On the Shared Transportation Problem: Computational Hardness and Exact Approach

Tom Davot, Rodolphe Giroudeau, Jean-Claude König

On the Shared Transportation Problem: Computational Hardness and Exact Approach

Tom Davot[1], Rodolphe Giroudeau[2], and Jean-Claude König[2]

[1] Université de Technologie de Compiègne, CNRS, Heudiasyc, France
tom.davot@hds.utc.fr
[2] LIRMM - CNRS UMR 5506 - Université de Montpellier, France
giroudeau,konig@lirmm.fr

Abstract. In our modern societies, a certain number of people do not own a car, by choice or by obligation. For some trips, there is no or few alternatives to the car. One way to make these trips possible for these people is to be transported by others who have already planned their trips. We propose to model this problem using as path-finding problem in a list edge-colored graph. This problem is a generalization of the $s-t$-path problem, studied by Böhmová et al. We consider two optimization functions: minimizing the number of color changes and minimizing the number of colors. We study for the previous problems, the classic complexity (polynomial-case, NP-completeness, hardness of approximation) and parameter complexity (W[2]-hardness) even in restricted cases. We also propose a lower bound for exact algorithm. On the positive side we provide a polynomial-time approximation algorithm and a FPT algorithm.

Keywords: Complexity · Approximation · List edge coloring

1 Introduction

Shared mobility received a lot of attention in the last decades, both from industry and academics. The motivation behind this is ecological awareness, savings and social benefits. The rise of this kind of transportation is traduced by the apparition of mobility platforms and the emergence of scientific studies focusing on the different various relative questions. In particular, researchers in the field of operational research have been interested in studying various optimization problems resulting from shared mobility systems. In these systems, we seek to match people having similar itineraries on the same dates. A survey on ride-sharing systems can be found in [11]. The authors present a classification of different existing ride-sharing systems and identify some challenges. In [20], the

Preprint version. The editor version is available at https://doi.org/https://doi.org/10.1142/S0129054123410058

authors present dynamic ride-sharing systems. The authors show the need of optimization technologies for the success of this type of ride-sharing systems.

Different types of mobilities sharing systems exist. Carpooling is proposed by large companies to encourage their employees to share itineraries to and from work, in order to reduce the use of private cars. In dial-a-ride problems (DARP) [6, 7, 13], schedules and vehicles routes are designed based on user requests. In Vanpool problem [13], passengers drive to a park-and-ride location then they share their trips with a van to the target location. An exhaustive survey on optimization for shared mobility can be found in [19].

In this paper, we consider a problem where one person aims to travel from a place to another and can not make the trip by their own means. This can be due to several reasons: disability, absence of driving license, personal choice... In order to make the trip possible, we can use the help of drivers that have already planned their travel and offer to transport another person. We aim to match one or more driver that can share its/their trip. We model this problem as an $s-t$-path problem on list edge-colored digraph. We consider two objective functions. The first objective function is a color minimization which is a common objective for optimization problems on colored graphs. The second function, rather less classical, is to minimize the number of color changes along the path.

Results and related works Similar problems have been studied in the literature. In [2], the authors showed that the Minimum Label Path/Cycle Problem in undirected graph is NP-hard. The authors also provide some exact exponential-time and approximation algorithms to solve the problem. Another approximation algorithm and approximation hardness results have been presented in [12]. Some parameterized intractability results for minimum label path and other different minimum labeling problems have been presented in [9]. Other optimization problems on edge-labeled graphs have been considered in the literature. The minimum labeling spanning tree is widely studied [4, 5, 16]. The objective is to find a spanning tree such that the number of labels is the smallest possible. Another variant of the problem is considered in [22]. The problem is called Label-Constrained Minimum Spanning Tree Problem and the objective is to find the minimum weight spanning tree using at most k labels.

The number of problems in the literature using an edge-coloring or a list edge-coloring is large. We can cite the classic proper edge-coloring as example [3]. A close related problem has been proposed by Broersma et al. [2]: the aim is to find a path/cycle in a colored graph with a minimum number of colors. This problem is NP-hard even in bipartite planar. The authors also propose several exact and approximation algorithms. Finally, the complexity of the exact algorithms and the performance ratio of the approximation algorithms are also analyzed.

