, E)$ is called a *walk* from $s$ to $t$, if $v_0 = s$, $v_k = t$, and $(v_{i-1}, v_i) \in E$, for $i = 1, \ldots, k$. A *path* from $s$ to $t$ in $G$ is a walk without repeated edges, *i.e.* $e_i, e_j \in P$ if $e_i \neq e_j$ for $i \neq j$. A *length* of the path, denoted here by $|P|$, is the number of its vertices, or – alternatively – a weighted function of its edges and/or vertices (in the following paragraphs path length is calculated as the number of involved vertices). If the path traverses each edge $e \in E$ exactly once it is called *Eulerian path.* A *simple path* from $s$ to $t$ in $G$ is a path without repeated vertices, *i.e.* $v_i, v_j \in P$ if $v_i \neq v_j$ for $i \neq j$. A simple path $P$ in $G$ is called a *Hamiltonian path* if every vertex $v \in V$ is traversed by $P$ exactly once, *i.e.* $|P| = n$.

We say, that path $P$ in graph $G$ is *properly colored* if $|P| \geq 3$ (path crosses at least two edges) and every two consecutive edges $e_i, e_{i+1} \in P$ of this path have distinct colors: $f(e_i) \neq f(e_{i+1})$. Let us note, that a properly colored path can exist also in non-properly colored graph. If the path is properly colored in two colors, then we call it an *alternating path* [14]. Every alternating path is properly colored [5].

Let us now consider a $c$-edge-colored graph $G$, and a sequence $O = \langle c_i, c_{i+1}, \ldots, c_k \rangle$ of colors from $C$ that defines a color ordering (color pattern). Path $P$ in $G$ is *orderly colored*, if $|P| \geq 3$ and the colors of consecutive edges in $P$ follow the defined color sequence $O$. If path $P$ is orderly colored due to the color sequence $O$, where the length of $O$ – here denoted by $|O|$ – equals 2, and colors in $O$ are distinct, then the path is alternating. It should be noted, that in general the orderly colored path may or may not be properly colored, depending on what is the predefined color sequence: *e.g.* $O_1 = \langle green, red, blue \rangle$ is properly colored and so is the path following this color sequence, but if the path follows $O_2 = \langle green, red, red, blue \rangle$ it is improperly colored. In what follows, we analyse problems, where colors in sequence $O$ are distinct and $|O| = c$.

The Orderly Colored Longest Path problem (OCLP) is the problem of finding the longest orderly colored path or the longest orderly colored simple path in a $c$-edge-colored graph (see an example in Fig. 1). The example orderly colored simple path $P_1 = \langle 1, 2, 3, 4, 5, 6 \rangle$ from Figure 1c is a Hamiltonian path, since $|P_1| = n$ and each vertex $v \in V$ is traversed by $P_1$ exactly once. Figure 1d shows the orderly colored longest path $P_2 = \langle 1, 2, 3, 4, 5, 6, 3, 5, 1 \rangle$ in $G$. Its length, calculated as the

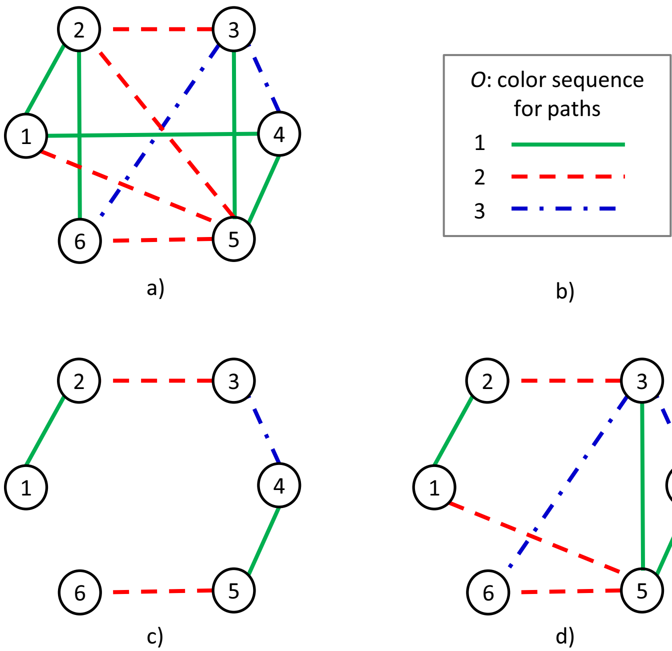


FIGURE 1. (a) An example of 3-edge-colored graph $G$, (b) predefined color sequence $O$, (c) the longest orderly colored simple path $P_1$ in $G$, and (d) the longest orderly colored path $P_2$ in $G$. Both paths, $P_1$ and $P_2$, follow color sequence $O$.

number of traversed vertices, equals $|P_2| = 9$ (let us note, that $P_2$ is not simple, thus, some vertices of $G$ are traversed more than once).

Path problems in edge-colored graphs have been addressed by many researchers (see [1, 2, 4, 5, 7, 20, 27, 28, 36, 43, 44, 49]), who discussed their different variants and computational complexity. Yeo [49] has designed an algorithm for deciding whether the given $c$-edge-colored graph contains an alternating simple cycle(s) of any length. He has proved that this decision problem is computationally easy for any number $c$ of colors. In the same paper, Yeo considered the problem of alternating cycles (of any length) passing through two given vertices of the graph. The decision version of this problem has been shown to be NP-complete. Alternating and PEC (properly edge-colored) paths that cross a predefined subset of nodes have been addressed in a number of papers. In particular, [1, 28, 33, 44] have been focused on $s - t$ paths in edge-colored graphs, *i.e.* paths starting in node denoted by $s$, and ending in the $t$ node. In [33], Manoussakis has proved that the question of the existence of two disjoint $s - t$ alternating simple paths in the edge-colored graph constitutes an NP-complete decision problem. Moreover, he has shown that the problem of the existence of two disjoint orderly colored $s - t$ paths in the

$c$-edge-colored complete graph is also NP-complete when $c \geq 4$. On the other hand, Szeider [44], and Abouelaoualim *et al.* [1] have dealt with the properly edge-colored $s-t$ paths. Szeider [44] has proved that the decision version of a properly colored $s-t$ path in $c$-edge-colored graph is easy. Abouelaoualim *et al.* [1] have shown that the question of the existence of $k$ pairwise vertex/edge disjoint PEC $s-t$ paths in a $c$-edge-colored graph is an NP-complete problem for $k \geq 2$ and $c = \Omega(n^2)$ (where $n$ is the number of graph vertices). It remains NP-complete if the graph has no properly edge-colored cycles and $c = \Omega(n)$. In [1], it has been also shown that the following problems: the shortest PEC path, the shortest PEC path with forbidden pairs, and the shortest PEC cycle, can be solved by polynomial time algorithms for a particular class of instances (see [1] for details). Following the results presented in [1], Gourves *et al.* [28] have shown that, given a $c$-edge-colored graph without PEC cycles, it is possible to find – in polynomial time – the properly edge-colored $s-t$ paths, which visit all vertices of the graph a prescribed number of times. In a consequence, it has been proved, that PEC Eulerian $s-t$ path problem is polynomially solvable for $c$-edge-colored graphs, which do not contain PEC cycles. Another study on Eulerian and Hamiltonian problems in edge-colored graphs is given in [7]. Benkouar *et al.* [7] have focused on simple paths and cycles in complete graphs. They have shown the NP-completeness of several variants of the Hamiltonian path/cycle problem: (i) the existence of a Hamiltonian (123) cycle in 3-edge-colored complete graph; (ii) the existence of $(x_1, x_2, x_3)$ cycle containing 6 selected vertices of the 3-edge-colored complete graph; (iii) the existence of a Hamiltonian $x_1 x_2 \ldots x_k$ cycle in a $k$-edge-colored complete graph, where $k \geq 4$ and $n = kp$ ($n$ is the number of graph vertices, $p$ is a positive integer); (iv) variant similar to (iii) but for $n = kp + 1$; (v) the existence of orderly colored cycle of length $pk$, repeating the order of colors $p$ times, in a complete $k$-edge-colored graph. On the other hand, a proof of NP-hardness of the Hamiltonian path problem in 2-edge-colored simple graph has been presented in [2]. It has been based upon the transformation of the Hamiltonian path in non-colored graphs to the considered problem. Another problem, focusing on finding the longest orderly colored simple path, first announced in [41], has been discussed in detail in [43] where the proof of its NP-hardness has been provided. Finally, path problems in the edge-colored multigraphs have been analysed in [4]. As shown by Bang-Jensen and Gutin [4], the problem of existence/finding a longest alternating simple path in a 2-edge-colored complete multigraph is computationally easy.

Various issues can be modelled as the orderly colored path problem. Some have been already studied in the literature, especially those related to colored Eulerian paths. Here, we focus our attention more on questions that require routing through graph nodes, in particular on colored Hamiltonian paths. Considering such problems, we show three formulations for OCLP where an edge-colored graph is transformed splitting the vertex set in $k$ partitions (a $k$-partite digraph) and additional constraints are imposed over these partitions. The presented formulations share some characteristics with the Shortest Path Tour Problem (SPTP) described in [25], shown to be solvable either with the Shortest Path algorithms (see [3,19,26]) or with dynamic programming [8]. Indeed, SPTP is a polynomial-time transformation into a single source destination shortest path problem of a multi-stage digraph with nonnegative arc lengths, subject to constraints on the partite sets of vertices. Nevertheless, the substantial difference between SPTP and OCLP is to be found in the formulation of constraints on the partition sets: in the former problem they are the covering constraints, while in the latter one they serve as packing constraints. This constitutes a main difference, that results in computational hardness of the latter problem.

The paper is structured as follows. In Section 2 we look through varius applications of the orderly colored longest path in modelling different real-world problems. Section 3 deals with three integer programming solutions to the problem in question, in particular: longest path in an acyclic $n$-partite graph (3.1), longest path in a cyclic $c$-partite graph (3.2) and longest path in a cyclic $c$-connected graph (3.3). In Section 4, we provide the results of some experimental runs over a set of simulated problems aimed to compare the presented algorithms. In Section 5 we summarize the results of previous research concerning edge-colored paths and we propose the directions of future study.

## 2. Applications of OCLP

Graphs with colored edges and/or vertices can be used to model search and decision problems. Here we focus on edge-colored graphs which have been found useful in modelling a variety of issues, for example when a certain subset of edges – a spanning tree or a matching - is required to use the minimum number of colors/labels (see [16–18]). We look at problems whose solution involves finding an orderly colored path (or simple path) in such graphs. In particular, we are interested in solutions with the maximum length. Since the longest path problem can be considered in terms of edges or nodes included in the solution, we provide examples for both cases. However, our interest lies basically in simple paths whose lengths are calculated as the the number of traversed nodes.

Let us first analyse a general class of problems that are modelled using grid, lattice or Mesh graphs. A grid can be transformed into $c$-edge-colored graph if node coordinates are not relevant when the modelled problem is solved. Then $c$ equals the grid dimensionality. For example a rectangular grid can be represented with a 2-edge-colored graph, where color $c_1$ represents horizontal edges and color $c_2$ is for vertical edges of a grid (Fig. 2). A similar rule applies to a transformation of lattice and Mesh graphs. Now, a problem of constructing a path going alternately through horizontal and vertical edges of a grid is represented by the alternating path problem in 2-edge-colored graph. A great advantage of edge-colored graph representation over grid is that we always deal with two-dimensional model of a problem (irrespectively of the grid dimension), which can be easily pictured and analysed even manually. Extending the number of grid dimensions results only in addition of yet another hue to paint the graph edges.

