Exact approaches for the orderly colored longest path problem: performance comparison

Francesco Carrabs[a,*], Raffaele Cerulli[a], Giovanni Felici[b], Gaurav Singh[c]

[a] *Department of Mathematics, University of Salerno, Fisciano, Italy*
[b] *IASI-CNR, Rome, Italy*
[c] *Global Technology, BHP Billiton, Perth, Australia*

Abstract

In this paper we consider a particular graph-optimization problem. Given an edge-colored graph and a set of constraints on the sequence of the colors, one is to find the longest path whose colored edges obey the constraints on the sequence of the colors. In the actual formulation, the problem generalizes already known NP-Complete problems, and, evidently, the alternating path problem in edge colored graphs. Recent literature has shown several contexts where such problem may be useful to model interesting applications, and has proposed exact IP models and related algorithms. We extend on these existing models and extensively test new formulations for the problem, showing how one of the newly developed model clearly exhibits better performance, allowing to solve at optimality instances of significant sizes.

Keywords: Edge Colored Graphs, Longest Path, Integer Programming.

1. Introduction

This paper deals with the problem of finding longest paths in edge-colored graphs with constraints on the color sequence, a generalization of the problem of finding an alternating path in c-edge-colored graphs, which is, in turn, a

*Corresponding author
Email addresses: fcarrabs@unisa.it (Francesco Carrabs), raffaele@unisa.it (Raffaele Cerulli), giovanni.felici@iasi.cnr.it (Giovanni Felici), gaurav.singh@bhpbilliton.com (Gaurav Singh)

generalization of the NP-Complete Hamiltonian path problem in non-colored graphs [11].

Edge-colored graphs play an important role in modelling certain real life problems, several of them coming from genetic and molecular biology, such as determining the spatial order of chromosomes [8], constructing physical maps of DNA sequences (the so-called Double Digest Problem, [15]), and, more recently, the reconstruction of RNA molecule structure from the correlation signals obtained by Nuclear Magnetic Resonance (NMR). The latter model, originally studied in [2, 4] for the 2-dimensional case, has been extended to the more complex case of 3-dimensions in [18] and then in [17], where the problem has been formalized as Orderly Colored Longest Path Problem (OCLP) and different optimization models have been proposed and successfully tested, based on the search of the longest path on certain expanded graphs.

The use of colors to add another layer of information on graphs has been formalized long ago, in its two main variants - node colored graphs and edge colored graphs, and their power in modeling different types of problems has been extensively discussed (among others, in [6, 7, 10, 12, 13]; more specifically, the complexity of finding special paths in edge-colored graphs is discussed in [3]; such problem, as shown later, is deeply connected with the models proposed in this paper, while in [1] the variants generated by graphs with fixed degree is discussed. Additional results for edge-colored paths on multigraph is also discussed in [1]. An overview of the potential role of colored path problems in applications is given in [16], where, besides extending the capabilities of the models presented in [17], different potential applications are listed, ranging from path optimization on grids (city block models, memory arrays, circuit design), pick-up and delivery problems, communication on networks and secure transmission, to the knight's tour problem.

To be noted is also the similarity of OCLP with the Shortest Path Tour Problem (SPTP) [9], where the latter can be equivalently formulated using packing constraints in place of the covering ones of the former. Such difference, albeit it may seem small, induces relevant modelling and solution issues.

The main contribution of this paper is to extend the set of optimization models already discussed in the literature for the OCLP problem, investigating alternative approaches and testing their performances on a large and diverse body of test problems, accurately generated to evaluate the performances of the models in different scenarios. The optimization problem indeed can be tackled from different angles and its solution may leverage, among others, on background from network flows, job scheduling, polyhedral methods. We can then conclude that one of the proposed models shows dominating performances and can thus be considered, at present, as the reference for solving OCLPs.

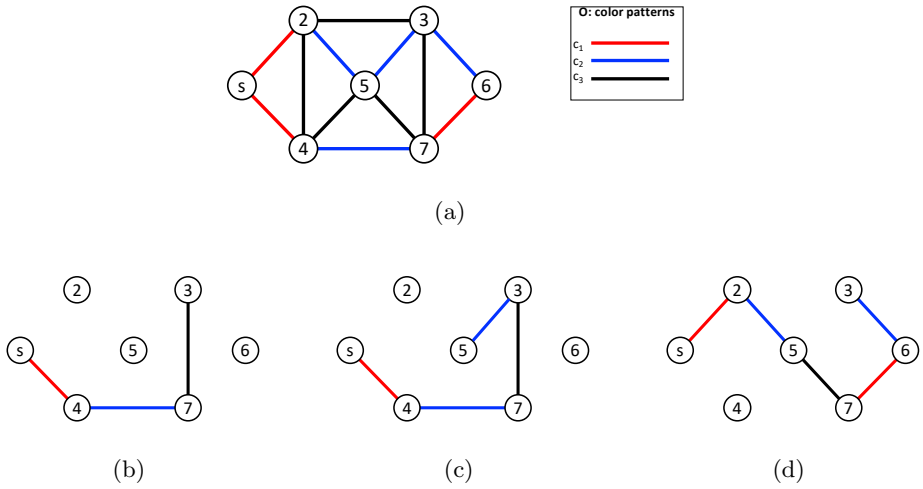
The paper is organized as follows.

Section 2 provides a formal definition of the OCLP problem that generalizes the definitions given in the previous literature. In this extension, each edge may have more than one color. Such simple modification allows to significantly extend the range of application of this model at a contained computational cost - as it will be shown later.

In Sections 3, 4, and 5 three alternative formulations and the related solution algorithms are presented. The first, referred to as *Expanded Flow Formulation*, is the most flexible and performing one among those discussed in [16, 17]; the second one, *the Scheduling Based Formulation*, is inspired to well-established approaches in the scheduling literature; the third, referred to as *Compact Flow Formulation*, has been specifically designed for OCLP, and is the one that presents the more interesting performances, as reported in Section 6. All three formulations pay specific attention to the efficiency of the model and to the opportunity offered by state-of-the-art MIP solvers to exploit polyhedral approaches with valid cuts or lazy constraints.

Once again, despite the availability of extraordinarily powerful and mature softwares, the role of a well designed - or possibly lucky - formulation is not negligible in the solution of large and complex optimization problems, reminding us how mathematical optimization modelling may still be an art.

General conclusions are drawn in Section 7.



(a) (b) (c) (d)

Figure 1: (a) A undirected and colored graph G and its color pattern. (b) An orderly colored path. (c) An infeasible path. (d) A longest orderly colored path of G.

2. Definitions and Notation

Let $G(V, E, C)$ be an undirected and colored graph, where $V = \{v_1, \ldots, v_n\}$ is the set of nodes, $E = \{e_1, \ldots, e_m\}$ is the set of edges and C is the set of colors associated to the edges of G, with $|C| \geq 2$. The *color pattern* $O \leq c_1, c_2, ..., c_k >$ is a fixed sequence of the C colors. In Figure 1(a) is shown an undirected and colored graph G and its color pattern $O =< c_1, c_2, c_3 >$. A *simple path* in G is a path that does not contain repeated vertices. An *orderly colored path* in G is a simple path in which the colors of its consecutive edges coincide with the color pattern O. In Figure 1(b) is depicted an *orderly colored path* in G while the path in Figure 1(c) does not respect the color pattern. The *Orderly Colored Longest Path* (OCLP) problem consists of finding in G the longest orderly colored path starting from a source node s. In Figure 1(d) is shown the optimal solution for the OCLP in G.