Our problem is equivalent to finding a path from s to t, using at most k colors, in an oriented graph G with a list of allowed colors for each arc. Each color represents a driver. It is a generalization of the st-Path problem studied by Böhmová et al.[1]. Whereas in the version studied by Böhmová et al. each subgraph induced by a color must be a path (representing a subway line), we have chosen not to put any restriction on these subgraphs in the problem formulation.

This enables us to model the route options that drivers could propose. However, in most of our results this restriction still holds.

Always in [1], some complexity results related to a subway network are presented: they propose an efficient algorithm for finding a *st*-route according to the number of line changes plus one. A non-approximation result, for the minimization of the used lines is proposed. Lastly, a polynomial-time algorithm is developed for the problem of enumerating all *st*-paths with a bounded length.

In the following we extend the complexity results of Böhmová et al. [1] by considering severals topologies or restricted cases.

Organization of the article. The next section is dedicated to notation and to the presentation of the two problems studied in this article. In Section 3, we present some restricted polynomial cases. Section 4 is devoted to the computational complexity according to the topology of the input graph (bipartite, planar, ...), the length of colored-path and the number of colors associated to each arc. In Section 5 we propose two negative results concerning the approximation and the parametrization. We develop a polynomial-time approximation algorithm in Section 6 and in Section 7, we show lower bounds for Exact algorithms. Finally, in Section 8, we present an FPT algorithm running in time $O(\binom{k}{C} \cdot n)$ with parameter the total number of colors in the graph, where k is the number of colors in the path, C is the number of colors in the graph and n is the number of vertices.

2 Problems description

2.1 Notation

Let $\mathcal{P}(\mathbb{N})$ be the powerset of naturals. In this article, we consider a specific oriented graph called *list arc-colored graph.* A list arc-colored graph $G = (V, A, \chi)$ is a graph with a set of vertices V, a set of arcs A and a function $\chi : A \mapsto \mathcal{P}(\mathbb{N}) \setminus \{\emptyset\}$ that associates a (sub-)set of colors to each arc. We denote by n and m the numbers of vertices and arcs of G, respectively. A *path* $P = (e_1, \dots, e_k)$ is a sequence of arcs such that there is a sequence of vertices $(v_1, \dots, v_{k+1})$ such that for each $i \leq k$, e_i is an output arc of v_i and an input arc of v_{i+1}. For each color i, we denote $G_i = (V_i, A_i)$ the subgraph induced by the arcs colored with color i. Formally, $A_i = \{e \mid i \in \chi(e)\}$ and $V_i = \{v \mid \exists uv \in A_i \vee \exists vu \in A_i\}$. Notice that the two extremities of an arc colored with color i belong to V_i. Let $\tilde{G}_i$ be the transitive closure of G_i (*i.e.* the arc uv belongs to $\tilde{G}_i$ if and only if there is a path from u to v in G_i) and we denote $\tilde{G} = \cup_i \tilde{G}_i$.

In the problems studied in this paper, given a list arc-colored graph G, the aim is to construct a *colored path* $\pi = (P, \lambda)$ where P is an oriented path of G and $\lambda : P \mapsto \mathbb{N}$ is an arc-coloring function. For each arc e of P, we attribute a unique color among $\chi(e)$, that is, $\lambda(e) \in \chi(e)$. The number of colors of π is denoted $\lambda_{\#}(\pi) = |\{\lambda(e) \mid e \in P\}|$. The number of *color changes*, denoted $\lambda_c(\pi)$ is the number of pairs of consecutive arcs of π that have different colors. Formally, let $P = (e_1, \dots, e_k)$ be a path, we have $\lambda_c(\pi) = |\{(e_i, e_{i+1}) \mid \lambda(e_i) \neq \lambda(e_{i+1})\}|$.

Problem	Topology	Complexity	
k-COLORED PATH	G_i is a bounded-length path	NP-C	Theorem 3
	Planar	NP-C	Theorem 5
	Bipartite	NP-C	Theorem 4
	Planar and bipartite	NP-C	Theorem 7
	General	LOG-APX-Hard	See [1]
	Path	LOG-APX-Hard	Corollary 3
	G_i is strongly connected	P	Theorem 2
	G_i is a path of length two	P	Lemma 2
k-COLOR CHANGE PATH	General	P	Theorem 1

Table 1. Overview of complexity results

Finally, we denote by $\pi[i]$ the subgraph of π induced by the arcs with color i. For simplicity, we sometimes denote $\pi[i]$ as the subgraph induced by the color i.

