ISTITUTO DI ANALISI DEI SISTEMI ED INFORMATICA
"Antonio Ruberti"
CONSIGLIO NAZIONALE DELLE RICERCHE

M.C. De Cola, G. Felici, M. Szachniuk

THE ORDERLY COLORED LONGEST PATH PROBLEM

R. 29, 2012

Maria Cristina De Cola – Department of Statistics, University of Rome La Sapienza, Rome, Italy, mariacristina.decola@uniroma1.it.

Giovanni Felici – Institute of Systems Analysis and Computer Science "A. Ruberti", National Research Council, Rome, Italy, giovanni.felici@iasi.cnr.it.

Marta Szachniuk – Institute of Bioorganic Chemistry, Polish Academy of Sciences and Poznan University of Technology, Poland, marta.szachniuk@cs.put.poznan.pl.

ISSN: 1128–3378

Collana dei Rapporti dell'Istituto di Analisi dei Sistemi ed Informatica "Antonio Ruberti", CNR

viale Manzoni 30, 00185 ROMA, Italy

tel. ++39-06-77161
fax ++39-06-7716461
email: iasi@iasi.cnr.it
URL: http://www.iasi.cnr.it

Abstract

We consider the problem of finding a longest path in a graph whose edges are colored with c colors, under the constraint that the edges of the path follow a given order of colors. The problem is referred to as the Orderly Colored Longest Path on a c-edge colored graph (OCLP). We show its relations with similar problems already studied and consider three alternative integer programming models for this problem.

The three models are formulated by means of max flow problems on a directed graph with packing constraints over certain partitions of the nodes.

This problem has not been modeled and studied in previous literature. A recent and relevant application is related with the interpretation of Nuclear Magnetic Resonance experiments. Besides, applications can be found in transportation, games, and grid graphs. The interestingness of this problem is motivated by the fact that it models the relations of consecutive edges of the path. All three integer programming formulations have been implemented and compared on a set of randomly generated instances.

Key words: Longest path problem; alternating path; edge colored graph; integer programming.

1. Introduction

The Orderly Colored Longest Path problem (OCLP) is the problem of finding the longest path on a c-edge-colored graph such that edges composing the path follow a given order of colors. In particular, if $c = 2$ the edges of the path are alternating in only two colors and the path will be called *alternating*. The problem of finding an alternating path in c-edge-colored graphs is NP-complete since it generalizes the Hamiltonian path problem in non-colored graphs [18].

In the following we assume basic knowledge of both standard graph-theoretic terminology and network flow problems (see [9], [3] and [8]). All graphs and digraphs considered are finite and simple.

Let $G = (V, E)$ be a graph with $|V| = n$ and $|E| = m$. G is a *c-edge-colored graph* if to each edge of G is assigned a color (label) $l \in \{1, \dots, c\}$, $c \geq 2$. A *walk* from s to t in G is a sequence of vertices and edges $\{v_0, e_0, v_1, e_1, \dots, e_k, v_{k+1}\}$ such that $v_0 = s$, $v_{k+1} = t$ and $e_i = (v_i, v_{i+1})$ for $i = 0, \dots, k$. A *path* from s to t in G is a walk without repeated vertices, i.e., $v_i \neq v_j$ for $i \neq j$. The *length* of a path is the number of its edges or a weighted function of its edges and/or vertices.

As above mentioned, when $c = 2$ a path is called *alternating* and the color of the edges along the path is alternated. For $c > 2$, a path is called *properly colored* if any two adjacent edges differ in color.

The problem of finding in G a properly edge-colored path was first solved for two colors [4] and then extended to include any number of colors (see [28]). Although it has been proven to be NP-Complete (see [19] and [1]), there exist algorithms that find a properly edge-colored path (or a Hamiltonian properly edge-colored path) in a polynomial time, when certain conditions are satisfied; for instance, when no alternating cycle is present in the graph [19] and [31], or when the path is forced to cross a given set of vertices (see [14]).

Turning to the problem of checking whether there exists in G a properly colored path, many results show that this problem is polynomially time solvable only if G is a not directed graph, even if there are adjacent edges in G with same color; (see for details [1], [5], [22] and [20]).

Graphs with colored edges are often used to model decision problems, for example when a certain subset of the edges (e.g., a spanning tree, or a matching) is required to use the minimum number of colors (see [10], [11], and [12]).

Paths colored according to proper coloration have applications in various fields, especially in problems modeled using grid graphs or *Mesh graphs*. For instance, if $c = 2$ then an alternating path problem is obtained from a rectangular grid where horizontal edges are colored with one color and vertical edges with the other one. Grid graphs are used to model city blocks where intersections are vertices and streets are edges [24]. Analogously, a grid structure may represent a two-dimensional memory array where memory can be accessed moving down or across from each cell; here vertices correspond to each cell and edges connect adjacent cells. In this case finding a longest path in the grid corresponds to accessing all the data [23].

Of interest are also the applications on grid graphs that contain obstacles (e.g. forbidden vertices), as in the Longest Path Routing problem discussed in [29].

Path constrained in edge colors are also used to model chessboard problems, in particular the knight's L move, that generates the knight's tour problem [30]. Modeling the chessboard as a grid graph, the different colors on the edges allows to model the problem of seeking an Hamiltonian (or longest) path composed of knight's moves.

Similarly, edge colored schemes could be useful to model particular routing problems: pick-up and delivery (where different types of pick-up and deliveries must alternate over the edges) or electric vehicles routing, where loading and discharging arcs must alternate, or separated waste collection routing.

In the last two decades c-edge colored graphs have also been proposed to model problems in genetic and molecular biology [25]. For instance, in [15] is discussed the existence of alternating Hamiltonian circuits that determine the spatial order of the chromosomes within haploid chromosome complements; whereas in [26] alternating Eulerian cycles are used to represent the solutions of the Double Digest Problem (i.e., the problem of constructing a physical maps of DNA sequences, also belonging to the NP class).

More recently, the correlation signals occurring between the nuclei of RNA molecule during Nuclear Magnetic Resonance (NMR) experiments have been represented with a graph model ([7]-[2]). In [27] the authors present an enumerative algorithm that solves this assignment problem in the proposed graph model, assuming that each edge of a c-edge colored graph is colored according to the type of interaction represented as a transition between any pair of cross-peaks in a 3D NMR spectrum. Thus, an orderly colored path along the vertices of G is the reconstruction of a transfer pathway between the cross-peaks of the spectral graph, needed for shape reconstruction.

In this paper we describe three formulations for OCLP where a directed 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 first model is based on a longest path problem in a n-partite graph with a number of partitions equal to the number of vertices of the original graph. The second model is a transformation of the latter, where the number of partite sets depends on the number of colors. The third model is formulated on a n-partite graph, where each partition is a c-connected subgraph of the original graph.

The proposed formulations for OCLP problem share some characteristics with the Shortest Path Tour Problem (SPTP) described in [16], shown to be solvable either with Shortest Path algorithms (see among others [3], [13] and [17]) or with dynamic programming [6]. Indeed, SPTP is a polynomial-time reduction into a source-single 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 the constraints on the partite sets: in the former they are covering constraints, while in the latter they are packing constraints.

Moreover, the graph of OCLP is not complete, and many well-known algorithms cannot be applied to test the existence of an Hamiltonian path or to find a longest path (see for example [21], where a longest path in a complete multipartite digraph with n vertices is found in $O(n^3)$).

The paper is structured as follows. In Section 2 we introduce a simple example that will be used in the remainder of the paper. The 3 following sections (Section 3, Section 4, Section 5) treat in detail the three proposed models. In Section 6 we provide the results of some experimental runs with the three formulations over a set of simulated problems. In Section 7, we draw some conclusions and describe some possible lines of future works.