In this paper we face a more general version of the OCLP problem in which neither the starting node nor the starting color are fixed a priori. In addition, we will allow a color to have more predecessors in the color pattern. According to this more general version of the problem, the path $p =< 5, 7, 6, 3, 2 >$ is an

orderly colored path of G (Figure 1a). The new version of the problem is harder to solve with respect to the original one but it is more suitable for real world applications.

Let $\rho : O \to O^+$ and $\eta : O \to O^+$ be two functions that, given an input color c, return the set of colors that immediately precede and succeed c in O, respectively. For instance, according to the color pattern in Figure 1, we have that $\rho(blue) = red$ and $\eta(blue) = black$. Similarly, let us define the function $\ell : E^+ \to O^+$ that, given an input set of edges, returns the set of colors associated to these edges.

The set of nodes adjacent to a node u, through an edge of color c, is denoted by $V_c(u)$. Formally $V_c(u) = \{v \in V : (u, v) \in E, \ell(u, v) = c\}$. Let $V_c = \{u \in V : V_c(u) \neq \emptyset\}$ be the set of nodes having at least one outgoing edges of color c. Finally, given a set of nodes $S \subseteq V$ we define $E(S) = \{(u, v) \in E : u \in S, v \in S\}$ and $E_c(S) = \{(u, v) \in E : u \in S, v \in S, \ell(u, v) = c\}$.

Since we use directed graphs in some formulations, we introduce some notations even for this type of graphs. Given a directed colored graph and a node u of this graph, we denote by $\delta^+(u) = \{(u, v) \in E\}$ and $\delta^-(u) = \{(w, u) \in E\}$ the set of edges outgoing from and ingoing in u, respectively. Moreover, let $\delta^+_c(u) = \{(u, v) \in \delta^+(u) : \ell(u, v) = c\}$ and $\delta^-_c(u) = \{(w, u) \in \delta^-(i) : \ell(w, u) = c\}$ be the subset of edges, having color c, outgoing from and ingoing in u.

3. Expanded Flow Formulation

Here we briefly describe the rationale behind a formulation for OCLP already presented in [16, 17]. For more details, interested readers may refer to the mentioned references, where the original version of the formulation is referred to as `LPcCP`, Longest path in the cyclic c-connected graph problem. We recall that such formulation has proven to be the most effective among those based on flows in previous studies and is then here taken as reference.

Figure 2 provides a visual glance of the formulation, by depicting the transformation that undergo the simple OCLP problem represented in Figure 1(a).

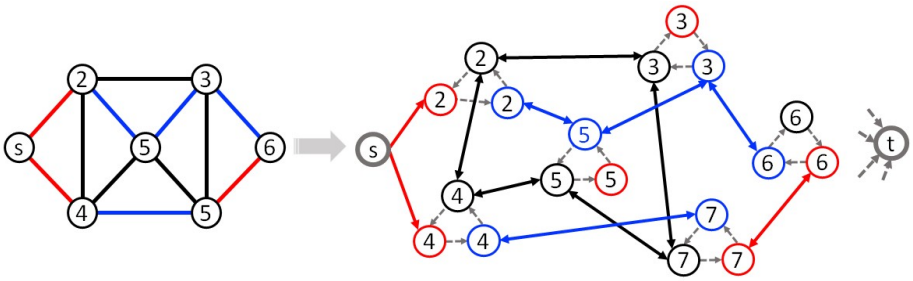


Figure 2: Transformation of the OCLP in Figure 1(a). Each vertex is duplicated in a colored copy; arcs between the copy of the same node represent allowed color transitions. Each colored arc connects the copy of the same color of its head and tail.

We assume that the original colored graph is undirected, but the transformation would be equivalent for directed graphs. Each original vertex is duplicated into as many colored copies as colors; among the colored copy of the same vertex, the allowed color transitions are expressed by directed arcs (dashed in grey in Figure 2). Then, the original colored edges (here, undirected) are represented in the expanded graph by a pair of direct colored arcs that connects the copy of the same color of its head and tail. For better readability, we represent the arcs that connect the same two nodes with different sense with a single edge with an arrow symbol in both extremes. E.g., the blue edge $(2, 5)$ is translated into the two blue arcs that connect the blue copy of node 2 with the blue copy of node 5, and vice versa.

The described scheme differs for the source vertex, when present, that does not get duplicated by colors, but is connected to the properly colored copy of the nodes connected by the arcs in its star; and for the insertion of a sink node t, that is connected to all the nodes in the graph with incoming arcs to properly reproduce the flow mode (such arcs are not reported in the figure to improve its readability).

The basic idea is that when the flow leaving s and bound to t enters a set of colored nodes associated with an original vertex, it can use only one arc in that set and then has to exit. It is easy to see that such a flow would obey any set of color transition rules that do not form cycles among the c colors. Such

trick is not sufficient, though, to guarantee a correct formulation for OCLP: the complete graph can contain cycles and the search for the longest path may get trapped into augmenting ones. For this reason, the formulation needs to be completed with cycle elimination constraints, typically in exponential number.

The full optimization model, resulting in a maximum flow formulation with additional packing and cycle elimination constraints, is named *Expanded Flow Formulation (EFF)*, and is given below. As customary, we indicate with $\delta^+(v)$ (resp. $\delta^-(v)$) the set of arcs entering (resp. exiting) node v, and $\delta(v) = \delta^+(v) \cup \delta^-(v)$.

- original graph $G_o = (V_o, E_o)$, with $|V_o| = n$;
- set $V = \bigcup_{i=1}^{n} V_i$, where V_i contains a set of colored nodes associated to vertex i in V_o;
- source node s, sink node t;
- set $E = \delta(s) \cup \delta(t) \cup E^* \cup E^1 \cup ... \cup E^n$, where E^* is composed of the arcs replicated from the edges of the original graph, and E^i is composed of the arcs connecting the nodes in V_i, $i \in \{1, ...n\}$;
- graph G = (V, E);
- Γ, the set of cycles of G;
- x_e, binary variable equal to 1 if arc $e \in E$ is in the optimal path, and 0 otherwise;
- p_e, value of arc $e \in E$ (typically, $p_e = 1, e \in E^*$, and 0 otherwise).

$$(\textbf{EFF}) \qquad \max \sum_{e \in E} p_e x_e \tag{1}$$

subject to:

$$\sum_{e \in \delta^+(v)} x_e - \sum_{e \in \delta^-(v)} x_e = 0 \qquad \forall v \in V \tag{2}$$

$$\sum_{e \in \delta^+(s)} x_e = 1 \tag{3}$$

$$\sum_{e \in \delta^-(t)} x_e = 1 \tag{4}$$

$$\sum_{e \in E^v} x_e \leq 1 \qquad \forall v \in V \tag{5}$$

$$\sum_{e \in \delta(v) \cap E^*} x_e \leq 1 \qquad \forall v \in V \tag{6}$$

$$\sum_{e \in C} x_e \leq |C| - 1 \qquad \forall C \in \Gamma \tag{7}$$

$$x_e \in \{0, 1\}, \qquad \forall e \in E \tag{8}$$

Above, Constraint (2) imposes the balance of the flow in all nodes of G, saved for s and t whose role of source and sink is established by Constraint (3) and (4); Constraint (5) requires that at most one arc is used within each set E^i, so to guarantee that the color sequence constraints are respected; then we require, with Constraint (6), that once the flow reaches any copy of the colored node in set V_i, it does not exit from the very same node to another set V_i, but rather proceeds to only another color-compatible copy within V_i. Finally, Constraint (7) represents the elimination of the cycles from the solution. As said, the enumeration of the cycles in Γ is unpractical. As explained in [17], such constraints are relaxed and a linear cycle separation oracle is used to separate cycles as they appear in the optimal solution of the relaxed problems, in a Branch & Cut scheme.