2.2 Objective functions

In this article, we consider two objective functions consisting of minimizing the number of colors or the number of color changes of a colored path. Hence, we define the two following problems.

k-COLORED PATH (k-CP)
Input: A list arc-colored oriented graph $G = (V, A, \chi)$, two given vertices s and t and a positive integer k.
Question: Is there a colored path π between s and t in G such that $\lambda_{\#}(\pi) \leq k$?

k-COLOR CHANGE PATH (k-CCP)
Input: A list arc-colored oriented graph $G = (V, A, \chi)$, two given vertices s and t and a positive integer k.
Question: Is there a colored path π between s and t such that $\lambda_c(\pi) \leq k$?

We study the complexity of these problems according to some graph topologies. An overview of the result is available in Table 1.

3 Polynomial cases

We present in this section some polynomial time algorithms for some specific cases related to the connectivity of colored subgraphs $G_i, \forall i$. For this, we show that the research for a shortest path in the modified graph $\tilde{G}$ guarantees obtaining an optimal solution.

Theorem 1. k-COLOR CHANGE PATH *admits a polynomial-time algorithm in* $O(n^3)$ *time.*

Algorithm 1 Polynomial-time Algorithm for k-COLOR CHANGE PATH and k-COLORED PATH

Require: A list arc-colored graph $G = (V, A, \chi)$, two vertices s and t.
Let $G_i = (V_i, A_i)$ be the subgraph induced by the arcs colored with color i.
Let $\tilde{G}_i$ be the transitive closure of G_i.
Let $\tilde{G} = \cup_i G_i$ be the input graph.
Apply Dijkstra algorithm on the graph $\tilde{G}$ with s and t.

Proof. Let G be a list arc-colored oriented graph. First, note that $\tilde{G}$ can be constructed in $O(n^3)$. We show that G contains a colored path π between s and t with $\lambda_c(\pi) = k$ if and only if there is an oriented path p of length at most $k+1$ between s and t in $\tilde{G}$.

- Let π be a colored path between s and t in G. For each monochromatic subpath $(u, \dots, v)$ of π, introduce the arc uv in p (uv exists in $\tilde{G}$ by definition). Since $\lambda_c(\pi) = k$, it exists $(k+1)$ monochromatic subpaths in π. Thus, we construct a oriented path of length $k+1$ in $\tilde{G}$.
- Let p be an oriented path of length $k+1$ between s and t in $\tilde{G}$. For each arc uv of p, it exists a monochromatic path p' between u and v in G, by construction of $\tilde{G}$. We add p' in π. Therefore, we obtain a colored path π between s and t in G that is constituted by at most $k+1$ maximal monochromatic paths. Hence, we obtain a path π with $\lambda_c(\pi) = k$.

Thus, we can construct an optimal colored path in G by computing a shortest path in $\tilde{G}$ and then apply the transformation described above. Since, a shortest path can be computed in $O(n^2)$ using Dijkstra's algorithm [10], the overall complexity is $O(n^3)$.

We propose to extend the previous result to k-COLORED PATH in some restricted case. So, we introduce the following lemma.

Lemma 1. *Let G be a list arc-colored oriented graph. k-*COLORED PATH *can be solved in time $O(n^3)$ if it exists an optimal colored path π such that for each color i, $\pi[i]$ is connected.*

Proof. Let G be a list arc-colored oriented graph. Using the same argument as in the proof of Theorem 1, we can show that G contains a colored path π respecting lemma's property with $\lambda_{\#}(\pi) = k$ between s and t if and only if there is a path p of length at most k between s and t in $\tilde{G}$. Hence, we can derive an optimal colored path in G by computing a shortest path in $\tilde{G}$ and by applying the transformation described in the proof of Theorem 1.

Theorem 2. *k-*COLORED PATH *admits an $O(n^3)$ time algorithm if for each color i of χ, G_i is strongly connected.*

Proof. Let G be a list arc-colored oriented graph such that each subgraph G_i is strongly connected and let π be a colored path between s and t. If there is a color

i such that $\pi[i]$ contains two non-connected subpaths $(u, \ldots, v)$ and $(u', \ldots, v')$, then since G_i is strongly connected, we can replace the subpath $(u, \ldots, v')$ in π by a path in G_i from u to v' without increasing $\lambda_{\#}(\pi)$. By doing that, we ensure that for each color i, $\pi[i]$ is connected. Thus, by Lemma 1, we conclude that k-COLORED PATH can be solved in $O(n^3)$-time in G.