2. A simple example of OCLP

Let $G = (V, E)$ be a c-edge-colored graph, where to each edge of G is assigned a color from $C = \{1, \ldots, c\}$, $c \geq 2$, and $|V| = n$ and $|E| = m$. We refer with c_{ij} to the cost of the edge $(v^i, v^j) \in E$.

In the following three formulations for OCLP associated with G are discussed. The descriptions of the models are based on the simple example depicted in Figure 1; we consider the small graph in Figure 1.b. composed of 6 nodes, 10 edges, and 3 colors; the three colors *green* (4 arcs), *red* (4 arcs) and *blue* (2 arcs), have to appear in a path following the sequence given in Figure 1.a. The longest orderly colored path is composed of 6 nodes (in the order: 1,2,...,6) and 5 edges, as represented in Figure 1.c. For simplicity, we assume that a feasible path must always start from a given color (green, in this case); such assumption can be easily removed in the models described, either below by the addition of zero-cost arcs, or by solving c instances of the same model.

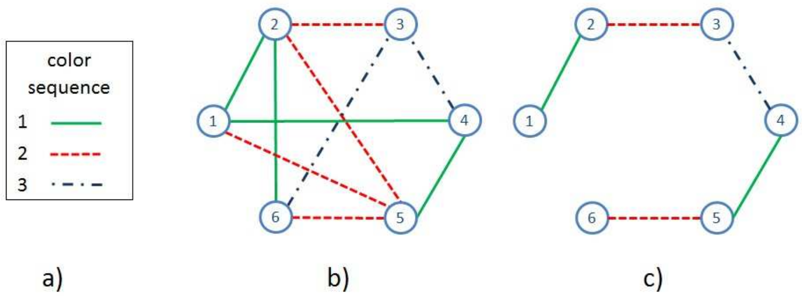

Figure 1: A simple example of 3-edge colored graph. Edges have to follow a given sequence of colors (a). The graph is composed of 6 nodes and 10 edges (b). The longest orderly colored path touches all nodes as in (c). For better readability, green arcs are solid, red arcs are dotted, blue arcs are dashes.

3. Longest Path over acyclic n-partite graph with packing constraints (LP-nPP)

Let $D = (V', A)$ be a n-partite digraph where:

1. The set of vertices $V = \{v^1, v^2, \ldots, v^n\}$ from G is repeated n times in V'; thus, we have a n-partition of V' such that

   - $V' = V^1 \cup V^2 \cup \cdots \cup V^n$ and $V^i \cap V^j = \emptyset$, for each $i, j \in V : i \neq j$.
   - $V^l = \{v_1^l, v_2^l, \ldots, v_n^l\}$ for each $l \in \{1, \ldots, n\}$.
   - $|V'| = n^2$.

   Note that a generic subset V^l contains n copies of vertex $v^l \in V$, e.g., if $l = 3$ then $V^3 = \{v_1^3 = v^3, \ldots, v_n^3 = v^3\}$. We refer to each V^l as a *partite set.*

2. We define a *level set* as $L_r = \{v_r^1, v_r^2, \ldots, v_r^n\}$ such that any vertex v_r^l belongs to a different partite set, thus $V^l \cap L_r = \{v_r^l\}$ for each $r = 1, \ldots, n$ and $l \in \{1, \ldots, n\}$. Note that the vertices in each level set L_r are copies of the vertices of V in G, e.g., if $r = 3$ then $L_3 = \{v_3^1 = v^1, v_3^2 = v^2, \ldots, v_3^n = v^n\}$.

3. Let (L_r, L_{r+1}) be a pair of two consecutive level sets. Each edge $(v^i, v^j) \in E$ is replaced by two arcs in A, (v_r^i, v_{r+1}^j) and (v_r^j, v_{r+1}^i), in order to connect (L_r, L_{r+1}). Both arcs have the same cost c_{ij} associated with $(v^i, v^j) \in E$.

4. Each pair of level sets (L_r, L_{r+1}) is connected by arcs of the same color.

   On the basis of the color sequence requested from the path, in the $n-1$ level set pairs the arcs are recursively alternated in color. E.g., if $c = 2$ then all pairs of arcs $(v_1^i, v_2^j) \in A$ and $(v_1^j, v_2^i) \in A$ in (L_1, L_2) are from those edges $(v^i, v^j) \in E$ with color 1 in G; all pairs of arcs $(v_2^i, v_3^j) \in A$ and $(v_2^j, v_3^i) \in A$ in (L_2, L_3) are from those edges $(v^i, v^j) \in E$ with color 2 in G; all pairs of arcs $(v_3^i, v_4^j) \in A$ and $(v_3^j, v_4^i) \in A$ in (L_3, L_4) are from those edges $(v^i, v^j) \in E$ with color 1 in G and so on.

5. One source vertex s and one destination vertex t are added to D, thus:

   - $|V'| = n^2 + 2$;
   - $L_0 = \{s\}$;
   - $L_{n+1} = \{t\}$.

   Arcs with null weight connect s with each vertex in L_1, and each vertex in V' to t (the latter referred to as *exit arcs*).

Example 1. *We consider an edge-colored directed graph $G = (V, E)$ where V contains the $n = 6$ vertices of the edge-colored graph in Figure 1 and the associated edges. Then, the corresponding network has 38 vertices and 6 partite sets as in Figure 2. The grey horizontal box represents the first partite set containing 6 copies of the vertex 1, whereas the grey vertical box represents the fifth level set containing one copy for each vertex of V. The arcs in each level set pair are alternated on the basis of the three colors* {green, red, blue}*. Note that in the figure the arcs represented by light grey lines are the exit arcs; for better readability in Figure 2 only exit arcs from nodes belonging to the first and the last partite sets have been represented; although the model accounts for exit arcs from each one of the 36 nodes to the sink node.*

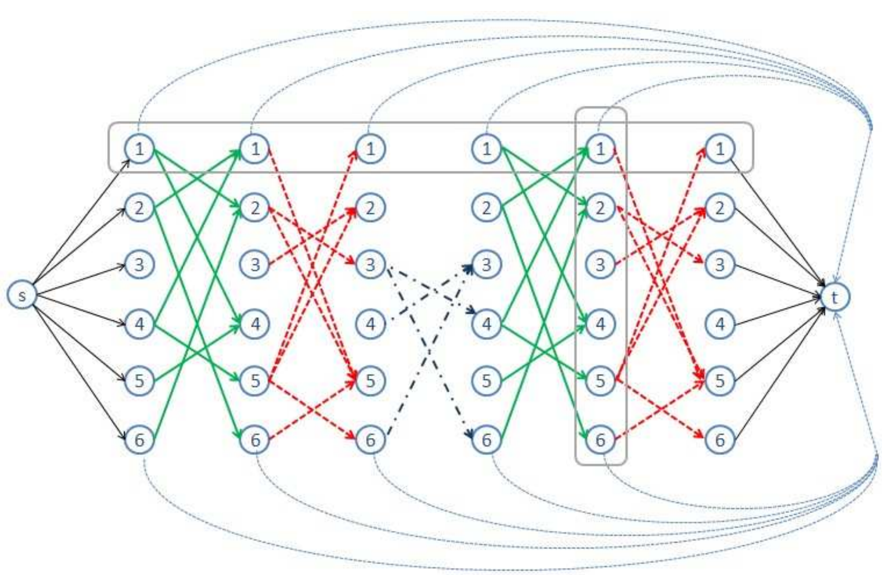

Figure 2: The 6-partite graph constructed from the edge-colored graph of Figure 1.