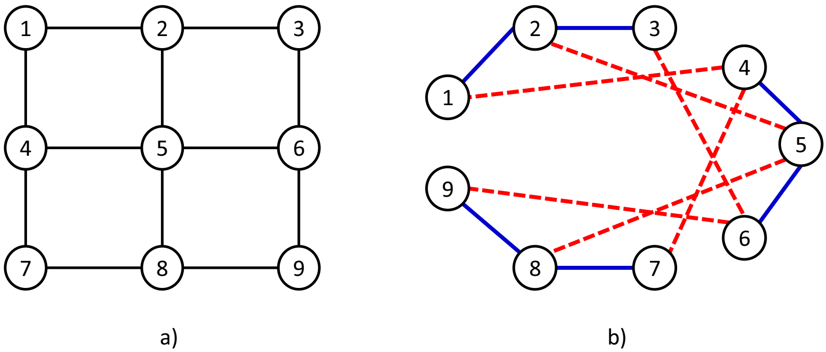


FIGURE 2. (a) An example grid graph $G$, and (b) its representation as 2-edge-colored graph $G'$ (red edge in $G'$ represents vertical edge from $G$, blue edge in $G'$ is for horizontal edge from $G$).

Grid graphs, and hence also edge-colored graphs, are used to model city blocks where intersections are vertices and streets are edges [34]. Grid is also a straightforward representation of the Manhattan topology. Considering the problem of a traveling salesman in the city, provided that his points of interests are located at the intersections, we solve the Hamiltonian path problem in the city-block graph. If additional constraints are given for a graph (*e.g.* some edges represent one-way streets) and/or for a required salesman route (*e.g.* there are some preferences according to which points of interest must be visited before the others), the problem can be modelled using the edge-colored graph. Then, solving the TSP requires searching for the orderly colored simple path in the edge-colored graph.

Similarly, the edge-colored graph may represent a two-dimensional memory array where memory is accessed moving down or across from each cell. In this problem each vertex of a graph corresponds to one memory cell and edges connect adjacent cells. A color of an edge corresponds to the direction of transition between the two cells. Finding the longest simple path in the graph modelling this problem corresponds to accessing all the data stored in the memory array [32].

Edge-colored graphs are also applied to represent grid graphs with obstacles (*e.g.* forbidden vertices). An example of such a problem is the longest path routing discussed in [45]. It concerns the process of designing the printed circuit boards (PCB) and printed wiring boards (PWB). PCBs and PWBs are found in many commercially produced electronic devices, where they are used as platforms to connect electronic components via conductive paths, tracks and signal tracing. Although their production and soldering is automated, it must be preceded by the initial design and layout, which involve separate algorithms supporting circuit and/or wire placement. One of the crucial problems to be solved in this initial stage is bus routing within tight minimum and maximum length bounds. The issue is formulated as a gridded longest path routing with obstacles and can be optimally solved for example by a mixed integer linear programming procedure.

There are other interesting examples of routing problems which can be modelled using edge-colored graphs and orderly colored paths, *e.g.* pick-up and delivery, where different types of pick-up and deliveries must follow a predefined scheme, electric vehicles routing, where loading and discharging arcs must alternate, or separated waste collection routing. Some applications are also found for routing in colored graphs modelling fibre-optic networks. Recent developments in optical technology have resulted in the rapid growth of fibre-optic networks applying wavelenght division multiplexing (WDM) in the physical layer. The WDM technology allows for simultaneous transmission within a single optical fibre on a certain number of wavelengths, and the capacity $n$ of the link may even exceed one hundred wavelengths. Whereas in theory this could result in the increase of the bandwidth of a single link even by a factor of $n$, in practice utilizing such a gain is difficult and requires the application of efficient routing and switching algorithms. This problem can be solved as a minimum cardinality wavelength routing problem. Thus, given now the set of routes in a given network, the problem of assigning a minimum number of wavelengths is simply to find such a coloring of routes (every route receives one color) that a maximum number of colors assigned to any edge (belonging to different routes) is at its minimum. In general, this problem is NP-hard but some polynomial time algorithms maybe also given [6, 9, 29, 30, 35].

Edge-colored schemes are also useful in modelling homogeneous faults in networks [46]. This problem refers to the area of communication networks. In such a network, a broken node or edge causes a communication failure, if it takes place along the route passing the damaged component. In fact, a failure of a node (*e.g.* representing a router/a modem) can be modelled as a failure of a communication link attached to this node (*i.e.* an edge). Thus, a problem of network robustness against faults can be studied in the context of edge failures in the communication network. While preparing a graph model of the network, one can use colors to mark the devices according to their failure probability, *e.g.* all routers of brand $K$ have the same failure probability, so the corresponding edges obtain the same color. It seems reasonable to assume that edges with the same color, *i.e.* all brand $K$ devices, can fail at the same time, because of some platform dependent virus attack. Such failures are called homogeneous. The number of different probability values imposes the number of colors used in the graph. Communication networks should be robust agaist homogeneous faults, what can be achieved in several ways – each of them defines the other network problem. One should study the number of colors (*i.e.* the number of non-homogeneous devices in the network) necessary to provide survivable communication. But this is equally important to ensure the communication in case of some local failure. The latter issue can be modelled as a problem of routing through the edge-colored graph with restraints (*e.g.* avoiding some colors during communication route or following some pattern of colors – *i.e.* passing through specified devices in the required order).

The other application of edge-colored graphs and properly/orderly colored paths can be found in modelling secure transmission of messages in networks. In the graph model of this problem, a color assigned to an edge represents a transmission code. When the encoded message is propagated through a network (represented by the edge-colored graph) along a monochromatic path, then it may be easily decoded in any local decoding spot. But if the same message is passed through a properly edge-colored path the security of transmission greatly increases. Thus, a problem of planning the secure message route between two nodes of the network (let us denote them by $s$ and $t$) is reduced to the problem of finding the properly colored $s-t$ path in the edge-colored graph [33].

Path constrained in edge colors can be also used to model chessboard problems. A good example here is the problem of knight's moves, known in recreational mathematics as the knight's tour problem [47]. The knight is the only chessman that cannot move straight. It is only allowed to move along the $L$-shape line, *i.e.* first it takes two squares in vertical or horizonal direction, and – next – one square in a perpendicular direction. In the knight's tour problem one tries to visit all squares of the chessboard in a sequence of knight's moves, and end the tour in the starting square. A solution to this problem usually starts from proposing a graph representation of the chessboard, where each graph node corresponds to one chessboard square. To model the problem one can apply a grid graph or an edge-colored graph, where edge colors represent possible moves. Finding a solution the knight's tour problem amounts to finding a specific Hamiltonian cycle in the graph representation of the chessboard. However, unlike the Hamiltonian cycle problem, the knight's tour has been shown to be solvable in linear time [22].

Finally, in the last two decades $c$-edge-colored graphs have been proposed to model various problems in genetic and molecular biology [37,41]. For instance, [23] discusses the problem of spatial order of chromosomes in metaphase nuclei. It seems that the chromosome order is responsible for gene expression as well as some mechanisms involved in the pairing processes. Bennett's principle defining chromosome arrangement based on similarity relation between their arms, has given rise to the design of a multigraph containing two types of edges (2-edge-colored graph). The authors of [23] study the chromosome order solving the problem of alternating Hamiltonian cycles in the corresponding graph model. It is proved, that alternating Hamiltonian cycle determines the order of chromosomes in case their number is even.

Another problem that can be modelled using edge-colored graphs concerns the Double Digest Problem (DDP), which occurs during DNA mapping. Physical maps that show DNA molecule with the cleavage sites, *i.e.* points in which DNA is cleaved by the restriction enzymes, are basic structures used in molecular genetics. However, a construction of such maps (DDP) is a difficult problem, even for small-scale mapping. The number of solutions to DDP increases exponentially with the DNA sequence length and is hard to be analysed, although a great majority of these solutions are highly similar and easily transformable into one another. In [36], Pevzner studies these multiple solutions from the combinatorial point of view. He uses 2-edge-colored graphs to represent the DDP and shows an association between DDP solutions and alternating Eulerian cycles (which are longest orderly colored paths in his edge-colored graphs). Based on the previous studies concerning the transformations between similar DDP solutions, the transformations of alternating cycles are presented for 2-edge-colored graphs.

Recently, the correlation signals occurring between the nuclei of RNA molecule during Nuclear Magnetic Resonance (NMR) experiments have been represented with an edge-colored graph model [2, 10, 11, 42, 43]. NMR experiments are used in the determination and analysis of protein and nucleic acid structures, they often complement in silico prediction [12] and are applied in the validation of computed molecular models [48]. In [41] the authors present an enumerative algorithm that solves the problem of NMR signal assignment in the proposed graph model. It is assumed that each edge of a $c$-edge-colored graph is colored according to the type of interaction represented as a transition between a pair of cross-peaks in a 3D NMR spectrum. The longest orderly colored simple path along the vertices of $G$ is the reconstruction of a transfer pathway between the cross-peaks in the spectrum, and - respectively – a magnetization transfer between the nuclei of RNA molecule. Similarly, in [2, 10, 42], the problem of NMR signal assignment is studied for the two-dimensional spectra. The search space in this case can be modelled as 2-edge-colored graph, and the solution to the 2D assignment problem is the longest alternating simple path. Based on this graph problem solution, NMR experimenter can start the process of a reconstruction of the three-dimensional shape of the analysed RNA molecule [38–40].

It seems that graphs with colored edges can be applied to model various problems, from molecular biology and genetics to engineering. In particular, the problems with ordering restraints can be solved via construction of the orderly colored longest paths in edge-colored graphs. Edge colors in such graphs, as well as color patterns defined for paths are suitable to represent dependencies between graph nodes. A variety of solutions can be proposed to solve the OCLP for both cases: paths and simple paths. In the following section we present three methods, selected to solve the longest orderly colored simple path problem, being recently intensively studied in the literature.

## 3. Models and algorithms

In this Section, we describe three new models and algorithms for the OCLP problem (first introduced in report [21]). In the next Section, the results of computational experiments run to compare their efficiency and usability will be provided. The first model is based on the longest path problem in an $n$-partite graph, in which the number of partitions equals the number of vertices in the original edge-colored graph. The second model is a transformation of the latter one, where the number of partite sets depends on the number of colors. The third model is constructed upon an $n$-partite graph, where each partition is a $c$-connected subgraph of the original graph $G$. All formulations are described based on the example graph pictured in Figure 1a. For simplicity, in all cases we assume that a feasible path must start from the first color in the color sequence (which is green in the example from Fig. 1a). Such an assumption can be easily removed in the presented models, either by the addition of zero-cost arcs, or by solving $c$ instances of the same model.

### 3.1. LPnPP: Longest path in the acyclic $n$-partite graph problem

In this section, we propose a formulation of OCLP as the longest path problem in the acyclic directed graph (network) with vertex set divided into $n$ partitions, and we propose an algorithm to solve this problem.