Corollary 1. k-COLOR CHANGE PATH *in a non-oriented graph G admits a polynomial-time algorithm if $\forall i, G_i$ is connected.*

Hereafter, we propose a polynomial-time algorithm for the case of each subgraph G_i induced by a color i is a path of length at most two.

Lemma 2. k-COLORED PATH *can be solved in $O(n^3)$-time in graphs for which each color induces a path of length at most two.*

Proof. Let π be an optimal colored path. Suppose that there is a color i such that $\pi[i]$ is not connected. Let (v_1, v_2, v_3) be the path constituting G_i. Since both arcs v_1v_2 and v_2v_3 appears in π, then π is not an elementary path, contradicting its optimality. Hence, for each color i, $\pi[i]$ is connected and by Lemma 1, k-COLORED PATH can be solved in $O(n^3)$ in G.

4 Computational hardness

In this section, we consider k-COLOR CHANGE PATH problem in which each graph induced by a color is a path. We show that in that case, k-COLOR CHANGE PATH is NP-complete. We then show that it remains NP-complete even if the graph is bipartite or planar.

4.1 Each color induces a path of bounded length

We now show that k-COLORED PATH is NP-complete. We use a similar idea as the proof proposed for the NP-completeness of the problem of minimizing the number of used colored in an edge-colored graph [2]. We reduce from the following classical NP-complete problem.

3-SAT
Input: A Boolean formula φ where each clause contains exactly three literals
Question: Is φ satisfiable?

Construction 1 *Let φ be an instance of* 3-SAT *with m' clauses $C_0, \ldots, C_{m'-1}$ and n' variables $x_0, \ldots, x_{n'-1}$. For each variable x_i, let ψ_i (resp. $\overline{\psi}_i$) be the list of clauses where x_i appears positively (resp. negatively). We construct a list arc-colored graph $G = (V, A, \chi)$ as follows:*

- *create a vertex Q_m,*
- *for each variable x_i, create a vertex v_i,*

– *for each clause C_j create a vertex Q_j and create a vertex q_j^i, for each literal ℓ_i of C_j,*
– *for each variable x_i and for each clause C_j in which x_i appears, introduce an oriented path $(Q_j, q_j^i, v_i, v_{i+1})$ (or $(Q_j, q_j^i, v_{n-1}, Q_0)$ if $i = n-1$) with color c_j^i,*
– *for each variable x_i and for each pair of clauses (C_j, C_k) in ψ_i (resp. $\overline{\psi}_i$) such that $j < k$, introduce an oriented path $(Q_k, q_k^i, q_j^i, C_{j+1})$ with color $t_{j,k}^i$, and*
– *finally, for each variable x_i, let C_j be the last clause of ψ_i (resp. $\overline{\psi}_i$), introduce the arc (q_j^i, Q_{j+1}) with color z_i (resp. $\bar{z}_i$).*

Notice that the graph induced by each color is a path of length exactly one or three.

Theorem 3. *k-COLOR CHANGE PATH remains* NP*-complete even if each color induces a path of length at most three.*

Proof. Let φ be 3-SAT formula and G the graph obtained by Construction 1. Clearly, k-COLOR CHANGE PATH is in NP. We show that φ is satisfiable if and only if it exists a colored path between v_0 and Q_m with $\lambda_{\#}(\pi) = n + m$.

– Suppose that φ is satisfiable and consider ϕ a satisfying assignment for φ. Let $f_\phi : \{C_0, \ldots, C_{m-1}\} \mapsto \{x_0, \ldots, x_{n-1}\}$ be a function that assigns to each clause C_j a unique variable x_i such that the assignment of x_i satisfies C_j. We suppose that $f_\phi^{-1}(x_i)$ is an ordered list sorted in ascending order of the indices. We construct π as follows. For each variable x_i and for each clause $C_j \in f_\phi^{-1}(x_i)$:
  - If C_j is the first clause of $f_\phi^{-1}(x_i)$, add in π the outgoing arc of x_i with color c_j^i and the arc (C_j, q_j^i) with color c_j^i. Otherwise, let C_k be the clause that precedes C_j in $f_\phi^{-1}(x_i)$, add in π the two arcs (q_k^i, C_{k+1}) and (C_j, q_j^i) with color t_j^k.
  - If C_j is the last clause of $f_\phi^{-1}(x_i)$, add in π the arc (q_j^i, C_{j+1}) with color z_j.