Proposition 1. *Let $D = (V', A)$ be a n-partite digraph as above defined. Then a longest path from s to t in D such that successive vertices belong to different partite sets V^l is an orderly colored longest path in G.*

We now describe the corresponding Integer Programming (IP) formulation. Let $D = (V', A)$ be a directed graph as above described. For each arc $(v_r^i, v_{r+1}^j) \in A$ we associate the decision variable x_r^{ij}, where $x_r^{ij} = 1$ if there is flow on the arc (v_r^i, v_{r+1}^j) and $x_r^{ij} = 0$ otherwise; moreover, with $c_r^{ij} = 1$ we refer to the arc cost (recall that the so-called *exit* arcs from each node to the sink are assigned zero cost). The subscript r is used to distinguish the same arcs repeated in different level set pairs. Let s and t be the source and the destination vertices, respectively. The longest path n-partite problem with packing constraints (LPnPP) is an optimization model formulated as follows:

$$\begin{array}{lllr}
\text{Maximize} & \sum_{(v_r^i, v_{r+1}^j) \in A} c_r^{ij} x_r^{ij} & & (OBJ) \\
\text{subject to:} & & & \\
& \sum_{(v_r^i, v_{r+1}^j) \in A} x_r^{ij} - \sum_{(v_{r-1}^j, v_r^i) \in A} x_r^{ji} = 0 & \forall v_r^i \in V' - \{s, t\} & (C1) \\
& \sum_{(s, v_1^j) \in A} x_1^{sj} = 1 & & (C2) \\
& \sum_{(v_r^i, t) \in A} x_r^{it} = 1 & & (C3) \\
& \sum_{\substack{(v_{r-1}^j, v_r^i) \in A \\ v_r^i \in V^l}} x_r^{ji} \leq 1 & l = 1, 2, \ldots, n & (C4) \\
& x_r^{ij} \in \{0, 1\} & \forall (v_r^i, v_{r+1}^j) \in A & (C5)
\end{array}$$

where the constraints (C1) are the classical flow balance constraints of a network flow problem. The first term of these constraints represents the total flow emanating from a vertex v_r^i and 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 constraints (C2) and (C3). Thus, the solution of the problem will send 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 to each set V^l is at most 1, e.g., for each set V^l at most one vertex can be visited by the path. In other words, these constraints ensure that only one copy of the same vertex of the original graph is visited by the path; e.g., they make sure that the expansion of the vertices needed to manage properly the sequence of the colors does not create unfeasible paths w.r.t. the original problem. According to these consideration, it is easy to state the following:

Proposition 2. *A feasible (optimal) solution of LPnPP is an orderly colored (longest) path in G; an optimal solution of LPnPP composed by $n - 1$ vertices is an orderly colored Hamiltonian path in G.*

4. Longest Path over cyclic c-Ppartite graph with packing constraints (LPcPP)

In this second formulation we significantly simplify the dimension of the graph used to search the longest paths introducing cycle elimination constraints in the formulation. These constraints can be separated, making this approach potentially interesting from the computational point of view, as detailed in the experimental section.

Let $D = (V', A)$ be a c-partite digraph where:

1. The set of vertices $V = \{v^1, v^2, \ldots, v^n\}$ from G is repeated c times in V'. Thus, the n-partition of V' is such that

- $V' = V^1 \cup V^2 \cup \cdots \cup V^n$ and $V^i \cap V^j = \emptyset$, for each $i, j \in V : i \neq j$.
- $V^l = \{v_1^l, v_2^l, \ldots, v_c^l\}$ for each $l \in \{1, \ldots, n\}$.
- $|V'| = n \times c$.

Note that a generic subset V^l contains c copies of v^l in V, e.g., if $l = 3$ then $V^3 = \{v_1^3 = v^3, \ldots, v_c^3 = v^3\}$. We call it a *partite set*.

2. We define a *level set* as $L_r = \{v_r^1, v_r^2, \ldots, v_r^n\}$ such that any vertex v_r^l belongs to a different partite set, thus $V^l \cap L_r = \{v_r^l\}$ for each $r = 1, \ldots, c$ and $l \in \{1, \ldots, n\}$.

   We note that all level sets L_r with $r = 1, \ldots, c$ form a different partition of V' such that $V' = L_1 \cup l_2 \cup \cdots \cup L_c$ and $L_i \cap L_j = \emptyset$ for each $i, j \in C : i \neq j$. Indeed, if we consider this partition then D is a c-partite digraph. For clarity, we keep the same mathematical notation adopted in LPnPP. Also in this formulation the vertices in each level set L_r are the same copies of the vertices of V in G, e.g., if $r = 3$ then $L_3 = \{v_3^1 = v^1, v_3^2 = v^2, \ldots, v_3^n = v^n\}$.

3. Let (L_r, L_{r+1}) be a pair of two consecutive level sets. Arcs are directed from L_r to L_{r+1}, $r = 1, \ldots, c$. Each edge $(v^i, v^j) \in E$ is replaced by two arcs in D, (v_r^i, v_{r+1}^j) and (v_r^j, v_{r+1}^i), in order to connect the consecutive level sets (L_r, L_{r+1}). Both arcs have the same cost c_{ij} from the edge $(v^i, v^j) \in E$.

   Differently from LPnPP, here the number of level set pairs is $c-1$, instead of $n-1$. Indeed, the last pair of level sets is composed by (L_c, L_1), because the arcs labeled with color c connecting the last level set L_c to the first one L_1. Below we will indicate a level set pair with (L_p, L_q) and its arc with (v_p^i, v_q^j), where $q = (p\%c) + 1$.

4. Each pair of level sets (L_p, L_q) contains arcs with same color in G.

   On the basis of the color sequence requested from the path, in the c level set pairs the arcs are recursively alternated in color. E.g., if $c = 3$ then all pairs of arcs $(v_1^i, v_2^j) \in A$ and $(v_1^j, v_2^i) \in A$ in (L_1, L_2) are from those edges $(v^i, v^j) \in E$ with color 1 in G; all pairs of arcs $(v_2^i, v_3^j) \in A$ and $(v_2^j, v_3^i) \in A$ in (L_2, L_3) are from those edges $(v^i, v^j) \in E$ with color 2 in G and all pairs of arcs $(v_3^i, v_1^j) \in A$ and $(v_3^j, v_1^i) \in A$ in (L_3, L_1) are from those edges $(v^i, v^j) \in E$ with color 3 in G.

5. One source vertex s and one destination vertex t are added to D, thus $|V'| = nc + 2$; Arcs with null weight connect s with each vertex in L_1, each vertex in V' to t (the latter referred to as *exit arcs*).