Let $G = (V, E)$ denote an original $c$-edge-colored undirected graph with $n$ vertices, $V = \{v^1, v^2, \ldots, v^n\}$, and $m$ edges, which are colored with $c$ colors. Let $O = \langle c_1, \ldots, c_c \rangle$ be the sequence determining the order of colors to be followed by the orderly colored path in $G$. We propose to transform $G$ into $n$-partite digraph $G' = (V', A')$ (called also a network) according to the following procedure:

1. A set of vertices $V'$ is composed of $n$ copies of $V$ from the original graph. The $i$-th copy of $V$ in digraph $G'$ is called a *partite set* and denoted as $L_i$. $L_i = \{v_i^1, v_i^2, \ldots, v_i^n\}$ for $i = 1, \ldots, n$. Each partite set is refered to as partition of $G'$, and thus, $G'$ is an $n$-partite graph. Additionally, one source vertex $s$ and one sink (destination vertex) $t$ are added to $V'$: $L_0 = \{s\}$ and $L_{n+1} = \{t\}$. Consequently, $V' = L_0 \cup L_1 \cup L_2 \cup \ldots \cup L_n \cup L_{n+1}$, where $L_i \cap L_j = \emptyset$ for $i \neq j$, and $|V'| = n^2 + 2$.
   We can also say, that vertex set $V'$ is composed of $n$ *level sets*: $V' = V^1 \cup V^2 \cup \ldots \cup V^n$, where $V^i \cap V^j = \emptyset$ for $i \neq j$. The $i$-th level set $V^i$ contains $n$ copies of vertex $v^i \in V$ from the original graph, *i.e.* $V^i = \{v_1^i, v_2^i, \ldots, v_n^i\}$, for each $i \in \{1, \ldots, n\}$. Note that $V^l \cap L_r = \{v_r^l\}$ for each $r, l \in \{1, \ldots, n\}$.
2. A set of arcs $A'$ contains weighted arcs between vertices of $G'$ and is created in the following way:
   - The source vertex $s$ is connected to every vertex in $L_1$ by an arc with zero cost.
   - Every vertex $v \in V' - \{s, t\}$ is connected to the sink $t$ by an arc (referred to as *exit arc*) with zero cost.
   - Every edge $(v^i, v^j) \in E$ with color $c_y$, $y \in \{1, \ldots, c\}$, is replaced by arcs with cost one in $A'$, $(v_r^i, v_{r+1}^j)$ and $(v_r^j, v_{r+1}^i)$, for all $r = 1, \ldots, n-1$: $(r \mod c) = y$ or $((r \mod c) = 0$ and $y = c)$.

   Note, that all arcs connecting two neighboring partite sets, $L_r$ and $L_{r+1}$, have the same color. Moreover, one edge from $G$ can correspond to more than two arcs in $G'$ if $n > c$. The colors of arcs between consecutive partitions follow the color sequence $O$ (see Fig. 3).

**Proposition 3.1.** *Let $G' = (V', A')$ be an $n$-partite digraph constructed upon the edge-colored graph $G$ due to the above procedure. The longest simple path $P$ from $s$ to $t$ in $G'$, such that each level set of $V'$ is touched by $P$ only once, is the orderly colored longest path in $G$.*

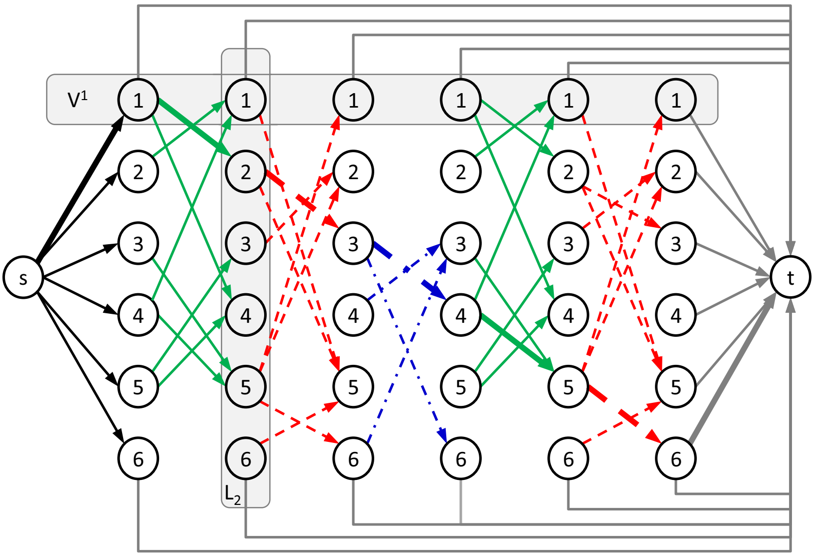


FIGURE 3. The 6-partite digraph constructed upon the 3-edge-colored graph from Figure 1a. The *green-red-blue* path is marked with thick arcs.

To clarify the procedure of network construction, let us consider the example edge-colored graph $G = (V, E)$ (Fig. 1a) and its transformation to $n$-partite digraph $G' = (V', A')$ with respect to the color sequence $O = \langle green, red, blue\rangle$ (Fig. 1b). The vertex set $V$ from the original graph contains $n = 6$ vertices and $m = 11$ edges in three colors: red, green, and blue. The corresponding digraph $G'$ should have 38 vertices, *i.e.* 36 vertices collected in 6 partition sets and 2 additional vertices $s, t$. The first level set $V^1$ containing 6 copies of vertex $v^1$ is presented in grey horizontal box. The grey vertical box represents the second partition $L_2$ being a copy of the whole vertex set $V$. Note, that – for clarity – not all exit arcs have been displayed in the figure, but only those connecting vertices from $V^1$ and $V^6$ to node $t$.

An algorithm finding the longest path in $n$-partite digraph $G' = (V', A')$ operates on the Integer Programming (IP) formulation of the problem. The following variables are used in the formulation:

- $x_r^{ij}$ – a decision variable associated with arc $(v_r^i, v_{r+1}^j) \in A'$. $x_r^{ij} = 1$ if there is a flow on the corresponding arc, otherwise $x_r^{ij} = 0$.
- $p_r^{ij}$ – cost of arc $(v_r^i, v_{r+1}^j) \in A'$. $p_r^{ij} = 0$ if the corresponding arc is adjacent either to the source or to the sink, otherwise $p_r^{ij} = 1$.

The Longest Path $n$-Partite graph Problem with packing constraints (LPnPP) is an optimization problem formulated as follows:

$$\text{Maximize} \quad \sum_{(v_r^i, v_{r+1}^j)\in A'} p_r^{ij} x_r^{ij} \qquad \text{(OBJ)}$$

subject to:

$$\sum_{(v_r^i, v_{r+1}^j)\in A'} x_r^{ij} - \sum_{(v_{r-1}^j, v_r^i)\in A'} x_r^{ji} = 0 \;\; \forall v_r^i \in V' - \{s,t\} \qquad \text{(C1)}$$

$$\sum_{(s, v_1^j)\in A'} x_1^{sj} = 1 \qquad \text{(C2)}$$

$$\sum_{(v_r^i, t)\in A'} x_r^{it} = 1 \qquad \text{(C3)}$$

$$\sum_{\substack{(v_{r-1}^j, v_r^i)\in A' \\ v_r^i \in V^l}} x_r^{ji} \leq 1 \qquad l = 1, 2, \ldots, n \qquad \text{(C4)}$$

$$x_r^{ij} \in \{0, 1\} \qquad \forall (v_r^i, v_{r+1}^j) \in A' \qquad \text{(C5)}$$

(C1) is the classical flow balance constraint of a network flow problem. Its first term represents the total flow emanating from vertex $v_r^i$, the second term represents the total flow entering into $v_r^i$. From the source ($s$) only one unit of flow is sent to the sink ($t$) as imposed by (C2) and (C3). Thus, the solution of the problem sends one unit of flow from $s$ to $t$ along a path $P$. The flow must satisfy the *packing constraints* (C4), which state that the total flow entering each level set $V^l$ equals at most 1 (*i.e.* for each $V^l$ at most one vertex can be visited by the path). Thus, (C4) ensures that only one copy of the same vertex from the original graph is visited by $P$. Summing up we can formulate the following proposition:

**Proposition 3.2.** *A feasible (optimal) solution of LPnPP is an orderly colored (longest) path in the corresponding edge-colored graph $G$. An optimal solution of LPnPP composed of $n-1$ vertices is an orderly colored Hamiltonian path in $G$.*

### 3.2. LPcPP: Longest path in the cyclic $c$-partite graph problem

Below, we propose a formulation of OCLP as the longest path problem in the cyclic directed graph with vertex set divided into $c$ partitions, and we propose an algorithm to solve this problem. In relation to the previous proposition (Sect. 3.1), this model significantly reduces the dimension of a graph, what results from its cyclic characteristics. On the other hand, it requires an introduction of cycle elimination constraints and cycle separation by the algorithm.

Again, let $G = (V, E)$ denote an original $c$-edge-colored undirected graph with $n$ vertices, $V = \{v^1, v^2, \ldots, v^n\}$, and $m$ edges, which are colored with $c$ colors. Let $O = \langle c_1, \ldots, c_c\rangle$ be the sequence determining the order of colors to be followed by the orderly colored path in $G$. We propose to transform $G$ into $c$-partite digraph $G^* = (V^*, A^*)$ according to the following procedure:

1. A set of vertices $V^*$ is composed of $c$ copies of $V$ from the original graph. The $i$-th copy of $V$ in digraph $G^*$ is called a *partite set* and denoted as $L_i$. $L_i = \{v_i^1, v_i^2, \ldots, v_i^n\}$ for $i = 1, \ldots, c$. Each partite set is refered to as partition of $G^*$, and thus, $G^*$ is a $c$-partite graph. Additionally, one source vertex $s$ and

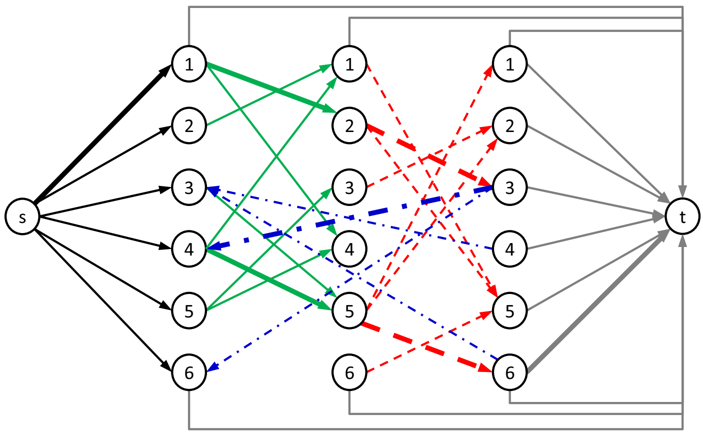


FIGURE 4. The 3-partite digraph constructed upon graph from Figure 1a. The *green-red-blue* path is marked with thick arcs.

one sink (destination vertex) $t$ are added to $V^*$: $L_0 = \{s\}$ and $L_{c+1} = \{t\}$. Consequently, $V^* = L_0 \cup L_1 \cup L_2 \cup \ldots \cup L_c \cup L_{c+1}$, where $L_i \cap L_j = \emptyset$ for $i \neq j$, and $|V^*| = nc + 2$.
We can also say, that vertex set $V^*$ is composed of $n$ *level sets*: $V^* = V^1 \cup V^2 \cup \ldots \cup V^n$, where $V^i \cap V^j = \emptyset$ for $i \neq j$. The $i$-th level set $V^i$ contains $c$ copies of vertex $v^i \in V$ from the original graph, *i.e.* $V^i = \{v_1^i, v_2^i, \ldots, v_c^i\}$, for each $i \in \{1, \ldots, n\}$. Note that $V^l \cap L_r = \{v_r^l\}$ for each $l \in \{1, \ldots, n\}$, $r \in \{1, \ldots, c\}$.