  Since each clause C_j belongs to a list f_ϕ^{-1}, we construct a path between x_0 and C_m using $n + m$ colors.
– Let π be a path between x_0 and C_m using $n+m$ colors. Consider the function $g_\pi : \{x_0, ..., x_{n-1}\} \mapsto \mathcal{P}(\{C_0, \ldots, C_{m-1}\})$ defined by $g_\pi(x_i) = \{C_j \mid q_j^i \in \pi\}$. First, by construction for each vertex $x_i \in \pi$, we have $x_{i+1} \in \pi$ (or C_0 if $i = n-1$). Thus, π contains the subpath $P_{literals} = (x_0, \ldots, x_{n-1}, C_0)$ and uses clearly n colors in it. Moreover, always by construction, for each vertex $C_j \in \pi$, the only way to reach a vertex C_k, with $k > j$ from C_j is to take a path (C_j, q_j^i, C_{j+1}). Thus, by extension, π contains a subpath $P_{clauses} = (C_0, q_0^i, C_1, q_1^{i'}, \ldots, C_m)$ and so, each clause is contained in a set $g_\pi(x_i)$. Clearly, π uses a new color for each arc (q_j^i, C_{j+1}).
  Hence, since π can not use more than m colors in $P_{clauses}$, each arc $a_j = (C_j, q_j^i)$ is colored with a color already used in the subpath $(x_0, \ldots, C_j)$.

Hence, if $\lambda(a_j) = t_j^k$, then $(q_k^i, C_{k+1}) \in \pi$ and therefore, by construction, we have $\{C_k, C_j\} \subseteq \psi_i$ or $\{C_k, C_j\} \subseteq \overline{\psi}_i$. If $\lambda(a_j) = c_j^i$, then the arc (x_i, x_{i+1}) is colored with c_j^i in π. Since only one outgoing arc of x_i appears in π, by induction, we have $g_\pi(x_i) \subseteq \psi_i$ or $g_\pi(x_i) \subseteq \overline{\psi}_i$. If $g_\pi(x_i) \subseteq \psi_i$, we assign $x_i = \texttt{true}$ in ϕ and $x_i = \texttt{false}$ otherwise. The assignment of x_i satisfies every clause in $g_\pi(x_i)$ and since for each clause C_j, $g_\pi^{-1}(C_j)$ is defined, then ϕ is a satisfying assignment of φ.

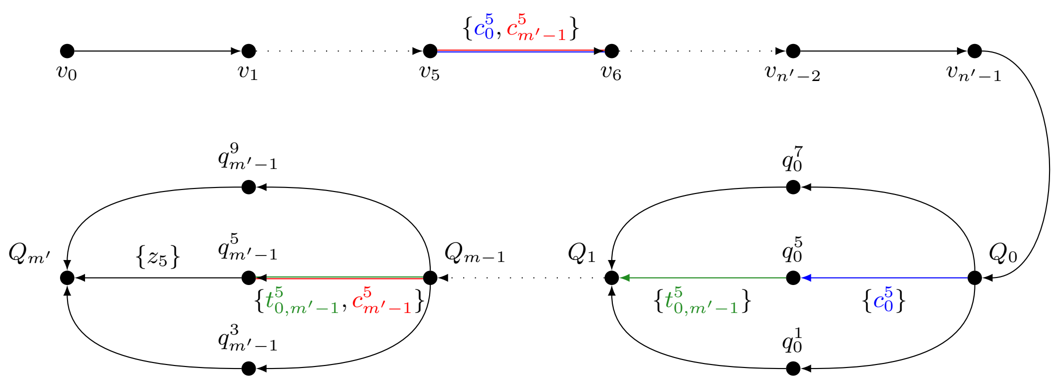

Fig. 1. Illustration of Construction 1. In this example, the literal x_5 appears in clauses $C_0 = (x_1, x_5, x_7)$ and $C_{m'-1} = (\bar{x}_3, x_5, \bar{x}_9)$. For simplicity, backward arcs $(q_0^5, v_5), (q_{m'-1}^5, v_5)$ and $(q_{m'-1}^5, q_0^5)$ are not drawn.