Example 2. *As in the example 1 we consider a the edge-colored graph of Figure 1. Here, the corresponding network has* 20 *vertices,* 6 *partite sets* V^l*, only* 3 *level sets* L_1*,* L_2*, and* L_3*, as shown in Figure 3. The arcs between the three level sets are directed from* L_1 *to* L_2 *if their color label is* green *in* G*; are directed from* L_2 *to* L_3 *if their color label is* red *in* G*; and then are directed from* L_3 *to* L_1 *if their color label is* blue *in* G*; Also here, arcs represented by light grey lines are the exit arcs, and only a part of them is represented for readability (exit arcs should be present from each node to the sink node).*

Also in this case it is possible to formulate the problem as an IP. Let $D = (V', A)$ be a directed graph as above described. Given that in this model there are not arcs repetitions, for each arc

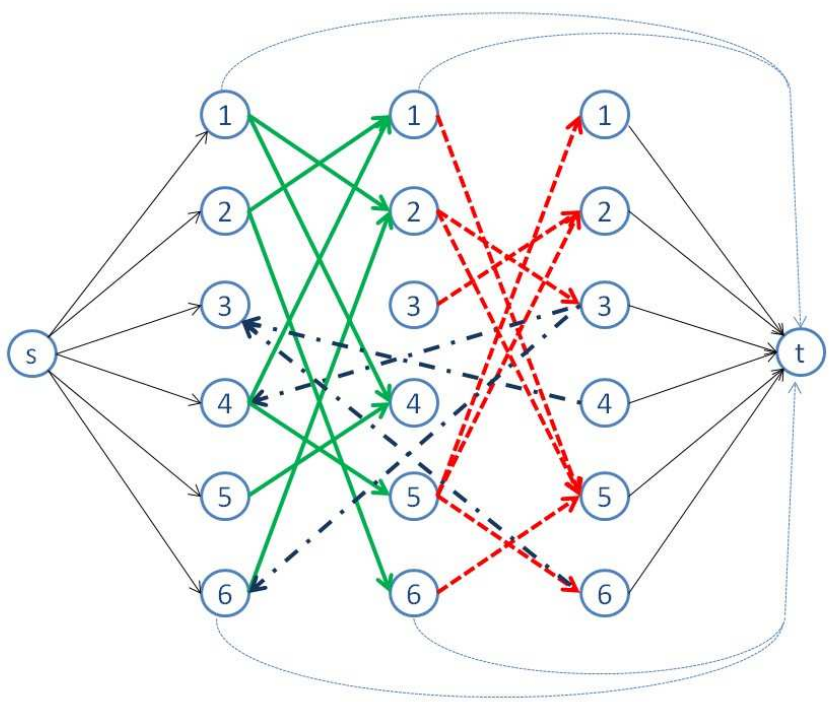

Figure 3: The 3-partite graph constructed from the edge-colored graph in Figure 1.

$(v_p^i, v_q^j) \in A$ we refer with x^{ij} to the decision variables, and with $c_r^{ij} = 1$ to the arc cost (null on exit arcs). Let s and t be the source and the destination vertices; the longest path on a c-partite graph with packing constraints (LPcPP) is thus formulated as follows:

$$
\begin{array}{llll}
\text{Maximize} & \sum_{(v_p^i, v_q^j) \in A} c^{ij} x^{ij} & & (OBJ) \\
\text{subject to:} & & & \\
& \sum_{(v_p^i, v_q^j) \in A} x^{ij} - \sum_{(v_q^j, v_p^i) \in A} x^{ji} = 0 & \forall v_p^i \in V' - \{s, t\} & (C1) \\
& \sum_{(s, v_1^j) \in A} x^{sj} = 1 & & (C2) \\
& \sum_{(v_p^i, t) \in A} x^{it} = 1 & & (C3) \\
& \sum_{\substack{(v_p^i, v_q^j) \in A \\ v_p^i \in V^l}} x^{ij} \leq 1 & l = 1, 2, \ldots, n & (C4) \\
& \sum_{(v_p^i, v_q^j) \in \Gamma} x^{ij} \leq \mid \Gamma \mid -1 & \Gamma \in \widehat{\Gamma} & (C5) \\
& x^{ij} \in \{0, 1\} & \forall\, (v_p^i, v_q^j) \in A & (C6)
\end{array}
$$

As in LPnPP, the constraints (C1-C3) are the flow balance constraints and the packing constraints (C4) state that in any set V^l at most one vertex can be visited from the path. Since the graph is not acyclic, we need to enforce the separations of all the orderly colored cycles; in the formulation this is achieved with constraints (C5), that for each cycle $\Gamma \in \widehat{\Gamma}$ expresses cycle elimination constraints.

Cycle elimination constraints are not added in the initial formulation of the problem when the problem is solved, but they are iteratively separated if a cycle appears in the current solution.

Therefore:

Proposition 3. *A feasible (optimal) solution of LPcPP is an orderly colored (longest) path in G; an optimal solution of LPcPP composed by $n-1$ vertices is an orderly colored Hamiltonian path in G.*

5. Longest Path over cyclic c-connected graph with packing constraints (LPCPP)

Here we propose a third ILP model for OCLP. This formulation requires cycle elimination constraints as the previous one, but manages in a possibly more efficient way the nodes and the arcs used to impose the correct order of colors in the path. Although the number of vertices still depends on the number of colors, here the graph is not really c-partite. For the sake of clarity, here we need to change the notation adopted in the first two models.

Let $D = (V', A)$ be a digraph where:

1. The set of vertices $V = \{v^1, v^2, \dots, v^n\}$ from G is repeated c times in V' such that
   - $V' = V^1 \cup V^2 \cup \dots \cup V^n$ and $V^i \cap V^j = \emptyset$, for each $i, j \in V : i \neq j$.
   - $V^i = \{v^i_1, v^i_2, \dots, v^i_c\}$ for each $i \in \{1, \dots, n\}$.
   - $|V'| = n \times c$.

   Note that a generic subset of vertices V^i contains c repetition of $v^i \in V$.

2. The arcs set is formed by:
   - n disjoint subsets A^i, where $|A^i| = c$ for each $i \in \{1, \dots, n\}$; such subsets are collected into $A' = A^1 \cup A^2 \cup \dots \cup A^n$. The c arcs in subset A^i define a cycle on the V^i nodes.
   - subset A'', composed of a pair of directed arcs associated with each edge of the original graph G; such arcs connect nodes belonging to different subsets $(V_i, V_j, i \neq j)$ as described in the following.

3. A subgraph $G^i = (V^i, A^i)$ is a directed cycle. Any two vertices v^i_l and v^i_r in V^i with $l, r \in \{1, \dots, c\}$ are connected by one of the arcs in A^i directed from v^i_l to v^i_r; thus, the last arc is precisely (v^i_c, v^i_1) and it ends a cycle in A^i. The sequence of the arcs depends on the requested color sequence; i.e., if the path crosses the arc $(v^i_l, v^i_r) \in A^i$ in D, then the corresponding path in G is changing in color from l to r. Arcs in A^i are used as bridge between two different colors and thus they have zero cost.

   E.g., if $c = 3$ then the connected subgraph G^i will be formed by $V^i = \{v^i_1, v^i_2, v^i_3\}$ and $A^i = \{(v^i_1, v^i_2), (v^i_2, v^i_3), (v^i_3, v^i_1)\}$: a simple example of the expansion of the nodes into these cycles of dimension 3 is in Figure 4.

4. Each edge $(v^i, v^j) \in E$ with color label $l \in C$ is replaced by two arcs in A'', (v^i_l, v^j_l) and (v^j_l, v^i_l). Therefore, the set A'' only contains arcs with one end in a connected subgraph and the other end in another connected subgraph.

   Both arcs have the same cost c_{ij} from the edge $(v^i, v^j) \in E$. E.g., if $c = 3$ then all pairs of arcs (v^i_1, v^j_1) and $(v^j_1, v^i_1) \in A''$ are from those edges $(v^i, v^j) \in E$ with color 1 in G; all pairs of arcs (v^i_2, v^j_2) and $(v^j_2, v^i_2) \in A''$ are from those edges $(v^i, v^j) \in E$ with color 2 in G, and all pairs of arcs (v^i_3, v^j_3) and $(v^j_3, v^i_3) \in A''$ are from those edges $(v^i, v^j) \in E$ with color 3 in G.