We finally note that, for the sake of simplicity, the formulation above has been described for the particular case of a problem where the source node is given, i.e., the flow must start from a given node. Nevertheless, the more general setting can be obtained with the insertion of a fictitious source node connected with outgoing arcs to all nodes in the graph, symmetrically to what is done for the sink node in Figure 2.

4. Scheduling Based Formulation

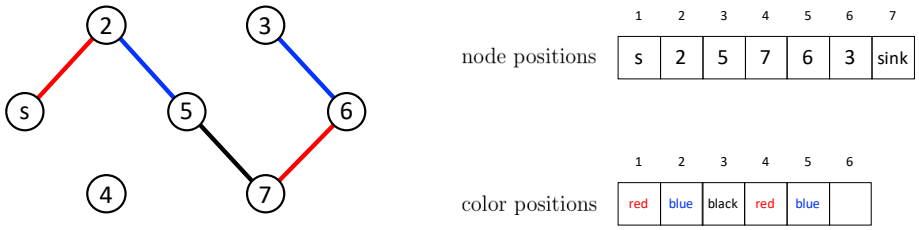
In this section, we introduce a formulation for the OCLP motivated by scheduling problems referred, in the following, as Scheduling Based Formulation (SBF). As in SBF the length of a longest simple path in G is equals to $|V|$ then $|V|$ is also the number of positions to be occupied in the path. The idea is to assign as many nodes of V as possible to the positions of this longest path by respecting the color pattern. If not all the nodes can be assigned, then we complete the construction of the path by using *sink* nodes that are different from the set of *proper* nodes V. In this way our model always returns a path with length $|V|$ but the proper nodes in it give us the real solution. For instance, the graph in Figure 1(a) contains 7 nodes and the optimal orderly colored path of this graph is $< s, 2, 5, 7, 6, 3 >$ (Figure 3(a)). Our model builds such an optimal solution by assigning to the nodes $< s, 2, 5, 7, 6, 3 >$ the positions $1, 2, 3, 4, 5, 6$, respectively, and it completes the path by assigning a sink node to the position 7 (Figure 3(b)). Moreover, it assigns the positions 1 and 4 to the color red, the positions 2 and 5 to the color blue and the position 3 to the color black while no color is assigned to position 6.

The decision variables are the following:

- $x_{u,k}$ binary variable equal to 1 if node $u \in V$ is scheduled at position k, and 0 otherwise;
- y_k binary variable equal to 1 if a sink node is scheduled at position k, and 0 otherwise;
- $z_{c,k}$ binary variable equal to 1 if color c is scheduled at position k, and 0 otherwise; Note that $z_{c,|V|} = 0\,, \forall c \in C$.

A complete mathematical programming formulation is presented below:

$$(\mathbf{SBF}) \qquad \min \sum_{k=1}^{|V|} y_k \tag{9}$$



(a) (b)

Figure 3: (a) The optimal orderly colored path of graph depicted in Figure 1(a). (b) The positions assigned by SBF to the nodes and to the colors according to this orderly colored path.

subject to:

$$\sum_{k=1}^{|V|} x_{u,k} \leq 1 \qquad u \in V \tag{10}$$

$$y_k + \sum_{u \in V} x_{u,k} = 1 \qquad k = 1, \ldots, |V| \tag{11}$$

$$y_k - y_{k+1} \leq 0 \qquad k = 1, \ldots, |V| - 1 \tag{12}$$

$$x_{u,k} + z_{c,k} - \sum_{v \in V_c(u)} x_{v,k+1} \leq 1 \qquad u \in V, c \in C, k = 1, \ldots, |V| - 1 \tag{13}$$

$$z_{c,k} + y_k + \sum_{u \in V \setminus V_c} x_{u,k} \leq 1 \qquad c \in C, k = 1, \ldots, |V| \tag{14}$$

$$y_{k+1} + \sum_{c \in C} z_{c,k} = 1 \qquad k = 1, \ldots, |V| - 1 \tag{15}$$

$$z_{c,k+1} - \sum_{c' \in \rho(c)} z_{c',k} \leq 0 \qquad c \in C, k = 1, \ldots, |V| - 2 \tag{16}$$

$$x_{u,k} \in \{0, 1\} \qquad u \in V, k = 1, \ldots, |V| \tag{17}$$

$$y_k \in \{0, 1\} \qquad k = 1, \ldots, |V| \tag{18}$$

$$z_{c,k} \in \{0, 1\} \qquad c \in C, k = 1, \ldots, |V| - 1 \tag{19}$$

The objective function (9) minimizes the number of sink nodes inside the orderly colored path that is equivalent to maximizing the number of proper

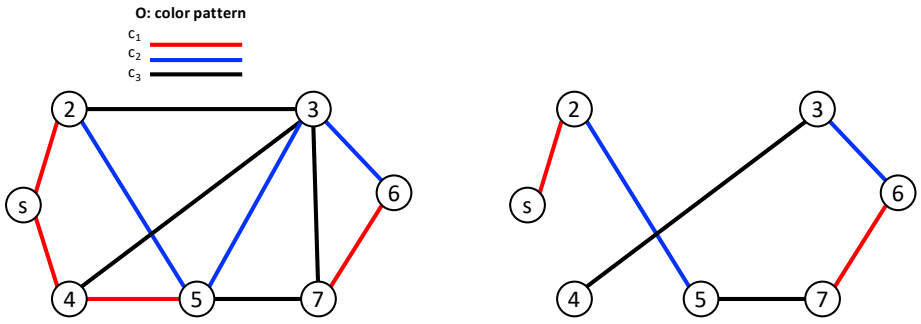
nodes into the path. Constraint (10) makes sure that every node u is allocated to at most one position. Constraint (11) ensures that at every position is assigned either a sink node or a proper node. Constraint (12) imposes that if a sink node is assigned to a position k then in every follow-up position $k' > k$ only sink nodes can be assigned. Constraint (13) states that if a node u is assigned position k and a color c is assigned the same position, then in position $k+1$ there must be a node v immediately adjacent to u through an edge (u, v) of color c. As a special case, when $|V_c(u)| = 0$, Constraint (13) also ensures that if u is assigned to position k, then color c cannot be assigned to the same position. Constraint (14) states that at every position k we can at most assign a color c or a sink node or a node u that does not have an edge with color c. Together with Constraint (11), Constraint (14) also ensures that if a color c is assigned at a position k then a node u, with $V_c(u) \neq \emptyset$, must be assigned at position k. Constraint (15) imposes that at every position k either a color is assigned or the next node must be a sink node. Finally, Constraint (16) ensures that path respects the required color pattern.