2. A set of arcs $A^*$ contains weighted arcs between vertices of $G^*$ and is created in the following way:
   - The source vertex $s$ is connected to every vertex in $L_1$ by an arc with zero cost.
   - Every vertex $v \in V^* - \{s, t\}$ is connected to the sink $t$ by an arc (referred to as *exit arc*) with zero cost.
   - Every edge $(v^i, v^j) \in E$ with color $c_y$, $y \in \{1, \ldots, c\}$, is replaced by two arcs with cost one in $A^*$:
     – $(v_r^i, v_{r+1}^j)$ and $(v_r^j, v_{r+1}^i)$ if $r = y$, $r \in \{1, \ldots, c-1\}$.
     – $(v_c^i, v_1^j)$ and $(v_c^j, v_1^i)$ if $y = c$.

   Note, that all arcs connecting two neighboring partitions have the same color ($L_c$ is considered a neighbor of $L_1$ and $L_{c-1}$). The colors of arcs between consecutive partite sets follow the color sequence $O$.

Similarly as in Section 3.1, we show (see Fig. 4) how the example edge-colored graph $G$ (Fig. 1a) is transformed into a $c$-partite digraph $G^* = (V^*, A^*)$ with respect to the color sequence $O = \langle green, red, blue \rangle$. $G^*$ should have 20 vertices, 3 partite sets $L_1, L_2, L_3$ and 6 level sets $V^l$ (see Fig. 4). Note, that green edges from $G$ have their arc representatives in $G^*$ between $L_1$ and $L_2$, red edges – between

$L_2$ and $L_3$, and blue edges - between $L_3$ and $L_1$. Exit arcs are light grey, and – for clarity – not all of them are drawn in the figure.

An algorithm to solve the above defined problem is based on the IP formulation, which uses two types of variables: $x^{ij}$ – a decision variable corresponding to arc $(v^i, v^j) \in A^*$, and $p^{ij}$ – cost of arc $(v^i, v^j) \in A^*$. The values of decision and cost variables are calculated in the same manner as for LPnPP. Given, that $s$ is the source vertex, and $t$ is the sink, the Longest Path $c$-Partite graph Problem with packing constraints (LPcPP) is formulated as follows:

$$
\begin{array}{lllr}
\text{Maximize} & \sum_{(v^i,v^j)\in A^*} p^{ij}x^{ij} & & \text{(OBJ)}\\
\text{subject to:} & & & \\
& \sum_{(v^i,v^j)\in A^*} x^{ij} - \sum_{(v^j,v^i)\in A^*} x^{ji} = 0 & \forall v^i \in V^* - \{s,t\} & \text{(C1)}\\
& \sum_{(s,v_1^j)\in A^*} x^{sj} = 1 & & \text{(C2)}\\
& \sum_{(v^i,t)\in A^*} x^{it} = 1 & & \text{(C3)}\\
& \sum_{\substack{(v^i,v^j)\in A^* \\ v^i\in V^l}} x^{ij} \le 1 & l = 1,2,\ldots,n & \text{(C4)}\\
& \sum_{(v^i,v^j)\in \Gamma} x^{ij} \le |\, \Gamma \,| -1 & \Gamma \in \widehat{\Gamma} & \text{(C5)}\\
& x^{ij} \in \{0,1\} & \forall\, (v^i,v^j) \in A^*. & \text{(C6)}
\end{array}
$$

Like in the previous model, (C1)–(C3) are the flow balance constraints, and the packing constraints (C4) state that in every level set $V^l$ at most one vertex can be visited by the path. Since the graph is cyclic, we need to enforce the elimination of all the orderly colored cycles. In the IP formulation this is achieved due to (C5). Cycle elimination constraints are not present in the initial stages of problem solving, but they are iteratively added and eliminated if they appear in the current solution. Concluding we can state:

**Proposition 3.3.** *A feasible (optimal) solution of LPcPP is an orderly colored (longest) path in the corresponding edge-colored graph $G$. An optimal solution of LPcPP composed of $n-1$ vertices is an orderly colored Hamiltonian path in $G$.*

### 3.3. LPcCP: Longest path in the cyclic $c$-connected graph problem

Here, we introduce the third ILP model for OCLP, which represents the problem as the longest path in the cyclic $c$-connected digraph. Similarly to LPcPP, this formulation requires elimination of cycles. Again, the number of vertices in the new representation depends on the number of colors, but the graph structure allows for more efficient usage of color sequence in the search for optimum solution.

Let us recall that $G = (V, E)$ denotes the original $c$-edge-colored undirected graph with $n$ vertices, $V = \{v^1, v^2, \ldots, v^n\}$, and $m$ edges, which are colored with $c$ colors. $O = \langle c_1, \ldots, c_c \rangle$ is the sequence determining the order of colors to be followed by the orderly colored path in $G$. Graph $G$ can be transformed $c$-connected digraph $G^\# = (V^\#, A^\#)$ according to the following procedure:

1. A set of vertices $V^{\#}$ is composed of $c$ copies of the original vertex set $V$. The $i$-th copy of $V$ in digraph $G^{\#}$ is denoted by $L_i$. Additionally, one source vertex $s$ and one sink (destination vertex) $t$ are added to $V^{\#}$. Consequently, $|V^{\#}| = nc + 2$.

   We can also say, that vertex set $V^{\#}$ contains $n$ subsets $V^i$, where the $i$-th subset contains $c$ copies of vertex $v^i \in V$ from the original graph, *i.e.* $V^i = \{v_1^i, v_2^i, \ldots, v_c^i\}$ for each $i \in \{1, \ldots, n\}$. Thus, $V^{\#} = V^1 \cup V^2 \cup \cdots \cup V^n \cup \{s, t\}$, where $V^i \cap V^j = \emptyset$ for $i \neq j$. Each copy of vertex $v^i$ is associated with a color, *i.e.* $v_j^i$ has color $c_j$.
2. A set of arcs $A^{\#}$ contains weighted arcs between vertices of $G^{\#}$ and is created in the following way:
   - For every $V^i \subset V^{\#}$, $i = 1, \ldots, n$, all vertices in $V^i$ are connected by a directed cycle, so that the sequence of vertex colors in the cycle follows the color sequence $O$. Thus, we obtain $n$ subgraphs $G^i = (V^i, A^i)$, where $|V^i| = c$, and $|A^i| = c$ for each $i \in \{1, \ldots, n\}$. Each $G^i \subset G^{\#}$ is a $c$-connected subgraph.

     Let $A'$ denote a subset of arcs composed of all $A^i$: $A' = A^1 \cup A^2 \cup \cdots \cup A^n$. Every arc in $A'$ has a zero cost.
   - Every edge $(v^i, v^j) \in E$ with color $c_y$, $y \in \{1, \ldots, c\}$, is replaced by two arcs with cost one, $(v_y^i, v_y^j)$ and $(v_y^j, v_y^i)$, in $A^{\#}$.

     Let $A''$ denote a subset of arcs corresponding to edges from the original graph $G$. $A''$ contains arcs which connect vertices from different subgraphs $G^i$.
   - The source vertex $s$ is connected to every vertex $v_1^i \in V^{\#}, i = 1, \ldots, n$, by arc with zero cost (these arcs belong to $A''$).
   - Every vertex $v \in V^{\#} - \{s, t\}$ is connected to the sink $t$ by an arc (referred to as *exit arc*) with zero cost (these arcs belong to $A''$).

Let us illustrate the above described model with an example (see Fig. 5). Again, the original edge-colored graph $G$ is the one from Figure 1a. $G$ is transformed into a $c$-connected digraph $G^{\#} = (V^{\#}, A^{\#})$ with respect to the color sequence $O = \langle green, red, blue \rangle$. $G^{\#}$ should have 20 vertices collected in 6 3-connected subtraphs $G^i$ (each original vertex from $G$ is represented by a cycle including 3 colored nodes in $G^{\#}$). A directed cycle in every subgraph $G^i$ gives the same order of node colors as that defined in $O$. Two vertices of the same color are connected by two arcs in opposite directions if they are connected by an edge with the same color in the original graph. The source node is connected to all green vertices. For clarity, exit arcs are not represented in the figure.

An algorithm to solve the Longest Path $c$-Connected graph Problem with packing constraints (LPcCP) is based on the IP model and uses the following variables:

- $x_l^{ij}$ – a decision variable corresponding to arc $(v_l^i, v_l^j) \in A''$;
- $x_{lr}^i$ – a decision variable corresponding to arc $(v_l^i, v_r^i) \in A^i$, $A^i \subset A'$;
- $p_l^{ij}$ – cost of arc $(v_l^i, v_l^j) \in A''$.

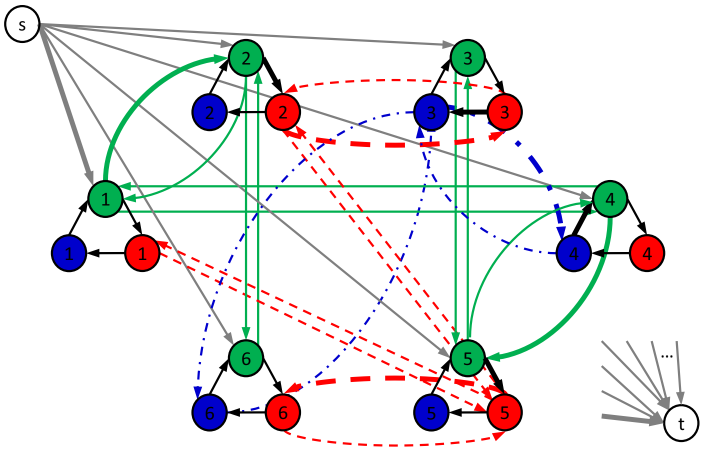


FIGURE 5. The 3-connected digraph constructed upon graph from Figure 1a. The *green-red-blue* path is marked with thick arcs.