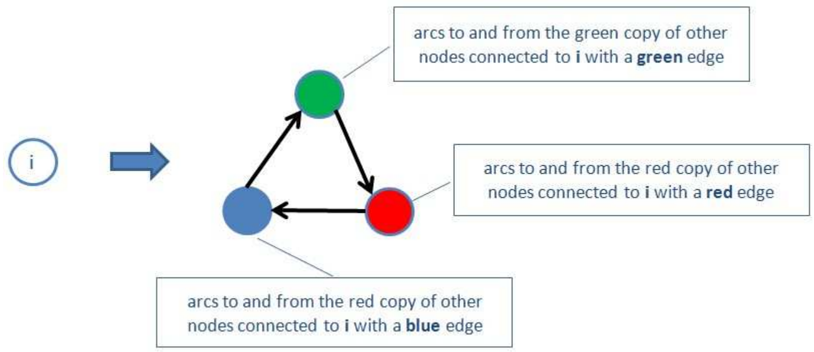

Figure 4: The expansion of a node of the original graph into the cycles with 3 nodes and 3 arcs.

5. One source vertex s and one destination vertex t are added to D, thus $|V'| = n \times c + 2$; Arcs with zero cost connect s with each vertex $v_1^i, i = 1, \dots, n$, and all vertices of V' to t (the latter referred to as *exit arcs*). Also these arcs belong to A''.

Example 3. *We consider the same example of the previous sections. The resulting network* $D = (V', A)$ *has* 20 *vertices. Each original vertex is represented by a cycle of 3 colored nodes. Two nodes of the same color are connected by two arcs in opposite directions only if they are connected by an edge of the same color in the original graph. The source node is connected to all nodes of color green; exit arcs to sink are not represented in Figure 5 but they need to be considered for all nodes.*

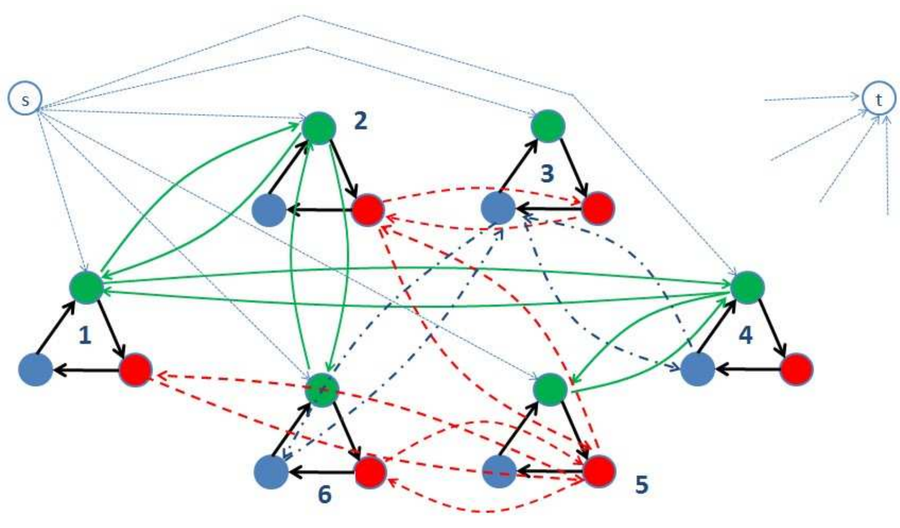

Figure 5: The 3-connected graph constructed from the edge-colored graph in Figure 1

Let us describe the corresponding IP formulation of LPCPP. Let $D = (V', A)$ be the directed graph depicted above. Given that in this model there are arcs connecting both vertices of different connected subgraphs and vertices belong to the same connected subgraph, we use two types of decision variables: for each arc $(v_l^i, v_l^j) \in A''$, where $v_l^i \in V^i$ and $v_l^j \in V^j$, we refer to

x_l^{ij}; for each arc $(v_l^i, v_r^i) \in A^i$ we refer to x_{lr}^i. Here both decision variables and arc costs are binaries, and with $c_l^{ij} = 1$ we refer to the cost of all the arcs in A''. Let s and t be the source and the destination vertices; the longest path on a c-connected graph with packing constraints (LPCPP) is thus formulated as follows:

$$
\begin{array}{llll}
\text{Maximize} & \sum_{(v_l^i, v_l^j) \in A''} c_l^{ij} x_l^{ij} & & (OBJ) \\
\text{subject to:} & & & \\
& \sum_{(v_l^j, v_l^i) \in A''} x_l^{ji} - \sum_{(v_l^i, v_r^i) \in A'} x_{lr}^i = 0 & \forall v_l^i \in V' - \{s,t\} & (C1) \\
& \sum_{(v_r^i, v_l^i) \in A'} x_{rl}^i - \sum_{(v_l^i, v_l^j) \in A''} x_l^{ji} = 0 & \forall v_l^i \in V' - \{s,t\} & (C2) \\
& \sum_{(s, v_1^j) \in A''} x_1^{sj} = 1 & & (C3) \\
& \sum_{(v_l^i, t) \in A''} x_l^{it} = 1 & & (C4) \\
& \sum_{(v_l^i, v_r^i) \in A^i} x_{lr}^i \leq 1 & i = 1, 2, \ldots, n & (C5) \\
& \sum_{(v_l^i, v_l^j) \in A''} x_l^{ij} \leq 1 & i = 1, 2, \ldots, n & (C6) \\
& \sum_{(v_l^j, v_l^i) \in A''} x_l^{ji} \leq 1 & i = 1, 2, \ldots, n & (C7) \\
& \sum_{(v_l^i, v_l^j) \in \Gamma} x_l^{ij} + \sum_{(v_l^i, v_r^i) \in \Gamma} x_{lr}^i \leq \mid \Gamma \mid -1 & \Gamma \in \widehat{\Gamma} & (C8) \\
& x_l^{ij} \in \{0,1\} & \forall \, (v_l^i, v_l^j) \in A'' & (C9) \\
& x_{lr}^i \in \{0,1\} & \forall \, (v_l^i, v_r^i) \in A' & (C10)
\end{array}
$$

Recalling that arcs internal to the connected subgraphs have zero cost, in the objective function only appear the variables corresponding to the arcs in A''. The extended flow balance constraints are (C1-C4). They state that whenever the flow touches one of the nodes, it must also use one of the arcs in the connected component where that nodes belongs (C1). Then, constraints(C2) ensure that the flow exits correctly from that component. These constraints ensure that anytime the path reaches a vertex through an arc with color l then the next arc will connect it to another vertex v_r^i, which is the head of an arc with color r. The constraints (C3) and (C4) impose that only one unit flows trough the path from the source s until the destination t. In this model three kinds of *packing constraints* (C5-C7) have been formulated. These constraints state that the total flow entering each connected subgraph G^i is at most 1, e.g., in any set A^i at most one arc can be part of the path (C5); from it follows that at most two vertices in V^i - for each i - can be visited by the path (C6-C7). As in LPcPP the graph is not acyclic, thus we need to enforce the separations of all the cycles adding the constraints (C8), that for each cycle $\Gamma \in \widehat{\Gamma}$ express the corresponding cycle elimination constraint.

Also in this case, we can state that:

Proposition 4. *A feasible (optimal) solution of LPCPP is an orderly colored (longest) path in G; an optimal solution of LPCPP composed by $n-1$ vertices is an orderly colored Hamiltonian path in G.*

6. Experimental Results

The three proposed models have been tested on a number of randomly generated edge-colored graphs. An instance simulator has been designed and implemented to produce an edge-colored graph $G = (V, E)$ as a list of the edges and of their color, in `DIMACS` format [32]. Three main parameters govern the generation of the graphs: 1) the number of nodes, 2) the density of the graph, and 3) the number of colors of the edges. Edges, once generated, are randomly assigned to one of the colors according to a uniform distribution.