4.1. Observations and tightening constraints

In this section, we introduce new constraints for the SBF by lifting some of the original constraints.

- If for a particular color c, two nodes u and v' are connected to same set, $V_c(u)$, of adjacent nodes then since only one node can take a particular position, the Constraint (13) can be lifted by adding $x_{v',k}$ in the lhs. This can be generalized by adding in the lhs of constraint (13) the variables $x_{v',k}$ associated to all the nodes $v' \in V$ such that $V_c(v') = V_c(u)$. As an example, consider the graph in Figure 4(a). Here, both nodes 2 and 4 have same node 3 connected by the color *black*. Hence, if we assign position k to either node 2 or 4 and color 'black' at that position then at position $k+1$ we must assign node 3.

  The final enhanced version of the constraints is the following:



(a) (b)

Figure 4: (a) An example undirected and colored graph G and its color pattern. (b) An optimal orderly colored path of G.

$$x_{u,k} + \sum_{\substack{v' \in V \\ V_c(v') = V_c(u)}} x_{v',k} + z_{c,k} - \sum_{v \in V_c(u)} x_{v,k+1} \leq 1 \qquad u \in V, c \in C, k = 1, \ldots, |V| - 1 \tag{20}$$

- Let us consider a node v such that $V_c(v) \neq \emptyset$ and $v \notin \bigcup_{c' \in \rho(c)} V_{c'}$. In other words, v is a node with at least one edge with color c but v has no edges that have a color that is a predecessor of c in color patterns. Let us denote by Δ_c the set of nodes that satisfy this condition, under which it is possible to add the nodes $v \in \Delta_c$ to the *lhs* of Constraint (14) thereby tightening it. As it is always possible to have a node in position 1 that may not have an edge with color in $\rho(c)$, this enhancement is only valid for $k > 1$. Combining this with Constraint (11), the enhanced constraint essentially says that if a color c is assigned a position $k > 1$ then to create a feasible path at position k we must assign a node that has an edge of color c and also an edge with color in $\rho(c)$. As an example, consider again the graph in Figure 4(a). Here for color *black*, nodes 4 and 7 have an edge with this color but no edge with preceding color *blue*. Hence, if we don't assign *sink* node and assign color *black* at any position $k > 1$ then node 6 cannot be assigned position k (as it has no edge with color *black*)

and nodes 4 and 7 may also not be assigned this position, as they do not have an edge of preceding color *blue*. In other words, if nodes 4, 6 or 7 are assigned position $k > 1$ then color *black* cannot be assigned the same position.

Formally, the constraint can be written as follows:

$$z_{c,k} + y_k + \sum_{u \in V \setminus V_c} x_{u,k} + \sum_{v \in \Delta_c} x_{v,k} \leq 1 \qquad c \in C, k = 2, \ldots, |V| \quad (21)$$

- Let v be a node having an edge with color c and no edges that have a color from $\eta(c)$. This implies that if we assign to the position k the color c and to the position $k+1$ the node v then after v there must be a sink node. Following the example from the graph in Figure 4(a), node 3 has no edge with color *red*, which is the successor to color *black*. Hence, if node 2 is assigned position k with color *black* at position k. Then, either node 3 or *sink* node is assigned position $k+1$. In either case, at position $k+2$ we must assign a *sink* node as node 3 has no edge with color *red*.

  If Γ_c is the set of nodes that satisfy the previous conditions, the following new constraint holds:

$$z_{c,k} + \sum_{v \in \Gamma_c} x_{v,k+1} \leq 1 + y_{k+2} \qquad c \in C, k = 1, \ldots, |V| - 2 \quad (22)$$

  Above constraint can be further lifted as follows:

$$z_{c,k} + \sum_{v \in \Gamma_c} x_{v,k+1} + 2y_{k+1} \leq 1 + y_{k+2} \qquad c \in C, k = 1, \ldots, |V| - 2 \quad (23)$$

- As x variables are integers due to equality constraint (11), the integrality restriction on the y variables can be relaxed.

- From the above point, since y can only take a value of 0 or 1, it can be also deduced that due to equality constraint (15), $\sum_{c \in C} z_{c,k}$ will be only 0 or 1. Let us suppose we relax the integrality constraints on the z variables and consider the case when $z_{c,k}$ is fractional for more than one color, say c_1 and c_2. As a node can be assigned only one position, Constraints (11) and (14) together force the assignment in position k of a node u that has edges with both colors c_1 and c_2. The Constraint (13) will then have to assign a node v that has edges with u of different color c_1 and c_2, which contradicts the fact that every pair of nodes is connected by a unique color. If only one of $z_{c,k}$ is fractional then since $\sum_{c \in C} z_{c,k}$ is only 0 or 1 it will automatically take only integer values of either 0 or 1. From this discussion it can be concluded that the integrality conditions on the z variables can also be relaxed.

  To our surprise, in early computational experiments, the performance of CPLEX was worst when y and z variables were made continuous. Even though SBF with continuous variables has much fewer integer variables to branch on, we feel that when these variables are left as integers, CPLEX or Gurobi can add some special cuts which can assist in improving the performance. So, in our final experiments, we left these variables as integers.

5. Compact Flow Formulation

In this section we introduce a formulation for the OCLP that works on a new graph G' derived from the original graph G. More in detail, starting from the graph G, we build a new directed graph $G' = (V', E', C')$. V' contains all vertices of V, as well as a dummy source s. E' contains all the edges of E and the arcs $\{(s, v) : v \in V\}$; these additional arcs have a new color $\hat{c}$ such that $\hat{c} \in \rho(c)\ \forall c \in C$. Finally, $C' = C \cup \{\hat{c}\}$. Our model will look for the longest orderly colored path of G' starting from s. The introduction of a dummy source node allows us to build orderly colored paths of G that start from any node

and any color. We refer to this formulation as the Compact Flow Formulation (CFF).

We use the binary variables y_{uv} equal to 1 if arc (u, v) belongs to the solution and 0 otherwise. The mathematical model is the following:

$$(\mathbf{CFF}) \qquad \max \sum_{(u,v)\in E'} y_{uv} \tag{24}$$

subject to:

$$\sum_{(w,u)\in\delta^-(u)} y_{wu} \leq 1 \qquad u \in V \tag{25}$$

$$\sum_{(0,v)\in\delta^+_{c_1}(s)} y_{sv} = 1 \tag{26}$$

$$\sum_{(u,v)\in\delta^+_c(u)} y_{uv} \leq \sum_{(w,u)\in\delta^-_{\rho(c)}(u)} y_{wu} \qquad u \in V, c \in \ell(\delta^+(u)) \tag{27}$$

$$\sum_{(u,v)\in E(S)} y_{uv} \leq |S| - 1 \qquad S \subseteq V', |S| \geq 3 \tag{28}$$

The objective function (24) maximizes the number of arcs selected. Constraint (25) ensures that each node can have at most one ingoing arc while Constraint (26) imposes that one edge outgoing from s must be selected. Constraint (27) states that any arc outgoing from u and with color c can be selected only if at least one arc ingoing in u with a color in $\rho(c)$ is selected. Moreover, these constraints, together with the Constraint (25), ensure that there can be at most one arc outgoing from any node. Finally, subtours are eliminated through Constraint (28).