The values of decision and cost variables are calculated in the same manner as in the previous models. Given, that $s$ is the source vertex, and $t$ is the sink, the LPcCP is formulated as follows:

$$\text{Maximize} \quad \sum_{(v_l^i, v_l^j) \in A''} p_l^{ij} x_l^{ij} \qquad \text{(OBJ)}$$

subject to:

$$\sum_{(v_l^j, v_l^i) \in A''} x_l^{ji} - \sum_{(v_l^i, v_r^i) \in A'} x_{lr}^i = 0 \qquad \forall v_l^i \in V^\# - \{s,t\} \qquad \text{(C1)}$$

$$\sum_{(v_r^i, v_l^i) \in A'} x_{rl}^i - \sum_{(v_l^i, v_l^j) \in A''} x_l^{ji} = 0 \qquad \forall v_l^i \in V^\# - \{s,t\} \qquad \text{(C2)}$$

$$\sum_{(s, v_1^j) \in A''} x_1^{sj} = 1 \qquad \text{(C3)}$$

$$\sum_{(v_l^i, t) \in A''} x_l^{it} = 1 \qquad \text{(C4)}$$

$$\sum_{(v_l^i, v_r^i) \in A'} x_{lr}^i \leq 1 \qquad i = 1, 2, \ldots, n \qquad \text{(C5)}$$

$$\sum_{(v_l^i, v_l^j) \in A''} x_l^{ij} \leq 1 \qquad i = 1, 2, \ldots, n \qquad \text{(C6)}$$

$$\sum_{(v_l^j, v_l^i) \in A''} x_l^{ji} \leq 1 \qquad i = 1, 2, \ldots, n \qquad \text{(C7)}$$

$$\sum_{(v_l^i, v_l^j) \in \varGamma} x_l^{ij} + \sum_{(v_l^i, v_r^i) \in \varGamma} x_{lr}^i \leq |\,\varGamma\,| - 1 \qquad \varGamma \in \widehat{\varGamma} \qquad \text{(C8)}$$

$$x_l^{ij} \in \{0, 1\} \qquad \forall\, (v_l^i, v_l^j) \in A'' \qquad \text{(C9)}$$

$$x_{lr}^i \in \{0, 1\} \qquad \forall\, (v_l^i, v_r^i) \in A'. \qquad \text{(C10)}$$

Let us recall that since all arcs in $A'$ have zero cost, the objective function contains only variables corresponding to arcs from $A''$. Next, (C1-C4) are the extended flow balance constraints. (C1) states that whenever the flow touches one of the nodes, it must also use one of the arcs in the connected subgraph $G^i$ including this node. (C2) ensures that the flow exits correctly from the visited subgraph $G^i$. These constraints determine that if the path reaches vertex $v_l^i \in G^i$ through $l$-colored arc, then it will go to $v_r^i \in G^i$ and exit the subgraph via $r$-colored arc. (C3) and (C4) constraints ensures that only one unit flows trough the path from $s$ to $t$. (C5-C7) are *packing constraints*. (C5) states that the total flow entering each connected subgraph $G^i$ is at most 1 (in any set $A^i$ at most one arc can be part of the path). From it follows that, for each $i$, at most two vertices in $G^i$ can be visited by the path (C6-C7). Graph $G^{\#}$ is cyclic, thus we need to enforce the elimination of all cycle solutions, by adding (C8), which for each cycle $\Gamma \in \widehat{\Gamma}$ express the corresponding cycle elimination constraint. Summing up. we can state that:

**Proposition 3.4.** *A feasible (optimal) solution of LPcCP is an orderly colored (longest) path in the corresponding edge-colored grapg $G$. An optimal solution of LPcCP composed by $n-1$ vertices is an orderly colored Hamiltonian path in $G$.*

## 4. Algorithms comparison in computational experiment

In this Section, we present computational tests that have been performed on randomly generated edge-colored graphs in order to compare LPnPP (Model 1), LPcPP (Model 2) and LPcCP (Model 3). To prepare test instances we have designed and implemented an instance simulator, which generates edge-colored graphs and stores them in `DIMACS` format [50] (a graph is represented as a list of edges with their colors). Three main parameters govern graph generation: (1) the number of vertices, (2) graph density, and (3) the number of colors. Edges, once generated, are colored randomly, according to a uniform distribution. The instance generator allows for an additional control related with the presence of an orderly colored Hamiltonian path. Thus, we can generate two types of instances: type 1 – non-Hamiltonian graphs (the longest orderly colored path is shorter than $n$, where $n$ is the number of vertices), type 2 – Hamiltonian graphs (containing at least one Hamiltonian path).

The experiments were performed using the Mixed Integer Linear Programming solver Cplex 12.2.0.0 by IBM Ilog with standard settings on a 8-core i7 Intel processor 2.597 GHz with 8GB RAM. The code was developed in C programming language and compiled with GNU CC compiler, running under Microsoft Windows 7.0 with optimization option O3.

Computational experiments have been run with the purpose of enumerating all the orderly colored paths with maximum length. In LPcPP and LPcCP, cycle elimination constraints have been added to the formulation whenever the current solution contained a cycle. A detailed report on the experiments is summarized in Tables 4–7. Each table refers to a different set of tests with increasing graph dimensions (20–100 vertices) and different graph density (10%, 20%, and 30%). Tables 4 and 5 concern the tests with 2-edge-colored graphs, for problems of type 1 and type, respectively. Tables 6 and 7 report on the results for 3-edge-colored graphs of type 1 and 2, respectively. Each table provides graph density (column #1), instance size, *i.e.* number of nodes in edge-colored graph (column #2), model

TABLE 1. Average computing time (seconds).

| No. of vertices | Model 1 | Model 2 | Model 3 |
|---|---|---|---|
| 10 | 0.005 | 0.006 | 0.003 |
| 20 | 0.013 | 0.011 | 0.016 |
| 30 | 0.193 | 0.045 | 0.056 |
| 50 | 505.822 | 45.359 | 65.915 |
| 70 | 3303.272 | 2765.605 | 3053.289 |
| 100 | 2506.478 | 2402.853 | 2404.793 |
| All Problems | 1052.631 | 868.980 | 920.679 |

identifier, *i.e.* 1-LPnPP, 2-LPcPP, 3-LPcCP (column #3), number of vertices in the network (column #4), number of arcs in the network (column #5), number of constraints in the associated ILP (column #6), optimum path length (column #7), the number of paths with maximum length (column #8), number of eliminated cycles (column #9), and computing time in seconds (column #10). When computing time exceeds 1 hour, the algorithm is stopped and the current state of solution reported.

Looking at the results collected in Tables 4–7 we can see that the studied problem poses a computational challenge for bigger instances. We notice a relevant difference between computing times of different models. An additional insight into this issue is provided in Table 1, where we report the average computing times by LPnPP, LPcPP and LPcCP algorithms run for 60 instances with different sizes. Table shows the quick rise of computing times with the number of nodes, in particular we can see long computing times required by Model 1 (LPnPP). For bigger instances the difference tends to reduce, because in most cases the algorithms are stopped after an hour. The table does not report that for many instances Model 1 does not find any optimal solution, while Model 2 and Model 3 always do.

A slightly different picture surfaces when we count the number of instances *solved* within an hour, *i.e.* graphs for which at least one optimal solution has been found before 1 h, even if the whole computation has not finished (not all optimal solutions were found). If we restrict the analysis to 36 large instances (with 50, 70 or 100 nodes) we see that Model 1 often fails in finding the optimal solution, in particular when the graph is dense (for graphs with 30% density Model 1 fails in over half of the cases).

Another comparison of the three models can be made if we consider how many times each model was better than the others in finding the optimal solution. For this comparison, let us declare that model A *wins* over model B if either model A can find all optimal solutions faster than model B, or model A finds more optimal solutions than model B within the given time bound. In case of a tie, both models are considered winners. Figure 6 shows results of this comparison. One can notice the superiority of Model 2 (LPcPP), that accounts for an increasing proportion of wins as the instance size increases. Together with the number of wins (for the whole

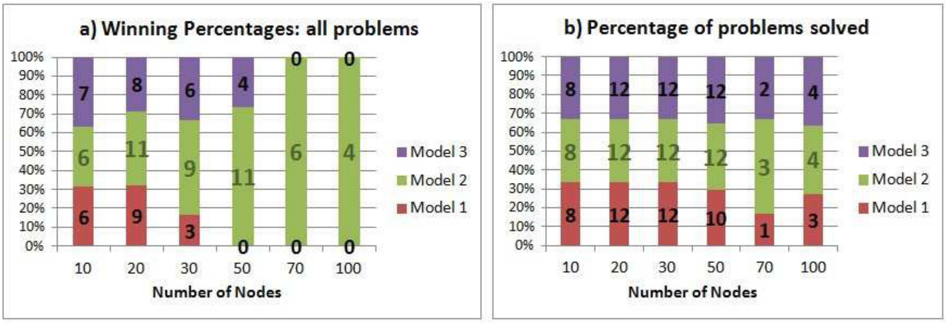


Figure 6. A distribution of (a) “wins” and (b) solved instances.

Table 2. Number of instances solved within 1 h.

| Graph density | Model 1 | Model 2 | Model 3 |
|---|---|---|---|
| 0.1 | 9 | 12 | 12 |
| 0.2 | 7 | 12 | 12 |
| 0.3 | 7 | 12 | 12 |
| All Problems | 23 | 36 | 36 |

set of 60 instances), a twin chart reports the distribution of 36 solved instances for different graph sizes.

The superiority of Model 2 (LPcPP) suggested by Figure 6a is more clearly displayed in Figure 7, where the same distribution is split up for instances with different number of colors (Fig. 7a, Fig. 7b), and according to the problem type (Fig. 7c, Fig. 7d). While the larger instances (100 nodes) are always “won” by Model 2, we see that for smaller ones the distribution is significantly modified. That suggests the absence of interactions among the adopted model and parameters that define the instance (*e.g.* number of colors, problem structure).

In order to confirm the previous results, we have considered another 10 randomly generated instances having the same parameters: 2-edge-colored graphs, non-Hamiltonian, with 100 nodes, and 20% density. From the previous experiments it seems, that such characteristics defines instances with an interesting degree of difficulty. All models have then been applied to these 10 instances and have given results presented in Table 3. Again the experiments show the superiority of Model 2 and Model 3 over Model 1. In this test, processing each instance by every algorithm consumed the whole alloted time. Model 2 seems to run slightly longer, determining, on average, a larger number of solutions than Model 3, and working a little harder in cutting out cycles. When the 10 problems are compared individually, according to the number of optimal solutions and eliminated cycles (see Fig. 8), Model 2 does not exhibit a clear dominance over Model 3.

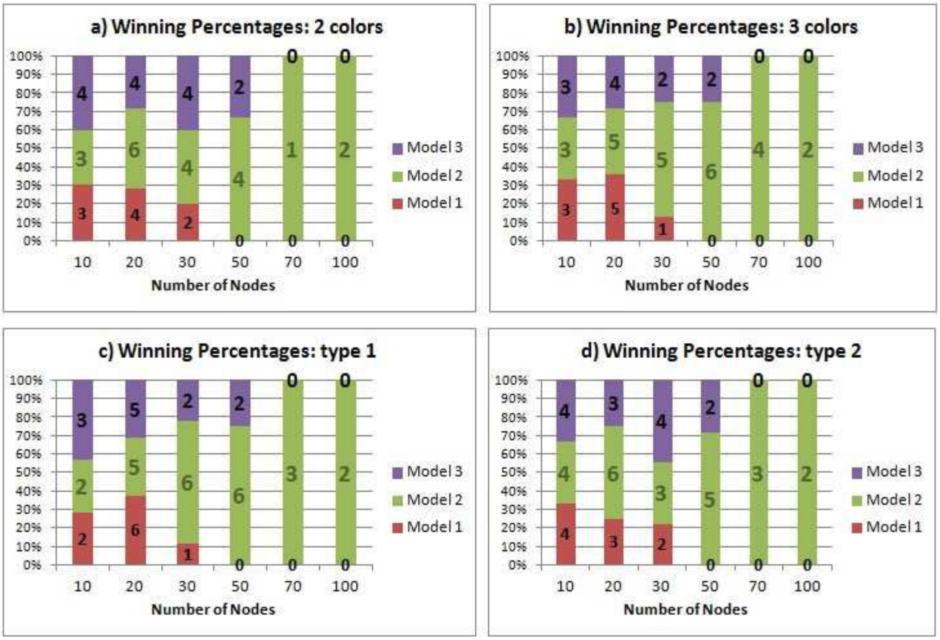


FIGURE 7. A distribution of "wins" for different instances: (a) 2-edge-colored graphs, (b) 3-edge-colored graphs, (c) non-Hamiltonian graphs, (d) Hamiltonian graphs.