The graph generator allows an additional control step related with the presence of an orderly colorable hamiltonian path; thus, we generate two types of problems:

- In *type 1* problems the orderly colored paths in the generated graph may have any length, up to $n-1$ (where n is the number of nodes);
- In problems of *type 2* the generation procedure is designed to create at least one orderly colored path of length $n-1$ (i.e., orderly colored hamiltonian path(s)).

The experiments were run with the purpose of enumerating all the orderly colored paths of maximum length; the problems were thus solved iteratively, cutting out, at each iteration, the current optimal solution with a constraint, until the value of the new optimal solution was decreased, confirming that all paths of maximal length had been found.

Moreover, in models LPcPP and LPCPP, cycle elimination constraints are added to the formulation whenever the current IP solution contains one or more cycles.

Experiments were run 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 2.597GHz 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.

A detailed report of the experiments is summarizes in Tables 4-7, each table being referred to a different set of experiments with graph of increasing dimension (from 20 to 100 nodes) and different arc densities (here 3 levels are considered: 10%, 20%, and 30%). Graphs with 2 colors on the edges are reported in Table 4 and Table 5, for problems of type 1 and type 2, respectively. Tables 6 and Table 7 are instead reporting 3-edge colored graphs of type 1 and 2, respectively.

Each table reports in the first column the density of the graph, followed by the number of nodes, the number of vertices and arcs (or variables) in the expanded partite graph, the number of constraints in the associated ILP (this number does not include all the constraints added to cut out the current solution, iteratively), the length of the optimal path and the number of paths of maximum length that have been found. The number of cycles that have been cut out during the computation follows in the tables; such figure is applicable only for experiments run with model 2 (LPcPP) and model 3 (LPCPP). The last column reports the total computation time in seconds; when solution time exceeded 1 hour, the algorithm was halted and the current state of the solution reported.

The results reported in Tables 4-7 bring to evidence some considerations. First, we note that the studied problem, besides being difficult from a theoretical standpoint, also poses a computational challenge when its reference dimensions (e.g., the number of nodes) reaches reasonable sizes.

Moreover, we notice a relevant difference in the computation times required for solution by the three models.

An additional insight in this direction is provided in Table 1, where we report the average solution times for the 3 models, for different sizes of the graph (the average is taken over all the dimensions considered, e.g., density, type, number of colors). The table shows very clearly the quick rise of solution times with the number of nodes, and, accordingly, the higher solution times required by Model 1 (LPnPP). Not surprisingly, for large problems the difference in average time for the 3 models tends to reduce, as all the 3 models end up spending the whole hour that is given as upper bound on computation time. Nevertheless, the table does not report the fact that in many cases Model 1 is not able to find any optimal solution, differently from Models 2 and 3.

Number of Nodes	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

Table 1: Solution time (secs.) for LPnPP, LPcPP, and LPCPP; average over all problems, by size

A slightly different picture surfaces when we count the number of *solved* problems, e..g, problems where at least one optimal solution has been found, even it has not been possible to list all optimal solutions within the time bound. If we restrict the analysis to problems of large size (with 50, 70 or 100 nodes) we see that Model 1 often fails in finding an optimal solution, failing more often when the problem is presumably more difficult (i.e., the graph is denser): with graph with density of 30% LPnPP fails more than half of the times.

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

Table 2: Number of solved problems (problems where at least 1 optimal solution is found within 3600 seconds of computation) for LPnPP, LPcPP, and LPCPP, by graph density; analysis limited to problem with **50, 70, 100 nodes**.

A more accurate comparisons of the three models can be performed by considering how many times a model produces the best solution for a given problem. In this case we declare that model A *wins* over model B if either model A is able to find all optimal solutions quicker than model B, or models A finds more optimal solutions than model B within the computation upper bound. In the case of ties, the "win" is assigned to both models. From Figure 6 emerges the superiority of Model 2 (LPcPP), that accounts for an increasing proportion of wins as the problem size increases. Together with the number of wins, a twin chart reports the distribution of solved problem for problem size.

The superiority of LPcPP suggested by Figure 6(a) is reinforced by the 4 charts of Figure 7, where the same distribution is split up for the number of colors (Figure 7(a), Figure 7(b)), and for problem type (Figure 7(c), for problems without the injection of the Hamiltonian path, and 7(d) for problems that contain at least 1 Hamiltonian path): while the larger problems (100 nodes) are always "won" by Model 2, we see that in the smaller problems the distributions are not significantly modified, suggesting the absence of interactions among the model adopted and the other parameters that define the problem (number of colors, problem structure).

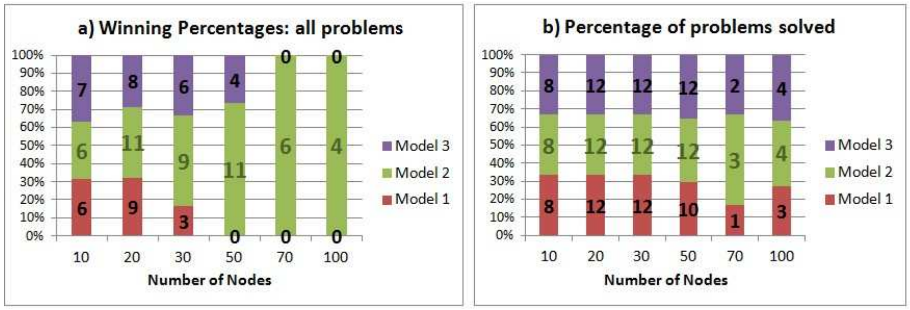

Figure 6: a) Distribution of "wins" for the 3 models; b) Distribution of problems solved (at least one optimal solution found) by the 3 models.

To further reinforce these last considerations, we have considered 10 different randomly generated problems with the same characteristics: 2 colors, 100 nodes, density 20%, no injected Hamiltonian path. These characteristics define, from the previous experiments, problems of an interesting degree of difficulty. The 3 models have then been applied to these 10 problems, in order to validate the previous results.

The experiments still confirm the superiority of Models 2 and 3 (LPcP and LPCPP) over Model 1 (LPnPP), while providing a weaker dominance of Model 2 over Model 3 (see Table 3): the respective number of wins is very close (8 to 9). Not surprisingly, in this case no instances are closed and all problems burn all the allotted time. Model 2 seems to run a slightly longest way, 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 of the eliminated cycles found (see Figure 8), Model 2 does not exhibit a clear dominance over Model 3.

Model	Number of solved (1) problems	Average number of long. paths	Average number of cycles	Total number of "wins"(2)
1	2	1	0	0
2	10	207.3	2855.6	9
3	10	196.1	2703.3	8

Table 3: Performance comparison over 10 **100 nodes, type 1** problems, with **2 colors** and graph **density 0.2**. (1) where at least one optimal solution was found; (2) A win is determined by shorter time in closing the problem or by largest number of optimal solutions found if problem is not closed within the time limit.