From Construction 1, it is easy to extend this result for the case where each path induced by a color has length exactly three: if a path has length one, we can extend it by adding two new vertices. Samewise, we can show that this problem remains NP-complete if every graph G_i is a disjoint union of arcs, by simply removing the backward arcs in the construction.

Corollary 2. *k-COLOR CHANGE PATH remains* NP*-complete even if:*

- *each color induces a path of length exactly three, or*
- *each color induces a collection of disjoint arcs.*

4.2 In more restricted cases

In the following, we reuse Construction 1 to extend the previous hardness result to even more restricted cases.

Let G be a graph resulting of Construction 1. We can make it bipartite by applying the following transformation.

Rule 1 *Let P_1 and P_2 be any partition of the vertices of G. For each arc $a = (v_1, v_2)$ such that $v_1 \in P_1$ and $v_2 \in P_2$, we introduce a new vertex x and replace a by the arcs (v_1, x) and (x, v_2).*

Notice that since it is possible to apply this transformation only to some backward arcs of G, the length of each path induced by a color is bounded by four. Hence, we obtain the following result.

Theorem 4. k-COLOR CHANGE PATH *is* NP*-complete even in bipartite graphs where each color induces a path of length at most four.*

Further, if we draw G as in Figure 1, only backward arcs can cross. In order to make such graph planar, we apply the classical technique consisting of adding a vertex for each arc intersection.

Rule 2 *Let (u, u') and (v, v') be two intersecting backward arcs with color i and j respectively. We introduce a new vertex v, remove (u, u') and (v, v') and we construct two paths (u, x, u') and (v, x, v') with color i and j, respectively.*

By using the previous rule, we add at most $O(m'^2)$ vertices to the construction. Notice that the lengths of the paths induced by a color are no longer bounded.

Theorem 5. k-COLOR CHANGE PATH *is* NP*-complete in planar graphs.*

Finally, we can show that this problem remains hard even if for each arc e we have $|\chi(e)| = 1$. We introduce the following rule.

Rule 3 *Let G be a list arc-colored graph. Let (u, v) be any arc such $|\chi(u, v)| \geq 1$ and let c be a color in $\chi(u, v)$. Emplace a new vertex v and construct the path (u, x, v) with color c. Finally, set $\chi(u, v) := \chi(u, v) \setminus \{c\}$.*

An example of this rule is given in Rule 3.

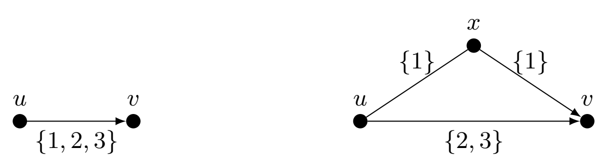

Fig. 2. Example of application of Rule 3.

Hence, we obtain the following theorem.

Theorem 6. k-COLOR CHANGE PATH *is* NP*-complete in list arc colored graphs where $|\chi(a)| = 1$ for each arc a.*

Finally, by combining the previous techniques, we obtain the following result.

Theorem 7. k-COLOR CHANGE PATH *remains* NP*-complete in planar bipartite list arc colored graphs where for each arc a, we have $|\chi(a)| = 1$.*

5 Hardness of approximation

We now show that k-COLOR CHANGE PATH in paths is equivalent to the classical problem SET COVER, defined as follows.

SET COVER (SC)
Input: A univers $U = (e_1, \dots, e_{n'})$ of n' elements, a collection $C = \{S_1, \dots, S_{m'}\}$ of m' subsets of U and a positive integer k.
Question: Is there a collection $C' \subseteq C$ such that $\bigcup_{S_i \in C'} S_i = U$ and $|C'| \leq k$?

Construction 2 *Let (U, C) be an instance of* SET COVER*, we construct a list-arc colored graph G as follows:*

- *construct an oriented path $(v_1, v_{n'+1})$, and*
- *for each subset $S_j \in C$ and each element $e_i \in S_j$, color the arc (v_i, v_{i+1}) with color j.*

An example of graph produced by Construction 2 is depicted in Figure 3.

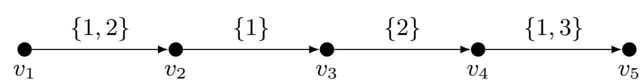

Fig. 3. Example of graph produced by Construction 2 on the univers containg the sets $S_1 = \{1, 2, 4\}, S_2 = \{1, 3\}$ and $S_3 = \{4\}$.