TABLE 3. Models' comparison over 10 problems.

| Model id. | No. of solved instances | Avg. no. of longest paths | Avg. no. of cycles | Total no. of "wins" |
|---|---|---|---|---|
| 1 | 2 | 1 | 0 | 0 |
| 2 | 10 | 207.3 | 2855.6 | 9 |
| 3 | 10 | 196.1 | 2703.3 | 8 |

## 5. Summary

The Orderly Colored Longest Path is a problem of finding the longest path in the edge-colored graph, where colors of the traversed edges must follow a predefined color sequence. In the paper, we have presented a survey through applications of OCLP to model various problems in engineering, routing, games, genetics and molecular biology. It appears, that OCLP usability lies especially in modelling problems with some geometric restraints and those with some ordering of actions imposed in advance. Considering solution to the problem in question, we have described three integer programming formulations, LPnPP, LPcPP and LPcCP, which allow for efficient path search in the graph. In order to demonstrate the

Table 4. Experimental results for 2-edge-colored non-Hamiltonian graphs.

| Graph density | Instance size | Model id. | No. of nodes | No. of arcs | No. of constr. | Path length | No. of paths | No. of cycles | Comp. time (s) |
|---|---|---|---|---|---|---|---|---|---|
| | 20 | 1 | 402 | 653 | 423 | 7 | 1 | 0 | 0 |
| 0.1 | 20 | 2 | 41 | 64 | 62 | 7 | 1 | 0 | 0 |
| | 20 | 3 | 41 | 104 | 141 | 7 | 1 | 0 | 0 |
| | 20 | 1 | 402 | 689 | 423 | 7 | 1 | 0 | 0 |
| 0.2 | 20 | 2 | 41 | 68 | 62 | 7 | 1 | 0 | 0 |
| | 20 | 3 | 41 | 108 | 141 | 7 | 1 | 0 | 0.02 |
| | 20 | 1 | 402 | 827 | 424 | 7 | 2 | 0 | 0.02 |
| 0.3 | 20 | 2 | 41 | 82 | 63 | 7 | 2 | 0 | 0.02 |
| | 20 | 3 | 41 | 122 | 142 | 7 | 2 | 0 | 0.02 |
| | 30 | 1 | 902 | 1513 | 933 | 6 | 1 | 0 | 0 |
| 0.1 | 30 | 2 | 61 | 100 | 92 | 6 | 1 | 0 | 0 |
| | 30 | 3 | 61 | 160 | 211 | 6 | 1 | 0 | 0 |
| | 30 | 1 | 902 | 1949 | 936 | 7 | 4 | 0 | 0.08 |
| 0.2 | 30 | 2 | 61 | 130 | 97 | 7 | 4 | 2 | 0.03 |
| | 30 | 3 | 61 | 190 | 216 | 7 | 4 | 2 | 0.06 |
| | 30 | 1 | 902 | 2327 | 936 | 11 | 4 | 0 | 0.25 |
| 0.3 | 30 | 2 | 61 | 156 | 95 | 11 | 4 | 0 | 0.06 |
| | 30 | 3 | 61 | 216 | 214 | 11 | 4 | 0 | 0.08 |
| | 50 | 1 | 2502 | 4427 | 2553 | 9 | 1 | 0 | 0.03 |
| 0.1 | 50 | 2 | 101 | 176 | 152 | 9 | 1 | 0 | 0 |
| | 50 | 3 | 101 | 276 | 351 | 9 | 1 | 0 | 0 |
| | 50 | 1 | 2502 | 6429 | 2553 | 19 | 1 | 0 | 1.08 |
| 0.2 | 50 | 2 | 101 | 258 | 154 | 19 | 1 | 2 | 0.05 |
| | 50 | 3 | 101 | 358 | 353 | 19 | 1 | 2 | 0.11 |
| | 50 | 1 | 2502 | 8535 | 2660 | 34 | 108 | 0 | 553.46 |
| 0.3 | 50 | 2 | 101 | 344 | 366 | 34 | 108 | 107 | 24.4 |
| | 50 | 3 | 101 | 444 | 564 | 34 | 108 | 106 | 36.29 |
| | 70 | 1 | 4902 | 22 743 | 4972 | 69 | 0 | 0 | >3600 |
| 0.1 | 70 | 2 | 141 | 656 | 211 | 69 | 325 | 1022 | >3600 |
| | 70 | 3 | 141 | 796 | 490 | 69 | 227 | 842 | >3600 |
| | 70 | 1 | 4902 | 22 743 | 4972 | 69 | 0 | 0 | >3600 |
| 0.2 | 70 | 2 | 141 | 656 | 211 | 69 | 237 | 1418 | >3600 |
| | 70 | 3 | 141 | 796 | 490 | 69 | 189 | 1231 | >3600 |
| | 70 | 1 | 4902 | 22 743 | 4972 | 69 | 9 | | >3600 |
| 0.3 | 70 | 2 | 141 | 656 | 211 | 69 | 736 | 1996 | >3600 |
| | 70 | 3 | 141 | 796 | 490 | 69 | 737 | 1961 | >3600 |
| | 100 | 1 | 10 002 | 28 013 | 10 107 | 33 | 5 | 0 | 40.08 |
| 0.1 | 100 | 2 | 201 | 562 | 306 | 33 | 5 | 0 | 0.25 |
| | 100 | 3 | 201 | 762 | 705 | 33 | 5 | 0 | 0.27 |
| | 100 | 1 | 10 002 | 43 457 | 10 102 | N.F. | 0 | 0 | >3600 |
| 0.2 | 100 | 2 | 201 | 874 | 2535 | 96 | 509 | 1725 | >3600 |
| | 100 | 3 | 201 | 1074 | 2658 | 96 | 430 | 1528 | >3600 |
| | 100 | 1 | 10 002 | 63 953 | 10 102 | N.F. | 0 | 0 | >3600 |
| 0.3 | 100 | 2 | 201 | 1288 | 3871 | 99 | 136 | 3434 | >3600 |
| | 100 | 3 | 201 | 1488 | 4197 | 99 | 163 | 3334 | >3600 |

Table 5. Experimental results for 2-edge-colored Hamiltonian graphs.

| Graph density | Instance size | Model id. | No. of nodes | No. of arcs | No. of constr. | Path length | No. of paths | No. of cycles | Comp. time (s) |
|---|---|---|---|---|---|---|---|---|---|
| | 20 | 1 | 402 | 641 | 423 | 5 | 1 | 0 | 0 |
| 0.1 | 20 | 2 | 61 | 94 | 82 | 5 | 1 | 0 | 0 |
| | 20 | 3 | 61 | 154 | 181 | 5 | 1 | 0 | 0 |
| | 20 | 1 | 402 | 725 | 423 | 5 | 1 | 0 | 0 |
| 0.2 | 20 | 2 | 61 | 108 | 83 | 5 | 1 | 1 | 0.02 |
| | 20 | 3 | 61 | 168 | 182 | 5 | 1 | 1 | 0 |
| | 20 | 1 | 402 | 881 | 428 | 8 | 6 | 0 | 0.05 |
| 0.3 | 20 | 2 | 61 | 132 | 88 | 8 | 6 | 1 | 0.05 |
| | 20 | 3 | 61 | 192 | 187 | 8 | 6 | 1 | 0.05 |
| | 30 | 1 | 902 | 1515 | 933 | 5 | 1 | 0 | 0.02 |
| 0.1 | 30 | 2 | 91 | 150 | 123 | 5 | 1 | 1 | 0 |
| | 30 | 3 | 91 | 240 | 272 | 5 | 1 | 1 | 0 |
| | 30 | 1 | 902 | 1921 | 933 | 10 | 1 | 0 | 0.02 |
| 0.2 | 30 | 2 | 91 | 192 | 123 | 10 | 1 | 1 | 0 |
| | 30 | 3 | 91 | 282 | 272 | 10 | 1 | 1 | 0.03 |
| | 30 | 1 | 902 | 2363 | 936 | 17 | 4 | 0 | 0.23 |
| 0.3 | 30 | 2 | 91 | 238 | 127 | 17 | 4 | 2 | 0.08 |
| | 30 | 3 | 91 | 328 | 276 | 17 | 4 | 2 | 0.14 |
| | 50 | 1 | 2502 | 4779 | 2554 | 5 | 2 | 0 | 0.03 |
| 0.1 | 50 | 2 | 151 | 286 | 205 | 5 | 2 | 2 | 0 |
| | 50 | 3 | 151 | 436 | 454 | 5 | 2 | 2 | 0.02 |
| | 50 | 1 | 2502 | 6765 | 2553 | 33 | 1 | 0 | 1.73 |
| 0.2 | 50 | 2 | 151 | 408 | 205 | 33 | 1 | 3 | 0.08 |
| | 50 | 3 | 151 | 558 | 454 | 33 | 1 | 3 | 0.08 |
| | 50 | 1 | 2502 | 8883 | 2560 | 44 | 8 | 0 | >3600 |
| 0.3 | 50 | 2 | 151 | 536 | 314 | 44 | 34 | 79 | 70.03 |
| | 50 | 3 | 151 | 686 | 563 | 44 | 34 | 79 | 101 |
| | 70 | 1 | 4902 | 21 485 | 4972 | N.F. | 0 | 0 | >3600 |
| 0.1 | 70 | 2 | 211 | 928 | 688 | 69 | 18 | 122 | >3600 |
| | 70 | 3 | 211 | 1138 | 1034 | 69 | 16 | 107 | >3600 |
| | 70 | 1 | 4902 | 21 485 | 4972 | N.F. | 0 | 0 | >3600 |
| 0.2 | 70 | 2 | 211 | 928 | 688 | 69 | 28 | 346 | >3600 |
| | 70 | 3 | 211 | 1138 | 1034 | 69 | 22 | 229 | >3600 |
| | 70 | 1 | 4902 | 21 485 | 4972 | N.F. | 0 | 0 | >3600 |
| 0.3 | 70 | 2 | 211 | 928 | 688 | 69 | 51 | 356 | >3600 |
| | 70 | 3 | 211 | 1138 | 1034 | 69 | 58 | 346 | >3600 |
| | 100 | 1 | 10 002 | 26 337 | 10 106 | 47 | 4 | 0 | 764.47 |
| 0.1 | 100 | 2 | 301 | 792 | 412 | 47 | 4 | 7 | 2.03 |
| | 100 | 3 | 301 | 1092 | 911 | 47 | 4 | 7 | 3.23 |
| | 100 | 1 | 10 002 | 44 619 | 10 102 | N.F. | 0 | 0 | >3600 |
| 0.2 | 100 | 2 | 301 | 1346 | 1087 | 97 | 53 | 633 | >3600 |
| | 100 | 3 | 301 | 1646 | 1515 | 98 | 47 | 568 | >3600 |
| | 100 | 1 | 10 002 | 58 545 | 10 102 | N.F. | 0 | 0 | >3600 |
| 0.3 | 100 | 2 | 301 | 1768 | 1567 | 99 | 35 | 1131 | >3600 |
| | 100 | 3 | 301 | 2068 | 2354 | 99 | 61 | 1393 | >3600 |