7. Conclusions

In this paper we have presented the problem of the Orderly Colored Longest Path and showed how it can be adapted to model different types of real problems. We have defined three alternative formulations, that differ in the way the orderly colored paths are represented over a graph. In order to demonstrate the efficacy of this approach, we have tested our models over a set of

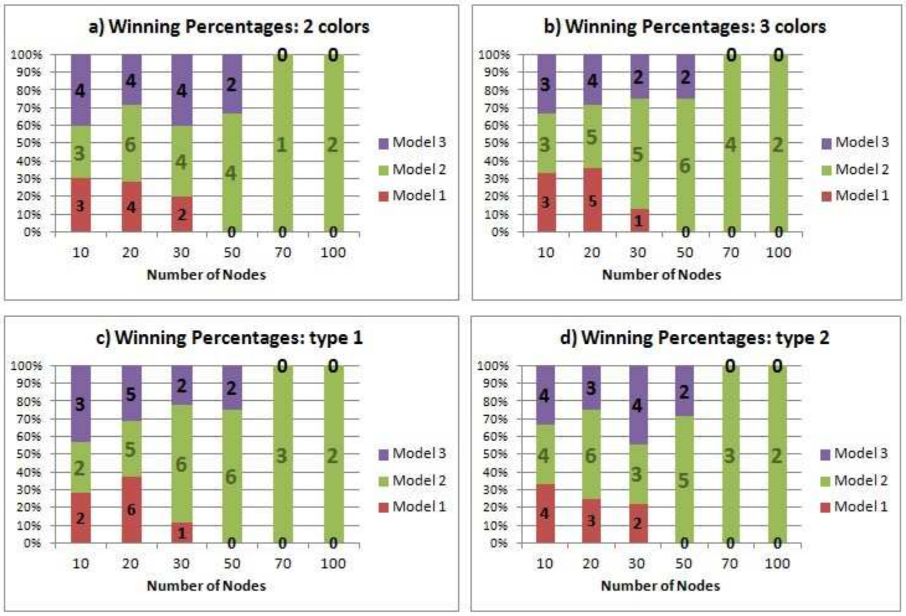

Figure 7: Distribution of "wins" for the 3 models acording to number of colors and problem type: problems with **2 colors** (a), problems with **3 colors** (b), **type 1** problems, without injection of the Hamiltonian path (c), **type 2** problem, with the injection of the Hamiltonian path (d).

randomly generated test problems with different characteristics.

The results obtained are used to compare the three models, leading to the conclusions that 2 of the formulations, where cycle separation is performed iteratively adding constraints to a lighter formulation, appear to perform much better.

To any extent, the experiments conducted bring to evidence the validity of the approach; problems of reasonable size with 2 and 3 colors can be solved in large but yet reasonable solution time.

Future work will follow two main directions. On the one hand, the extension of the test cases and the use of real problems from the applications mentioned, in particular for the case of the reconstruction of the magnetization sequence of nuclear magnetic resonance experiments; on the other hand, the refinement of the formulations of the models that obtained the better performances, combined with a more sophisticated procedure for the separation of orderly colored cycles.

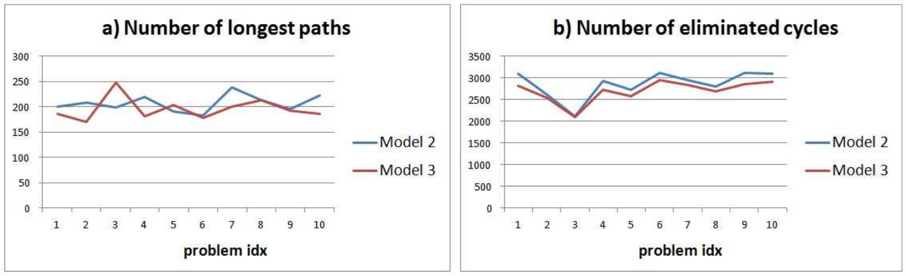

Figure 8: Number of optimal solutions (a) and eliminated paths (b) for 10 randomly generated problems of **large size** (100 nodes, 0.2 graph density, 2 colors, no Hamiltonian path injected). Analysis limited to Models 2 and 3 (LPcPP and LPCPP).

8. Acknowledgements

The authors are 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.

9. Acknowledgements

18.

References

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

[2] Adamiak, R.W., Blazewicz, J., Formanowicz, P., Gdaniec, Z., Kasprzak, M., Popenda, M., Szachniuk, M., An Algorithm for an Automatic NOE Pathways Analysis of 2D NMR Spectra of RNA Dueplexes, *Journal of Computational Biology*, 11, 163-179 (2004).

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

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

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

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

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

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

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

[10] Carrabs, F., Cerulli, R., Gentili, M., The Labeled Maximum Matching Problem, *Computers and Operations Research*, 36, 6, 1859-1867 (2009).

[11] Cerulli, R., Fink, A., Gentili, M., Voss, S., Extensions of the Minimum Labelling Spanning Tree Problem, *Journal of Telecommunications and Information Technology*, 4, 39-45 (2006).

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

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

[14] Chou, W.S., Manoussakis, Y., Megalakaki, O., Spyratos, M., Tuza, Zs., Paths through fixed vertices in edge-colored graphs, *Math. Inf. Sci.hum.*, 127, 49-58 (1994).

[15] Dorninger, D., Hamiltonian circuits determining the order of chromosomes, *Discrete Appl. Math.*, 50, 159-168 (1994).

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

[17] Gallo, G., Pallottino, S., Shortest path methods: A unifying approach, *Math. Programming Studies*, 26, 38-64 (1986).

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

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

[20] Gutin, G., Sudakov, B., Yeo, A., Note on alternating directed cycles, *Discrete Math.*, 191, 101-107 (1998).

[21] Gutin, G., Finding a Longest path in a Complete multipartite digraph, *SIAM Journal Discrete Math.*, 6, 270-273 (1993).

[22] Li, H., Wang, G., Zhou, S., Long alternating cycles in edge-colored complete graphs, *Lecture Notes in Computer Science*, 4613, 305-309 (2007).

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

[24] Myers, B.R., Enumeration of tour in Hamiltonian rectangular Lattice graphs, *Mathematical Magazine*, 54, 19-23 (1981).

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

[26] Pevzner, P.A., DNA Physical mapping snd alternating eulerian cycles in colored graphs, *Algorithmica*, 13, 1-2, 77-105 (1995).

[27] Szachniuk, M., Popenda, M., Adamiak, R.W., Blazewicz, J., An Assignment Walk through 3D NMR Spectrum, in *Proceedings of CIBCB'2009*, 215-219 (2009).

[28] Szeider, S., Finding paths in graphs avoiding forbidden transitions, *Discrete Appl. Math.*, 126, 239-251 (2003).

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

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

[31] Yeo, A., A note on alternating cycles in edge-colored graphs, *Journal of Combinatorial Theory*, Series B-69, 222-225 (1997).

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

Graph Density	Cross Peaks	Model	Nodes	Arcs (var.)	N. of Constr.	Path Length	N. of Paths	N. of Cycles	Total 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	22743	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	22743	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	22743	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	10002	28013	10107	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	10002	43457	10102	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	10002	63953	10102	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 4: Experimental results for LPnPP, LPcPP, and LPCPP models, for problems of **type 1** (no injected Hamiltonian path) with **2 colors** on the edges.