Theorem 8. *The optimization version of k-*COLORED PATH *is LOG-APX-hard even in paths.*

Proof. Let (U, C) be an instance of SET COVER and let G be its list arc-colored graph resulting from Construction 2. We show that (U, C) admits a set cover of size k if and only if G contains a colored path π between v_1 and $v_{n'+1}$ with $\lambda_{\#}(\pi) = k$.

- Let C' be a minimal set cover of (U, C) of size k. We construct π as follows. For each element e_i, let $S_j \in C'$ containing x_i. Add the arc $v_i v_{i+1}$ with color j in π. Clearly π cannot use more colors than $|C'|$, thus we obtain a colored path π between v_1 and $v_{n'+1}$ with $\lambda_{\#}(\pi) \leq |C'|$.
- Let π be a colored path between v_1 and $v_{n'+1}$ such that $\lambda_{\#}(\pi) = k$. We construct the following set cover $C' = \{S_j \mid \lambda(v_i v_{i+1}) = j\}$. Since G is an oriented path, for each vertex v_i, the arc $v_i v_{i+1}$ belongs to π. Thus, by construction, for each element $e_i \in U$, e_i is contained in the subset S_j, where $\lambda(x_i, x_{i+1}) = j$. Hence, C' is a set cover of (C, U).

Suppose it exists a polynomial-time algorithm A that can approximate k-COLORED PATH with a factor $R(G)$. Then, we can obtain a polynomial-time algorithm with the same approximation factor for SET COVER by applying successively Construction 2, A and the transformation to obtain a set cover from a colored path described above. Lund and Yannakakis show that SET COVER can not be approximated with a factor better than a logarithmic function [18] if $P \neq \text{NP}$. Hence, it implies that k-COLORED PATH can not be approximated in polynomial-time with a factor better than a logarithmic function if $P \neq \text{NP}$.

We now reduce k-COLORED PATH in paths to SET COVER.

Construction 3 *Let G be a list-arc colored path on the vertices $(v_1, \ldots, v_n)$. We construct an instance of* SET COVER (U, C) *as follows:*

- *construct the univers $U = \{e_1, \ldots, e_{n-1}\}$,*
- *for each color j, introduce a set S_j, and*
- *for each arc $a = (v_i, v_{i+1})$ and each color $j \in \chi(a)$, emplace the element e_i in S_j.*

Notice, that the previous construction is the inverse function of Construction 2. Thus, we can reuse the same argument as in Theorem 8 to show the following.

Lemma 3. *Let G be a list-arc colored path and (U, C) be its instance of* SET COVER *resulting of Construction 3. It exists a colored path π between v_1 and v_n with $\lambda_{\#}(\pi) = k$ if and only if it exists a set cover of (U, C) of size k.*

Corollary 3. SET COVER $\equiv$ k-COLORED PATH *in paths.*

Recall that SET COVER is $W[2]$-hard when parameterized by the score of the solution [8]. We can use Construction 2 to show that k-COLORED PATH is $W[2]$-hard when parameterized by the number of used colors in the path. We recall that a parameterized problem (Π_1, κ_1) is FPT-reductible to another problem (Π_2, κ_2) if it exists two applications f and g such that:

- given an instance (I_1, k_1) of (Π_1, k_1), f constructs in FPT-time an instance (I_2, k_2) of (Π_2, κ_2) such that (I_2, k_2) is a yes-instance if and only if (I_1, k_1) is a yes-instance,
- $g : \mathbb{N} \mapsto \mathbb{N}$ is a computable function such that $k_2 \leq g(k_2)$.

We now show the following

Theorem 9. k-COLORED PATH *is $W[2]$-hard when parameterized by k.*

Proof. Let (U, C) be an instance of SET COVER and let G be the list-arc colored produced by Construction 2. First, note that G is constructed in $\mathcal{O}(|U| + |C|)$ time. Second, for any value k, it exists a set cover of size k for (U, C) if and only if G contains a path between v_1 and $v_{n'+1}$ with k colors. Hence, we can set the function g as the function $g(k) = k$ to show that Construction 2 is a FPT-reduction.

6 Approximation results

In the following, we consider the problem in which each subgraph G_i is an oriented path of length at most $\ell_i \leq \ell$.

We develop a polynomial-time approximation algorithm based on the computation of a shortest path in $\tilde{G}$. As for the algorithms of Section 3, the overall time complexity of this approximation algorithm is $O(n^3)$.