TABLE 6. Experimental results for 3-edge-colored non-Hamiltonian graphs.

| Graph density | Instance size | Model id. | No. of nodes | No. of arcs | No. of constr. | Path length | No. of paths | No. of cycles | Comp. time (s) |
|---|---|---|---|---|---|---|---|---|---|
| | 20 | 1 | 402 | 783 | 423 | 19 | 1 | 0 | 0 |
| 0.1 | 20 | 2 | 41 | 78 | 62 | 19 | 1 | 0 | 0 |
| | 20 | 3 | 41 | 118 | 141 | 19 | 1 | 0 | 0 |
| | 20 | 1 | 402 | 895 | 423 | 19 | 1 | 0 | 0.03 |
| 0.2 | 20 | 2 | 41 | 90 | 62 | 19 | 1 | 0 | 0.02 |
| | 20 | 3 | 41 | 130 | 141 | 19 | 1 | 0 | 0.03 |
| | 20 | 1 | 402 | 1035 | 423 | 19 | 1 | 0 | 0.03 |
| 0.3 | 20 | 2 | 41 | 104 | 62 | 19 | 1 | 0 | 0.02 |
| | 20 | 3 | 41 | 144 | 141 | 19 | 1 | 0 | 0.02 |
| | 30 | 1 | 902 | 1923 | 933 | 29 | 1 | 0 | 0.02 |
| 0.1 | 30 | 2 | 61 | 128 | 92 | 29 | 1 | 0 | 0.02 |
| | 30 | 3 | 61 | 188 | 211 | 29 | 1 | 0 | 0.02 |
| | 30 | 1 | 902 | 2357 | 933 | 29 | 1 | 0 | 0.51 |
| 0.2 | 30 | 2 | 61 | 158 | 92 | 29 | 1 | 0 | 0.05 |
| | 30 | 3 | 61 | 218 | 211 | 29 | 1 | 0 | 0.03 |
| | 30 | 1 | 902 | 2767 | 934 | 29 | 2 | 0 | 0.69 |
| 0.3 | 30 | 2 | 61 | 186 | 97 | 29 | 2 | 4 | 0.2 |
| | 30 | 3 | 61 | 246 | 216 | 29 | 2 | 4 | 0.19 |
| | 50 | 1 | 2502 | 5737 | 2553 | 49 | 1 | 0 | 2.03 |
| 0.1 | 50 | 2 | 101 | 230 | 152 | 49 | 1 | 0 | 0.05 |
| | 50 | 3 | 101 | 330 | 351 | 49 | 1 | 0 | 0.08 |
| | 50 | 1 | 2502 | 8089 | 2558 | 49 | 6 | 0 | 5.16 |
| 0.2 | 50 | 2 | 101 | 326 | 161 | 49 | 6 | 4 | 0.53 |
| | 50 | 3 | 101 | 426 | 351 | 0 | 1 | 4 | 1.2 |
| | 50 | 1 | 2502 | 8089 | 2558 | 49 | 22 | 0 | 328 |
| 0.3 | 50 | 2 | 101 | 326 | 161 | 49 | 22 | 436 | 97 |
| | 50 | 3 | 101 | 426 | 351 | 0 | 22 | 436 | 124 |
| | 70 | 1 | 4902 | 19 937 | 4972 | 0 | 0 | 0 | >3600 |
| 0.1 | 70 | 2 | 141 | 574 | 211 | 69 | 124 | 768 | 1425 |
| | 70 | 3 | 141 | 714 | 490 | 69 | 124 | 894 | 1829 |
| | 70 | 1 | 4902 | 19 937 | 4972 | 0 | | | >3600 |
| 0.2 | 70 | 2 | 141 | 574 | 211 | 69 | 654 | 2276 | >3600 |
| | 70 | 3 | 141 | 714 | 490 | 69 | 559 | 2009 | >3600 |
| | 70 | 1 | 4902 | 26 783 | 4972 | 0 | | | >3600 |
| 0.3 | 70 | 2 | 141 | 772 | 211 | 69 | 257 | 2820 | 2206 |
| | 70 | 3 | 141 | 912 | 490 | 69 | 257 | 2975 | 2341 |
| | 100 | 1 | 10 002 | 33 279 | 10 122 | 99 | 20 | 0 | 454.98 |
| 0.1 | 100 | 2 | 201 | 668 | 341 | 99 | 20 | 20 | 6.54 |
| | 100 | 3 | 201 | 868 | 741 | 99 | 20 | 21 | 12.9 |
| | 100 | 1 | 10 002 | 51 865 | 10 102 | N.F. | 0 | 0 | >3600 |
| 0.2 | 100 | 2 | 201 | 1044 | 3412 | 99 | 151 | 2960 | >3600 |
| | 100 | 3 | 201 | 1244 | 3755 | 99 | 162 | 2893 | >3600 |
| | 100 | 1 | 10 002 | 66 155 | 10 102 | N.F. | 0 | 0 | >3600 |
| 0.3 | 100 | 2 | 201 | 1332 | 3983 | 99 | 143 | 3539 | >3600 |
| | 100 | 3 | 201 | 1532 | 4358 | 99 | 123 | 3535 | >3600 |

TABLE 7. Experimental results for 3-edge-colored Hamiltonian graphs.

| Graph density | Instance size | Model id. | No. of nodes | No. of arcs | No. of constr. | Path length | No. of paths | No. of cycles | Comp. time (s) |
|---|---|---|---|---|---|---|---|---|---|
| 0.1 | 20 | 1 | 402 | 663 | 423 | 19 | 1 | 0 | 0 |
| | 20 | 2 | 61 | 98 | 82 | 19 | 1 | 0 | 0 |
| | 20 | 3 | 61 | 158 | 181 | 19 | 1 | 0 | 0 |
| 0.2 | 20 | 1 | 402 | 763 | 423 | 19 | 1 | 0 | 0 |
| | 20 | 2 | 61 | 114 | 82 | 19 | 1 | 0 | 0 |
| | 20 | 3 | 61 | 174 | 181 | 19 | 1 | 0 | 0.03 |
| 0.3 | 20 | 1 | 402 | 921 | 423 | 19 | 1 | 0 | 0.03 |
| | 20 | 2 | 61 | 138 | 82 | 19 | 1 | 0 | 0 |
| | 20 | 3 | 61 | 198 | 181 | 19 | 1 | 0 | 0.02 |
| 0.1 | 30 | 1 | 902 | 1669 | 933 | 29 | 1 | 0 | 0 |
| | 30 | 2 | 91 | 166 | 122 | 29 | 1 | 0 | 0.02 |
| | 30 | 3 | 91 | 256 | 271 | 29 | 1 | 0 | 0.03 |
| 0.2 | 30 | 1 | 902 | 2117 | 933 | 29 | 1 | 0 | 0.05 |
| | 30 | 2 | 91 | 212 | 122 | 29 | 1 | 0 | 0.03 |
| | 30 | 3 | 91 | 302 | 271 | 29 | 1 | 0 | 0.03 |
| 0.3 | 30 | 1 | 902 | 2643 | 933 | 29 | 1 | 0 | 0.45 |
| | 30 | 2 | 91 | 266 | 122 | 29 | 1 | 0 | 0.05 |
| | 30 | 3 | 91 | 356 | 271 | 29 | 1 | 0 | 0.06 |
| 0.1 | 50 | 1 | 2502 | 5417 | 2553 | 49 | 1 | 0 | 0.3 |
| | 50 | 2 | 151 | 326 | 202 | 49 | 1 | 0 | 0.03 |
| | 50 | 3 | 151 | 476 | 451 | 49 | 1 | 0 | 0.03 |
| 0.2 | 50 | 1 | 2502 | 7321 | 2553 | 49 | 1 | 0 | 2.04 |
| | 50 | 2 | 151 | 442 | 202 | 49 | 1 | 0 | 0.14 |
| | 50 | 3 | 151 | 592 | 451 | 49 | 1 | 0 | 0.17 |
| 0.3 | 50 | 1 | 2502 | 7321 | 2553 | 49 | 16 | 0 | 822 |
| | 50 | 2 | 151 | 442 | 202 | 49 | 16 | 87 | 352 |
| | 50 | 3 | 151 | 592 | 451 | 49 | 16 | 87 | 528 |
| 0.1 | 70 | 1 | 4902 | 23 141 | 4972 | N.F. | 1 | 0 | 34 |
| | 70 | 2 | 211 | 1000 | 1330 | 69 | 1 | 422 | 8 |
| | 70 | 3 | 211 | 1210 | 2516 | 69 | 1 | 422 | 2 |
| 0.2 | 70 | 1 | 4902 | 23 141 | 4972 | N.F. | 0 | 0 | >3600 |
| | 70 | 2 | 211 | 1000 | 1330 | 69 | 48 | 642 | 1824 |
| | 70 | 3 | 211 | 1210 | 2516 | 69 | 23 | 328 | >3600 |
| 0.3 | 70 | 1 | 4902 | 23 141 | 4972 | N.F. | 0 | 0 | >3600 |
| | 70 | 2 | 211 | 1000 | 1330 | 69 | 78 | 971 | >3600 |
| | 70 | 3 | 211 | 1210 | 2516 | 69 | 166 | 1720 | >3600 |
| 0.1 | 100 | 1 | 10 002 | 30 693 | 10 103 | 99 | 1 | 0 | 17.08 |
| | 100 | 2 | 301 | 924 | 402 | 99 | 1 | 0 | 0.45 |
| | 100 | 3 | 301 | 1224 | 901 | 99 | 1 | 0 | 0.59 |
| 0.2 | 100 | 1 | 10 002 | 45 081 | 10 102 | N.F. | 0 | 0 | >3600 |
| | 100 | 2 | 301 | 1360 | 905 | 99 | 18 | 486 | >3600 |
| | 100 | 3 | 301 | 1660 | 1660 | 99 | 33 | 727 | >3600 |
| 0.3 | 100 | 1 | 10 002 | 60 525 | 10 102 | N.F. | 0 | 0 | >3600 |
| | 100 | 2 | 301 | 1828 | 1552 | 99 | 27 | 1124 | >3600 |
| | 100 | 3 | 301 | 2128 | 2854 | 99 | 70 | 1884 | >3600 |

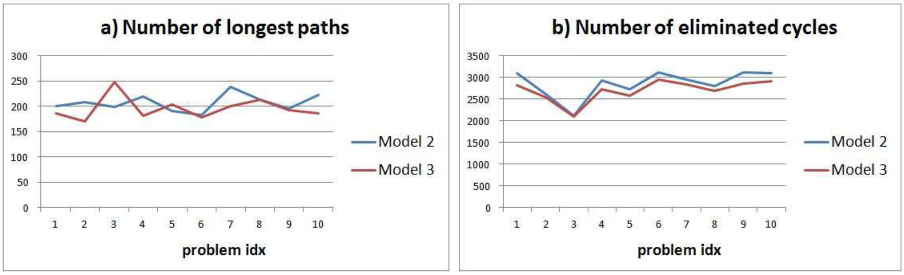


FIGURE 8. Number of (a) optimal solutions and (b) eliminated cycles for 10 random 2-edge-colored, non-Hamiltonian graphs with 100 nodes and 20% density. Results for Models 2 and 3.

efficiency of the approaches, we have tested all the alternative network models over a set of randomly generated test instances with different characteristics. The obtained results have been used to compare the models, leading to the conclusions that LPcPP and LPcCP with iterative cycle separation appear to perform much better than LPnPP. The tests shown also that problems with reasonable size, and up to 3 colors, can be solved in a reasonable time.