5.1. Valid Inequalities

In this section, we introduce two types of valid inequalities for the OCLP that speed up the convergence of the model. The first set of inequalities is derived from the following remarks.

Remark 1. A trivial upper bound of the length of any simple path in G is $|V| - 1$ (Hamiltonian path).

Remark 2. In any orderly colored path of G the number of arcs with the same color is upper bounded by $\left\lceil \frac{|V|-1}{|O|} \right\rceil$.

PROOF. From Remark 1 the maximum number of edges composing any orderly colored path p of G is $|V|-1$. Moreover, in order to respect the color pattern O, if the color $c \in O$ occurs in the position i of p, its next occurrence in p will be in the position $i + |O|$ and so on. Therefore, the maximum number of occurrences of any color $c \in O$ is $\left\lceil \frac{|V|-1}{|O|} \right\rceil$. ■

From Remarks 2, we derive the following set of valid inequalities:

$$\sum_{(u,v) \in E_c(V)} y_{uv} \leq \left\lceil \frac{|V|-1}{|O|} \right\rceil \qquad c \in C \tag{29}$$

Since the number of these inequalities is equal to $|C|$, no separation procedures are applied but they are directly introduced into the CFF model as a priori cuts.

Remark 3. Since $|O| \geq 2$ then two adjacent arcs have always different color in any orderly colored path.

From Remark 3 we deduce that given a subset $S \subseteq V$, the maximum number of arcs with the same color in this subset is $\left\lfloor \frac{|S|}{2} \right\rfloor$. Therefore, the following are valid inequalities:

$$\sum_{(i,j) \in E_c(S)} y_{ij} \leq \left\lfloor \frac{|S|}{2} \right\rfloor \qquad S \subseteq V, |S| \geq 3, c \in O \tag{30}$$

In Figure 5 is shown (a part of) the relaxed solution of CFF model computed by CPLEX on the root node. All the edges in this figure have the same color and the number on each arc represents the value of its variable. Now, let us

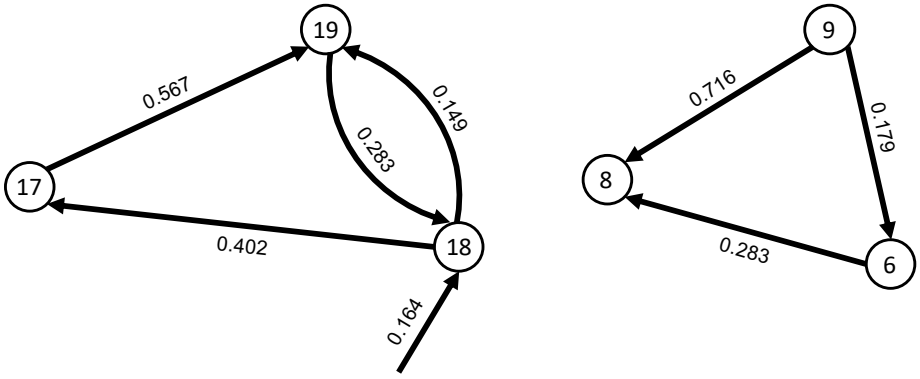


Figure 5: A part of a relaxed solution on the root node on which the valid inequalities (30) are violated.

consider the triangle composed of nodes 17, 18 and 19. Notice that the subtour elimination constraints are not invoked on this triangle because the sum of variable values is around 1.40 and then lower than 2. However, from Remark 3 we know that is possible to select at most one of arcs of this triangle and then, by introducing the inequalities $y_{17,18} + y_{18,17} + y_{17,19} + y_{19,17} + y_{18,19} + y_{19,18} \leq 1$, we cut away this solution. A similar reasoning holds for the other triangle composed of nodes 6, 8 and 9.

When $|S|$ is odd, the inequalities (30) are the well-known odd-set inequalities. These inequalities can be separated in exact way by using the algorithm proposed in [14]. However, since the separation procedure should be carried out for each color $c \in C$, it becomes too much expensive to use the exact separation. For this reason, we use a faster approach by separating the inequalities (30) through the Drop heuristic proposed in [5] for the Heaviest K-Subgraph problem.

6. Computational tests

In this section we present the results of the computational tests we made to evaluate the performance of the three models. All the mathematical formulations were coded in C++ and solved using the IBM ILOG CPLEX 12.6.1 solver.

All tests were performed in the single thread mode on a machine with an Intel Xeon E5 processor running at 2.3 GHz and 128 GB of RAM.

The computational tests are carried out on instances with a number of nodes ranging from 25 to 100, a density ranging from 0.2 to 0.4 and a number of colors equal to 3, 6, 9 and 12. For each combination of parameters, five different instances were generated that together represent a *scenario*. Thus, in total we had 240 individual instances, grouped in 48 scenarios, which form the *Set1* dataset. The name of the instance shows its characteristics and has the following format: *n_c_density_trans_seed*. The transition parameter (*trans*) can assume two values: 0 or 1. When $trans = 0$ the color pattern states that each color can be followed by a single other color. All the instances of $Set1$ have this parameter equal to zero. In addition, we generated a second set of instances, named *Set2*, in which the parameter *trans* is equal to 1. This means that a color $c \in C$ can be preceded by one or more colors in an orderly colored path. Due to this characteristic, the instances of $Set2$ have more feasible solutions to evaluate and then they result harder to solve. To allow the creation of different transitions among the colors, the instances of $Set2$ have at least six colors. Therefore, this dataset has 180 individual instances grouped in 36 scenarios.

The first comparison of the three models is carried out on the $Set1$. Table 1 reports the results of this comparison. We fixed a maximum running time equal to 1 hour for each instance. When a certified optimal solution is not found within this threshold, the best solution found is returned and used for the comparisons. Moreover, the solution values that are not proven optimal are marked with a "*" symbol in the table.