Graph Density	Cross Peaks	Model	Nodes	Arcs (var.)	N. of Constr.	Path Length	N. of Paths	N. of Cycles	Total Time (s)
0.1	20	1	402	641	423	5	1	0	0
	20	2	61	94	82	5	1	0	0
	20	3	61	154	181	5	1	0	0
0.2	20	1	402	725	423	5	1	0	0
	20	2	61	108	83	5	1	1	0.02
	20	3	61	168	182	5	1	1	0
0.3	20	1	402	881	428	8	6	0	0.05
	20	2	61	132	88	8	6	1	0.05
	20	3	61	192	187	8	6	1	0.05
0.1	30	1	902	1515	933	5	1	0	0.02
	30	2	91	150	123	5	1	1	0
	30	3	91	240	272	5	1	1	0
0.2	30	1	902	1921	933	10	1	0	0.02
	30	2	91	192	123	10	1	1	0
	30	3	91	282	272	10	1	1	0.03
0.3	30	1	902	2363	936	17	4	0	0.23
	30	2	91	238	127	17	4	2	0.08
	30	3	91	328	276	17	4	2	0.14
0.1	50	1	2502	4779	2554	5	2	0	0.03
	50	2	151	286	205	5	2	2	0
	50	3	151	436	454	5	2	2	0.02
0.2	50	1	2502	6765	2553	33	1	0	1.73
	50	2	151	408	205	33	1	3	0.08
	50	3	151	558	454	33	1	3	0.08
0.3	50	1	2502	8883	2560	44	8	0	>3600
	50	2	151	536	314	44	34	79	70.03
	50	3	151	686	563	44	34	79	101
0.1	70	1	4902	21485	4972	N.F.	0	0	>3600
	70	2	211	928	688	69	18	122	>3600
	70	3	211	1138	1034	69	16	107	>3600
0.2	70	1	4902	21485	4972	N.F.	0	0	>3600
	70	2	211	928	688	69	28	346	>3600
	70	3	211	1138	1034	69	22	229	>3600
0.3	70	1	4902	21485	4972	N.F.	0	0	>3600
	70	2	211	928	688	69	51	356	>3600
	70	3	211	1138	1034	69	58	346	>3600
0.1	100	1	10002	26337	10106	47	4	0	764.47
	100	2	301	792	412	47	4	7	2.03
	100	3	301	1092	911	47	4	7	3.23
0.2	100	1	10002	44619	10102	N.F.	0	0	>3600
	100	2	301	1346	1087	97	53	633	>3600
	100	3	301	1646	1515	98	47	568	>3600
0.3	100	1	10002	58545	10102	N.F.	0	0	>3600
	100	2	301	1768	1567	99	35	1131	>3600
	100	3	301	2068	2354	99	61	1393	>3600

Table 5: Experimental results for LPnPP, LPcPP, and LPCPP models for problems of **type 2** (injected Hamiltonian path(s)) with *2 colors* on the edges.

Graph Density	Cross Peaks	Model	Nodes	Arcs (var.)	N. of Constr.	Path Length	N. of Paths	N. of Cycles	Total Time (s)
0.1	20	1	402	783	423	19	1	0	0
	20	2	41	78	62	19	1	0	0
	20	3	41	118	141	19	1	0	0
0.2	20	1	402	895	423	19	1	0	0.03
	20	2	41	90	62	19	1	0	0.02
	20	3	41	130	141	19	1	0	0.03
0.3	20	1	402	1035	423	19	1	0	0.03
	20	2	41	104	62	19	1	0	0.02
	20	3	41	144	141	19	1	0	0.02
0.1	30	1	902	1923	933	29	1	0	0.02
	30	2	61	128	92	29	1	0	0.02
	30	3	61	188	211	29	1	0	0.02
0.2	30	1	902	2357	933	29	1	0	0.51
	30	2	61	158	92	29	1	0	0.05
	30	3	61	218	211	29	1	0	0.03
0.3	30	1	902	2767	934	29	2	0	0.69
	30	2	61	186	97	29	2	4	0.2
	30	3	61	246	216	29	2	4	0.19
0.1	50	1	2502	5737	2553	49	1	0	2.03
	50	2	101	230	152	49	1	0	0.05
	50	3	101	330	351	49	1	0	0.08
0.2	50	1	2502	8089	2558	49	6	0	5.16
	50	2	101	326	161	49	6	4	0.53
	50	3	101	426	351	0	1	4	1.2
0.3	50	1	2502	8089	2558	49	22	0	328
	50	2	101	326	161	49	22	436	97
	50	3	101	426	351	0	22	436	124
0.1	70	1	4902	19937	4972	0	0	0	>3600
	70	2	141	574	211	69	124	768	1425
	70	3	141	714	490	69	124	894	1829
0.2	70	1	4902	19937	4972	0			>3600
	70	2	141	574	211	69	654	2276	>3600
	70	3	141	714	490	69	559	2009	>3600
0.3	70	1	4902	26783	4972	0			>3600
	70	2	141	772	211	69	257	2820	2206
	70	3	141	912	490	69	257	2975	2341
0.1	100	1	10002	33279	10122	99	20	0	454.98
	100	2	201	668	341	99	20	20	6.54
	100	3	201	868	741	99	20	21	12.9
0.2	100	1	10002	51865	10102	N.F.	0	0	>3600
	100	2	201	1044	3412	99	151	2960	>3600
	100	3	201	1244	3755	99	162	2893	>3600
0.3	100	1	10002	66155	10102	N.F.	0	0	>3600
	100	2	201	1332	3983	99	143	3539	>3600
	100	3	201	1532	4358	99	123	3535	>3600

Table 6: Experimental results for LPnPP, LPcPP, and LPCPP models for problems of **type 1** (no injected Hamiltonian path) with *3 colors* on the edges.

Graph Density	Cross Peaks	Model	Nodes	Arcs (var.)	N. of Constr.	Path Length	N. of Paths	N. of Cycles	Total Time (s)
	20	1	402	663	423	19	1	0	0
0.1	20	2	61	98	82	19	1	0	0
	20	3	61	158	181	19	1	0	0
	20	1	402	763	423	19	1	0	0
0.2	20	2	61	114	82	19	1	0	0
	20	3	61	174	181	19	1	0	0.03
	20	1	402	921	423	19	1	0	0.03
0.3	20	2	61	138	82	19	1	0	0
	20	3	61	198	181	19	1	0	0.02
	30	1	902	1669	933	29	1	0	0
0.1	30	2	91	166	122	29	1	0	0.02
	30	3	91	256	271	29	1	0	0.03
	30	1	902	2117	933	29	1	0	0.05
0.2	30	2	91	212	122	29	1	0	0.03
	30	3	91	302	271	29	1	0	0.03
	30	1	902	2643	933	29	1	0	0.45
0.3	30	2	91	266	122	29	1	0	0.05
	30	3	91	356	271	29	1	0	0.06
	50	1	2502	5417	2553	49	1	0	0.3
0.1	50	2	151	326	202	49	1	0	0.03
	50	3	151	476	451	49	1	0	0.03
	50	1	2502	7321	2553	49	1	0	2.04
0.2	50	2	151	442	202	49	1	0	0.14
	50	3	151	592	451	49	1	0	0.17
	50	1	2502	7321	2553	49	16	0	822
0.3	50	2	151	442	202	49	16	87	352
	50	3	151	592	451	49	16	87	528
	70	1	4902	23141	4972	N.F.	1	0	34
0.1	70	2	211	1000	1330	69	1	422	8
	70	3	211	1210	2516	69	1	422	2
	70	1	4902	23141	4972	N.F.	0	0	>3600
0.2	70	2	211	1000	1330	69	48	642	1824
	70	3	211	1210	2516	69	23	328	>3600
	70	1	4902	23141	4972	N.F.	0	0	>3600
0.3	70	2	211	1000	1330	69	78	971	>3600
	70	3	211	1210	2516	69	166	1720	>3600
	100	1	10002	30693	10103	99	1	0	17.08
0.1	100	2	301	924	402	99	1	0	0.45
	100	3	301	1224	901	99	1	0	0.59
	100	1	10002	45081	10102	N.F.	0	0	>3600
0.2	100	2	301	1360	905	99	18	486	>3600
	100	3	301	1660	1660	99	33	727	>3600
	100	1	10002	60525	10102	N.F.	0	0	>3600
0.3	100	2	301	1828	1552	99	27	1124	>3600
	100	3	301	2128	2854	99	70	1884	>3600

Table 7: Experimental results for LPnPP, LPcPP, and LPCPP models for problems of **type 2** (injected Hamiltonian path(s)) with **3 colors** on the edges.