We propose that the future research can follow two main directions. The test cases should be extended to include more problem-specific data, and some problems with the real data (*e.g.* experimental) should be examined. On the other hand, the refinement of the IP formulations can be made, combined with a more sophisticated procedure for the separation of orderly colored cycles.

*Acknowledgements.* The authors have been partially supported by the cooperative programme 2010-2012 between the National Research Council of Italy (CNR) and the Polish Academy of Sciences (PAN) and by the FLAGSHIP *InterOmics* project (PB.P05) funded by the Italian MIUR and CNR institutions. We also acknowledge the support from the National Science Centre, Poland (2012/05/B/ ST6/03026, 2012/06/A/ST6/00384).

## REFERENCES

[1] A. Abouelaoualim, K.Ch. Das, L. Faria, Y. Manoussakis, C. Martinhon, R. Saad, Paths and trails in edge-colored graphs. *Theor. Comput. Sci.* **409** (2008) 497–510.

[2] R.W. Adamiak, J. Blazewicz, P. Formanowicz, Z. Gdaniec, M. Kasprzak, M. Popenda, M. Szachniuk, An algorithm for an automatic NOE pathways analysis of 2D NMR spectra of RNA duplexes. *J. Comput. Biol.* **11** (2004) 163–179.

[3] R.K. Ahuja, T.L. Magnanti, J.B. Orlin, *Network Flows: Theory, Algorithms and Applications.* Prentice Hall, Englewood Cliffs, New Jersey (1993).

[4] J. Bang-Jensen, G. Gutin, Alternating cycles and trails in 2-edge-colored complete multigraphs. *Disc. Math.* **188** (1998) 61–72.

[5] J. Bang-Jensen, G. Gutin, Properly colored hamiltonian paths in edge-coloured complete graphs. *Disc. Appl. Math.* **82** (1998) 247–250.

[6] B. Beauquier, S.Perennes and M. Syska, Efficient access to optical bandwidth routing and grooming in WDM networks: state-of-the-art survey. RESCCO Report IST-2001-33135, Universite de Nice-Sophia Antipolis (2002).

[7] A. Benkouar, Y.G. Manoussakis, V.Th. Paschos and R. Saad, On the complexity of some hamiltonian and eulerian problems in edge-colored complete graphs. *Lect. Not. Comput. Sci.* **557** (1991) 190–198.

[8] D.P. Bertsekas, *Network Optimization: Continuous and Discrete Models*. Athena Scientific (1998).

[9] J. Bialogrodzki, Path coloring and routing in graphs, *in Contemporary Mathematics*, edited by M. Kubale. American Mathematical Society (2004) 139–152.

[10] J. Blazewicz, M. Szachniuk and A. Wojtowicz, Evolutionary approach to NOE paths assignment in RNA structure elucidation. *Proceedings of the 2004 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology* (2004) 206–213.

[11] J. Blazewicz, M. Szachniuk, A. Wojtowicz, RNA tertiary structure determination: NOE pathways construction by tabu search. *Bioinformatics* **21** (2005) 2356–2361.

[12] J. Blazewicz, P.L. Hammer, P. Lukasiak, Predicting secondary structures of proteins. *IEEE Eng. Med. Biol. Mag.* **24** (2005) 88–94.

[13] B. Bollobás, *Graph Theory. An Introductory Course*, Springer-Verlag, New York (1979).

[14] B. Bollobás, P. Erdos, Alternating hamiltonian cycles. *Isr. J. Math.* **23** (1976) 126–131.

[15] J.A. Bondy, U.S.R Murty, *Graph Theory with Applications*, Macmillan Press, London (1976).

[16] F. Carrabs, R. Cerulli, M. Gentili, The Labeled Maximum Matching Problem. *Comput. Oper. Res.* **36** (2009) 1859–1867.

[17] R. Cerulli, A. Fink, M. Gentili, S. Voss, Metaheuristics comparison for the minimum labelling spanning tree problem, in *The Next Wave on Computing, Optimization, and Decision Technologies*, edited by B.L. Golden, S. Raghavan and E.A. Wasil. Springer, New York (2005) 93–106.

[18] R. Cerulli, A. Fink, M. Gentili, S. Voss, Extensions of the Minimum Labelling Spanning Tree Problem. *J. Telecom. Inform. Technol.* **4** (2006) 39–45.

[19] B.V. Cherkassky, A.V. Goldberg, Negative-cycle detection algorithms. *Math. Prog.* **85** (1999) 277–311.

[20] W.S. Chou, Y. Manoussakis, O. Megalakaki, M. Spyratos, Zs. Tuza, Paths through fixed vertices in edge-colored graphs. *Mathematiques, Informatique et Sciences Humaines* **127** (1994) 49–58.

[21] M.C. De Cola, G. Felici, M. Szachniuk, The Orderly Colored Longest Path Problem. *CNR-IASI Technical Report* **29** (2012).

[22] A. Conrad, T. Hindrichs, H. Morsy, I. Wegener, Solution of the Knight's Hamiltonian Path Problem on Chessboards. *Disc. Appl. Math.* **50** (1994) 125–134.

[23] D. Dorninger, Hamiltonian circuits determining the order of chromosomes. *Disc. Appl. Math.* **50** (1994) 159–168.

[24] J. Feng, H.-E. Giesen, Y. Guo, G.Gutin, T. Jensen, A. Rafiey, Characterization of edge-colored complete graphs with properly colored hamiltonian paths. *J. Graph Theor.* **53** (2006) 333–346.

[25] P. Festa, The shortest path tour problem: problem definition, modeling, and optimization. *Proceedings of INOC*'2009 (2009) 1–7.

[26] G. Gallo, S. Pallottino, Shortest path methods: a unifying approach. *Math. Program. Stud.* **26** (1986) 38–64.

[27] M.R. Garey, D.S. Johnson, *Computers and Intractability: A guide to the theory of NP-completeness*. W.H. Freeman & Co., San Francisco CA (1979).

[28] L. Gourvès, A. Lyra, C. Martinhon, J. Monnot, F. Protti, On s-t paths and trails in edge-colored graphs. *Electron. Notes Discrete Math.* **35** (2009) 221–226.

[29] M. Kchikech, O. Togni, Paths coloring algorithms in mesh networks. *Lect. Not. Comput. Sci.* (2003) 193–202.

[30] A. Kosowski, Path assignment with wavelength constraints. Ph.D. Thesis, Gdansk University of Technology, Poland (2006).

[31] H. Li, G. Wang, S. Zhou, Long alternating cycles in edge-colored complete graphs. *Lect. Not. Comput. Sci.* **4613** (2007) 305–309.

[32] F. Luccio, C. Mugnia, Hamiltonian paths on a rectangular chess-board. *Proceedings of 16th Annual Allerton Conference* (1978) 161–173.

[33] Y. Manoussaskis, Alternating paths in edge-colored complete graphs. *Disc. Appl. Math.* **56** (1995) 297–309.

[34] B.R. Myers, Enumeration of tour in hamiltonian rectangular lattice graphs. *Math. Mag.* **54** (1981) 19–23.

[35] C. Nomikos, Coloring in Graphs, Ph.D. Thesis, Dept.of Electrical and Computer Engineering, National Technical University of Athens, Greece (1997).

[36] P.A. Pevzner, DNA physical mapping and alternating eulerian cycles in colored graphs. Algorithmica **13** (1995) 77-105.

[37] P.A. Pevzner, *Computational Molecular Biology: an algorithmic approach.* MIT Press (2000).

[38] M. Popenda, M. Szachniuk, M. Antczak, K.J. Purzycka, P. Lukasiak, N. Bartol, J. Blazewicz, R.W. Adamiak, Automated 3D structure composition for large RNAs. *Nucleic Acids Res.* **40** (2012) e112.

[39] M. Popenda, M. Blazewicz, M. Szachniuk, R.W. Adamiak, RNA FRABASE version 1.0: an engine with a database to search for the three-dimensional fragments within RNA structures. *Nucleic Acids Res.* **36** (2008) D386-D391.

[40] M. Popenda, M. Szachniuk, M. Blazewicz, S. Wasik, E.K. Burke, J. Blazewicz, R.W. Adamiak, RNA FRABASE 2.0: an advanced web-accessible database with the capacity to search the three-dimensional fragments within RNA structures. *BMC Bioinformatics* **11** (2010) 231.

[41] M. Szachniuk, M. Popenda, R.W. Adamiak, J. Blazewicz, An assignment walk through 3D NMR spectrum. *Proceedings of the 2009 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology* (2009) 215–219.

[42] M. Szachniuk, M. Malaczynski, E. Pesch, E.K. Burke, J. Blazewicz, MLP accompanied beam search for the resonance assignment problem. *J. Heuristics* **19** (2013) 443–464.

[43] M. Szachniuk, M.C. De Cola, G. Felici, D. de Werra, J. Blazewicz, Optimal pathway reconstruction on 3D NMR maps. *Disc. Appl. Math.* (2013) submitted for publication.

[44] S. Szeider, Finding paths in graphs avoiding forbidden transitions. *Disc. Appl. Math.* **126** (2003) 239–251.

[45] I.L. Tseng, H.W. Chen, C.I. Lee, Obstacle-aware longest path routing with parallel MILP solvers. *Proceedings of WCECS* (2010).

[46] Y. Wang, Y. Desmedt, Edge-colored graphs with applications to homogeneous faults. *Inf. Proc. Lett.* **111** (2011) 634–641.

[47] J.J. Watkins, R.L. Hoenigman, Knight's tours on a torus. *Mathematics* **70** (1997) 175–184.

[48] T. Zok, M. Popenda, M. Szachniuk, MCQ4Structures to compute similarity of molecule structures. *Central Eur. J. Oper. Res.* (2013), doi:10.1007/s10100-013-0296-5.

[49] A. Yeo, A note on alternating cycles in edge-colored graphs. *J. Comb. Theor. B* **69** (1997) 222–225.

[50] http://prolland.free.fr/works/research/dsat/dimacs.html.