The table is organized as follows. Each line in the table represents a scenario composed of five instances with the same characteristics, and the results shown in each line are the average values of these five instances. The first column shows the name of the scenario while the next three columns report the characteristics of the instances: the number of nodes (n), the number of edges (m) and the number of colors (c). Finally, the last six columns report the solution value (Obj) and the computational time ($Time$), in seconds, of CFF, EFF and SBF

				CFF		EFF		SBF	
Scenario	**n**	**m**	**c**	**Obj**	**Time**	**Obj**	**Time**	**Obj**	**Time**
25_3_20_0_1	25	57	3	**15.0**	**0.02**	**15.0**	**0.23**	**15.0**	**4.98**
25_3_30_0_21	25	80	3	**22.0**	**0.09**	**22.0**	**1.38**	**22.0**	**9.63**
25_3_40_0_41	25	103	3	**23.0**	**0.64**	**23.0**	**8.25**	**23.0**	**141.79**
25_6_20_0_61	25	58	6	**5.0**	**0.02**	**5.0**	**0.28**	**5.0**	**0.65**
25_6_30_0_81	25	85	6	**9.0**	**0.05**	**9.0**	**0.63**	**9.0**	**0.26**
25_6_40_0_101	25	102	6	**13.0**	**0.43**	**13.0**	**1.34**	**13.0**	**12.39**
25_9_20_0_181	25	57	9	**5.0**	**0.01**	**5.0**	**0.37**	**5.0**	**0.14**
25_9_30_0_201	25	81	9	**7.0**	**0.06**	**7.0**	**1.15**	**7.0**	**0.88**
25_9_40_0_221	25	99	9	**7.0**	**0.08**	**7.0**	**1.57**	**7.0**	**0.79**
25_12_20_0_301	25	59	12	**6.0**	**0.02**	**6.0**	**0.65**	**6.0**	**0.09**
25_12_30_0_321	25	81	12	**9.0**	**0.03**	**9.0**	**1.77**	**9.0**	**0.13**
25_12_40_0_341	25	100	12	**9.0**	**0.18**	**9.0**	**2.09**	**9.0**	**0.62**
50_3_20_0_421	50	215	3	**48.0**	**11.08**	**48.0**	**292.80**	46.2*	3612.14
50_3_30_0_441	50	325	3	**49.0**	**1.21**	**49.0**	**42.05**	**49.0**	**146.28**
50_3_40_0_461	50	443	3	**49.0**	**1.07**	**49.0**	**2.34**	**49.0**	**15.07**
50_6_20_0_481	50	216	6	**27.0**	**2.69**	**27.0**	**402.69**	**27.0**	**699.80**
50_6_30_0_501	50	335	6	**46.0**	**252.56**	41.0*	3601.94	40.6*	3612.18
50_6_40_0_521	50	439	6	**49.0**	**382.49**	46.0*	3601.99	47.2*	3612.17
50_9_20_0_601	50	219	9	**14.0**	**1.21**	**14.0**	**259.94**	**14.0**	**13.40**
50_9_30_0_621	50	337	9	**34.0**	**20.77**	**34.0***	**3385.21**	**34.0***	**3612.21**
50_9_40_0_641	50	431	9	**44.0**	**416.09**	37.0*	3602.03	42.4*	3612.21
50_12_20_0_721	50	221	12	**7.0**	**0.36**	**7.0**	**27.56**	**7.0**	**25.85**
50_12_30_0_741	50	339	12	**24.0**	**5.71**	**24.0**	**244.55**	**24.0**	**1113.15**
50_12_40_0_761	50	450	12	**37.0**	**39.43**	35.4*	3602.05	36.6*	3612.25
75_3_20_0_841	75	515	3	**74.0**	**6.24**	**74.0**	**89.35**	71.4*	3612.22
75_3_30_0_861	75	764	3	**74.0**	**7.17**	**74.0**	**29.40**	**74.0**	**435.76**
75_3_40_0_881	75	1006	3	**74.0**	**0.35**	**74.0**	**4.67**	**74.0**	**150.88**
75_6_20_0_901	75	506	6	**68.0***	**3610.43**	62.0*	3601.97	51.2*	3612.24
75_6_30_0_921	75	753	6	**74.0**	**36.76**	60.6*	3602.02	62.2*	3612.26
75_6_40_0_941	75	1004	6	**74.0**	**36.10**	71.2*	3601.93	70.8*	3612.27
75_9_20_0_1021	75	516	9	**47.0**	**51.93**	30.2*	3602.05	39.8*	3612.64
75_9_30_0_1041	75	757	9	**65.0***	**3610.66**	45.2*	3602.06	48.2*	3612.27
75_9_40_0_1061	75	1009	9	**69.0***	**3611.20**	53.4*	3601.97	58.0*	3612.33
75_12_20_0_1141	75	518	12	**27.0**	**12.42**	24.0*	3602.06	26.4*	3612.34
75_12_30_0_1161	75	748	12	**57.0**	**1854.74**	33.6*	3602.06	46.4*	3612.35
75_12_40_0_1181	75	994	12	**65.0***	**3611.25**	29.8*	3602.03	51.0*	3612.39
100_3_20_0_1261	100	943	3	**99.0**	**0.94**	**99.0**	**53.29**	94.6*	3612.30
100_3_30_0_1281	100	1400	3	**99.0**	**0.96**	**99.0**	**14.76**	98.2*	2366.49
100_3_40_0_1301	100	1819	3	**99.0**	**2.89**	**99.0**	**4.91**	**99.0**	**0.34**
100_6_20_0_1321	100	960	6	**95.0***	**3611.16**	71.8*	3602.01	68.6*	3612.39
100_6_30_0_1341	100	1394	6	**99.0**	**680.17**	83.2*	3602.03	86.2*	3612.40
100_6_40_0_1361	100	1826	6	**99.0**	**37.66**	92.4*	3602.01	92.8*	3612.48
100_9_20_0_1441	100	944	9	**81.0***	**3611.05**	63.6*	3602.04	46.6*	3612.43
100_9_30_0_1461	100	1398	9	**90.0***	**3611.42**	73.4*	3602.08	67.4*	3612.51
100_9_40_0_1481	100	1830	9	**97.0***	**3611.55**	59.2*	3602.06	82.0*	3612.66
100_12_20_0_1561	100	945	12	**64.0**	**709.38**	34.2*	3602.07	35.2*	3612.53
100_12_30_0_1581	100	1392	12	**79.4***	**3610.62**	45.8*	3602.09	56.0*	3612.71
100_12_40_0_1601	100	1817	12	**82.0***	**3611.50**	60.6*	3602.07	66.8*	3612.77

Table 1: Comparison on the *Set*1 instances.

models, respectively. For each row, the best Obj value and the corresponding computation time are in bold.

All the scenarios with 25 nodes are optimally solved by three models, within the time limit. The required computational time is lower than 1 seconds, 10 seconds and 150 seconds for CFF, EFF and SBF, respectively.

CFF is the only model able to optimally solve all the scenarios with 50 nodes and, often, it does so in less than one minute. Instead, on the same scenarios, EFF and SBF fail to find the optimal solution 5 and 6 times, respectively. These results show that CFF is more efficient than the other two models, but the performance gap will be more evident in the remaining scenarios. On the scenarios with 75 nodes, CFF did not find the optimal solution 4 times while EFF and SBF fail to find the optimal solution 9 and 10 times, respectively. It is worth noting that both EFF and SBF fail on all the scenarios with at least 6 colors and the failure of CFF occurs on the same scenarios too. From this observation, we derive that the complexity of the instances increases as the number of colors increase. Regarding the computational time, all the scenarios optimally solved by CFF have required less than a minute, except for the instance 75_12_30_0_1161. In general, CFF is much faster than the other two models.

On the largest scenarios with 100 nodes, the optimal solution is found six times by CFF, three times by EFF and just one time by SBF. Again, we observe that the difficulties in obtaining the optimal solution experienced by the models essentially occur in the scenarios with more than 3 colors. We compute the gap percentage between the best solution value and the solutions found by three models by using the formula: $100 \times \frac{best-Obj}{best}$. Since, on the $Set1$, the best solution is always found by CFF then the gap percentage for CFF is always equal to 0% while it is often greater than 10% for both EFF and SBF. In particular, ruling out the cases in which the best solution is found, the gap percentage ranges from 3.78% to 54% for EFF and ranges from 0.81% to 45% for SBF.

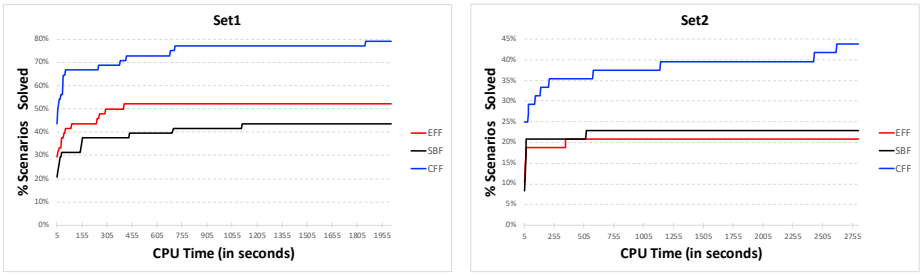
To further investigate how the colors affect the complexity of the scenarios, we defined new color patterns in which each color can be followed by more than

				CFF		EFF		SBF	
Scenario	**n**	**m**	**c**	**Obj**	**Time**	**Obj**	**Time**	**Obj**	**Time**
25_6_20_1_121	25	58	6	**7.0**	**0.03**	**7.0**	**0.82**	**7.0**	**6.50**
25_6_30_1_141	25	85	6	**16.0**	**0.52**	**16.0**	**8.17**	**16.0**	**12.31**
25_6_40_1_161	25	102	6	**17.0**	**0.64**	**17.0**	**15.38**	**17.0**	**19.94**
25_9_20_1_241	25	57	9	**11.0**	**0.02**	**11.0**	**2.15**	**11.0**	**3.89**
25_9_30_1_261	25	81	9	**8.0**	**0.04**	**8.0**	**3.06**	**8.0**	**2.65**
25_9_40_1_281	25	99	9	**11.0**	**0.32**	**11.0**	**10.15**	**11.0**	**11.71**
25_12_20_1_361	25	59	12	**5.0**	**0.02**	**5.0**	**0.91**	**5.0**	**0.42**
25_12_30_1_381	25	81	12	**10.0**	**0.03**	**10.0**	**2.34**	**10.0**	**8.47**
25_12_40_1_401	25	100	12	**11.0**	**0.63**	**11.0**	**6.48**	**11.0**	**1.55**
50_6_20_1_541	50	216	6	**40.0**	**31.85**	**40.0***	**3601.95**	33.4*	3612.19
50_6_30_1_561	50	335	6	**49.0**	**1143.41**	48.2*	3568.54	41.0*	3612.17
50_6_40_1_581	50	439	6	**49.0**	**0.51**	48.2*	3534.30	47.2*	3612.19
50_9_20_1_661	50	219	9	**20.0**	**4.32**	19.0*	3602.05	**20.0**	**517.73**
50_9_30_1_681	50	337	9	**43.0**	**578.07**	39.4*	3601.85	36.0*	3612.18
50_9_40_1_701	50	431	9	**44.8***	**3610.16**	40.0*	3601.48	38.0*	3612.22
50_12_20_1_781	50	221	12	**9.0**	**0.49**	**9.0**	**345.83**	**9.0**	**11.37**
50_12_30_1_801	50	339	12	**34.0**	**37.77**	28.0*	3602.03	**34.0***	**3612.23**
50_12_40_1_821	50	450	12	**42.0**	**2616.34**	37.0*	3602.02	38.6*	3612.20
75_6_20_1_961	75	506	6	**73.0***	**3611.28**	68.6*	3601.90	50.2*	3612.27
75_6_30_1_981	75	753	6	**74.0**	**2432.78**	72.6*	3575.86	61.6*	3612.27
75_6_40_1_1001	75	1004	6	**74.0**	**137.42**	73.8*	3449.69	72.8*	3612.28
75_9_20_1_1081	75	516	9	58.0*	3609.43	**58.2***	**3601.98**	40.8*	3612.31
75_9_30_1_1101	75	757	9	**64.0***	**3610.26**	62.6*	3602.05	51.4*	3612.32
75_9_40_1_1121	75	1009	9	**68.0***	**3611.21**	66.6*	3601.83	60.4*	3612.33
75_12_20_1_1201	75	518	12	**42.0**	**209.27**	37.4*	3602.06	29.0*	3612.33
75_12_30_1_1221	75	748	12	**59.0***	**3610.91**	42.0*	3602.07	37.8*	3612.40
75_12_40_1_1241	75	994	12	**68.0***	**3610.68**	46.4*	3602.07	48.0*	3612.40
100_6_20_1_1381	100	960	6	**98.0***	**3611.17**	89.8*	3601.93	71.6*	3612.38
100_6_30_1_1401	100	1394	6	**98.0***	**3611.87**	95.0*	3602.00	85.8*	3612.44
100_6_40_1_1421	100	1826	6	**99.0**	**90.75**	97.2*	3602.04	93.6*	3612.54
100_9_20_1_1501	100	944	9	**81.0***	**3611.15**	74.8*	3602.00	53.8*	3612.45
100_9_30_1_1521	100	1398	9	**93.0***	**3611.21**	83.4*	3602.03	72.2*	3612.50
100_9_40_1_1541	100	1830	9	**97.0***	**3611.43**	85.6*	3602.04	84.0*	3612.58
100_12_20_1_1621	100	945	12	**67.0***	**3610.25**	47.8*	3602.04	32.0*	3612.57
100_12_30_1_1641	100	1392	12	**86.0***	**3610.96**	71.2*	3602.03	55.0*	3612.67
100_12_40_1_1661	100	1817	12	**95.0***	**3611.28**	70.6*	3602.05	67.0*	3612.78

Table 2: Comparison on the $Set2$ instances.

one color. The instances of $Set2$ contain scenarios with this type of patterns. Table 2 reports the results of the three models on $Set2$.

All the scenarios with 25 nodes are optimally solved by the three models



(a) (b)

Figure 6: Performance comparison among CFF, EFF and SBF on the $Set1$ instances (a) and $Set2$ instances (b).

in less than 20 seconds. However, the situation changes when considering the scenarios with 50 nodes, where the performance of EFF and SBF are poorer than those of CFF.

Only three scenarios with 75 nodes are optimally solved by CFF, while no scenario is solved at optimality by EFF and SBF. By comparing these results with the ones of Table 1, it is evident that the instances of $Set2$ are harder to solve for all the models. To any extent, CFF seems to perform indeed better: on the scenarios with 12 colors, the gap value among CFF and EFF ranges from 11% to 32% while the gap value among CFF and SBF ranges from 29% to 35%. Finally, only one scenario with 100 nodes is optimally solved by CFF, while none by the other two models. The gap among the solution values is drastically increased on these instances. Indeed, on the scenarios with 12 colors, the gap value among CFF and EFF ranges from 17% to 28% and the gap value among CFF and SBF ranges from 29% to 52%.

Finally, in Figure 6 we represent the results from Table 1 and Table 2 in a way that better highlights the performance of the models. The horizontal axis represents the CPU time in seconds and the vertical axis represents the percentage of scenarios optimally solved within the time limit. This means that the faster the growth, the better the performance of the model. The blue curve is associated to CFF, the red curve to EFF and the black curve to SBF.

Average CFF Time - Set1			
Colors	Density		
	20	30	40
3	0.02	0.09	0.64
6	0.02	0.05	0.43
9	0.01	0.06	0.08
12	0.02	0.03	0.18

Average CFF Time - Set2			
Colors	Density		
	20	30	40
6	0.03	0.52	0.64
9	0.02	0.04	0.32
12	0.02	0.03	0.63

Average EFF Time - Set1			
Colors	Density		
	20	30	40
3	0.23	1.38	8.25
6	0.28	0.63	1.34
9	0.37	1.15	1.57
12	0.65	1.77	2.09

Average EFF Time - Set2			
Colors	Density		
	20	30	40
6	0.82	8.17	15.38
9	2.15	3.06	10.15
12	0.91	2.34	6.48

Average SBF Time - Set1			
Colors	Density		
	20	30	40
3	4.98	9.63	141.79
6	0.65	0.26	12.39
9	0.14	0.88	0.79
12	0.09	0.13	0.62

Average SBF Time - Set2			
Colors	Density		
	20	30	40
6	6.5	12.31	19.94
9	3.89	2.65	11.71
12	0.42	8.47	1.55

Figure 7: Average solution times with different density-color combinations for $Set1$ (left) and $Set2$ (right) instances. Average running times are limited to instances with 25 nodes.

75% of the scenarios are optimally solved by CFF in less than 800 seconds. Much lower is the percentage of solved scenarios of EFF and SBF. In particular, EFF optimally solves around 52% of scenarios in less than 500 seconds while SBF optimally solves around 43% in less than 1200 seconds. It is worth noting that, within 600 seconds, the percentage of scenarios optimally solved by CFF, EFF and SBF is 72.92%, 52.08% and 39.58%, respectively.

The results depicted in Figure 6(b) show that the effectiveness of the models drastically decreases on the $Set2$. One again, CFF is the best model but, on this set of instances, the percentage of scenarios optimally solved by it, within the time limit, decreases from approx. 80% to 45%. Even worse is the situation for the other two models with a percentage equal to 23% for SBF and to 21% to EFF.

We additionally analyze the interactions between the number of colors and the density of the matrix on the solution time. The analysis, limited to problems with 25 nodes - all easy solvable - and distinguished for problems in $Set1$ and problems in $Set2$, is summarized in Figure 7, where the average solution time is reported for the different combinations of the number of colors (on the rows)

and graph densities (on the columns). The 3 left (right) tables are associated with the results of the 3 algorithms on the $Set1$ ($Set2$) instances. The colorscale associated with the magnitude of the cells clearly indicated the marked effect of density on solution times, and moreover how, when density is high, a smaller number of colors results in longer solution times (this behaviour emerges particularly for the $Set2$ instances).

7. Conclusions

In this work we consider the Orderly Colored Longest Path Problem, a special case of the longest path on edge-colored graphs, recently considered to model several real life problems. We evolved the mathematical models already presented in the literature and compared the best optimization algorithm available with two new mathematical models, SBF and CFF, that provide improvements on the state of the art.

The performance of these models is enhanced by introducing a priori cuts, valid inequalities and strengthen versions of the basic constraints. The computational results are carried out on several scenarios and show that one of the three newly developed models outperforms the others both in terms of computational time and quality of the solutions, and could thus be considered the reference for the optimal solution of Orderly Colored Longest Path Problems.

With respect to future research directions, we intend to further improve the performance of CFF model by studying new valid inequalities that can improve the LP relaxation. Moreover, since we observed a drastic reduction of the performance of models on the Set2 instances, due to the presence of different transitions among the colors, a future direction of work is the identification of specific properties or reduction algorithms that can curb the computing time of our models in instances characterized by non-circular transitions among colors.

References

[1] A. Abouelaoualim, K.Ch. Das, W. Fernandez de la Vega, M. Karpinski, Y. Manoussakis, C.A. Martinhon, and R. Saad. Cycles and paths in edge-colored graphs with given degrees. *J. Graph Theory*, 64:63–86, 2010.

[2] R. W. Adamiak, J. Blazewicz, P. Formanowicz, Z. Gdaniec, M. Kasprzak, M. Popenda, and M. Szachniuk. An algorithm for an automatic NOE pathways analysis of 2D NMR spectra of RNA dueplexes. *Journal of Computational Biology*, 11:163–179, 2004.

[3] A. Benkouar, Y. Manoussakis, V. Paschos, and R. Saad. On the complexity of finding alternating hamiltonian and eulerian cycles in edge-coloured graphs. *Lecture Notes in Comput. Science*, 557:190–198, 1991.

[4] J. Blazewicz, M. Szachniuk, and A. Wojtowicz. RNA tertiary structure determination: NOE pathways construction by tabu search. *Bioinformatics*, 21(10):2356–2361, 2005.

[5] J. Brimberg, N. Mladenovic, D. Urosevic, and E. Ngai. Variable neighborhood search for the heaviest k-subgraph. *Computer & Operations Research*, 36(11):2885–2891, 2009.

[6] F. Carrabs, C. Cerrone, R. Cerulli, and S. Silvestri. On the complexity of rainbow spanning forest problem. *Optimization Letters*, 12(3):443–454, 2018.

[7] F. Carrabs, C. Cerrone, R. Cerulli, and S. Silvestri. The rainbow spanning forest problem. *Soft Computing*, 22(8):2765–2776, 2018.

[8] D. Dorninger. Hamiltonian circuits determining the order of chromosomes. *Discrete Applied Mathematics*, 50:159–168, 1994.

[9] P. Festa. The shortest path tour problem: problem definition, modeling, and optimization. In *Proceedings of INOC'2009*, volume 217, pages 1–7. 2009.

[10] S. Fujita and C. Magnant. Properly colored paths and cycles. *Discrete Applied Mathematics*, 159:1391–1397, 2011.

[11] M.R. Garey and D.S. Johnson. *Computers and Intractability: A guide to the Theory of NP-Completeness*, volume 217. Freeman & Co., San Francisco CA, 1979.

[12] A. Lo. A dirac type condition for properly coloured paths and cycles. *J. Graph Theory*, 76:60–87, 2014.

[13] A. Lo. An edge-coloured version of dirac's theorem. *SIAM J. Discrete Math.*, 28:18–36, 2014.

[14] M.W. Padberg and M.R. Rao. Odd minimum cut-sets and b-matchings. *Mathematics of operations research*, 7(1):67–80, 1982.

[15] P.A. Pevzner. DNA physical mapping and alternating eulerian cycles in colored graphs. *Algorithmica*, 13:77–105, 1995.

[16] M. Szachniuk, M.C. De Cola, G. Felici, and J. Blazewicz. The orderly colored longest path problem - a survey of applications and new algorithms. *RAIRO - Operations Research*, 48:25–51, 2014.

[17] M. Szachniuk, M.C. De Cola, G. Felici, J. Blazewicz, and D. de Werra. Optimal pathway reconstruction on 3D NMR maps. *Discrete Applied Mathematics*, 182:134–149, 2014.

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