Lemma 4. *Let G be a list arc-colored graph such that each subgraph G_i is an oriented path of length at most ℓ and let π be a colored path between s and t. For each color i, $\pi[i]$ contains at most $\lceil \frac{\ell}{2} \rceil$ connected components.*

Proof. Let $v_i v_j$ be an arc of π colored with color c. Let $v_j v_k$ be the outgoing arc of v_j in G_i. Since π is elementary, either π contains the subpath (v_i, v_j, v_k), or π does not contain $v_j v_k$. Hence, two consecutive arcs of G_i cannot appear in different connected components of $\pi[i]$ and then $\pi[i]$ contains at most $\lceil \frac{\ell}{2} \rceil$ connected components.

Theorem 10. *Let G be a list arc-colored graph such that each subgraph G_i is an oriented path of length at most ℓ. An optimal solution of k-COLOR CHANGE PATH in G is a $\lceil \frac{l}{2} \rceil$-approximation of k-COLOR CHANGE PATH.*

Proof. Let π be a colored path between s and t. By Lemma 4, we have

$$\lambda_c(\pi) \leq \lceil \frac{\ell}{2} \rceil \cdot \lambda_{\#}(\pi). \tag{1}$$

Let π_{opt} be an optimal solution of k-COLOR CHANGE PATH and π_{app} be an optimal solution of k-COLOR CHANGE PATH. We have

$$\lambda_{\#}(\pi_{app}) \leq \lambda_c(\pi_{app}) \tag{2}$$

and

$$\lambda_c(\pi_{app}) \leq \lambda_c(\pi_{opt}). \tag{3}$$

Thus,

$$\lambda_{\#}(\pi_{app}) \overset{(2)}{\leq} \lambda_c(\pi_{app}) \overset{(3)}{\leq} \lambda_c(\pi_{opt}) \overset{(1)}{\leq} \lceil \frac{\ell}{2} \rceil \cdot \lambda_{\#}(\pi_{opt}). \tag{4}$$

Therefore, we obtain

$$\frac{\lambda_{\#}(\pi_{app})}{\lambda_{\#}(\pi_{opt})} \leq \lceil \frac{\ell}{2} \rceil.$$

7 Lower bounds for Exact algorithms

We propose some negative results for k-COLOR CHANGE PATH about the existence of subexponential-time algorithms under ETH [14, 15].

Corollary 4. *There is no $2^{o(n)}$ (resp. $2^{o(\sqrt{n+m})}$)-time algorithm for the optimization version of k-*COLORED PATH *even in graphs where each color induces a path of length at most three or in bipartite graphs where each color induces a path of length at most four (resp. in bipartite planar graphs).*

Proof. Let φ be a 3-SAT formula with n' variables and m' clauses and G be its list arc-colored graph resulting from Construction 1. By construction, the number of arcs and vertices of G is $O(n' + m')$, even if we make the graph bipartite. Thus, since 3-SAT does not admit a $2^{o(n'+m')}$-time algorithm, k-COLORED PATH does not admit a $2^{o(n+m)}$-time algorithm [14, 17, 21]. Making the graph planar as described for Theorem 5 adds $O(m'^2)$ vertices in G. Thus, we can conclude that k-COLORED PATH does not admit a $2^{o(\sqrt{|V|+|E|})}$-time algorithm in bipartite planar graphs.

8 Fixed-parameter tractable algorithm for *k*-CP

In this section, we describe fixed-parameter tractable algorithm for k-COLORED PATH, parameterized by the total number of colors $C = |\bigcup_{e \in A} \chi(e)|$ in the graph $G = (V, A, \chi)$. We first introduce some specific notations and definitions. Let ℓ and ℓ' be two color sets. We say that ℓ' is *dominated* by ℓ if $\ell \subseteq \ell'$. Notice that it is a transitive relation: given l_1, l_2 and l_3, if l_1 dominates l_2 and l_2 dominates l_3, then l_1 dominates l_3. Let π be a colored path. We denote by $\lambda(\pi)$ the colors used in π, that is, $\lambda(\pi) = \bigcup_{a \in \pi} \lambda(a)$.

For our algorithm, we use the following semantic: for each vertex v, we introduce a list of color sets $L(v)$ such that for each color set ℓ of $L(v)$ it exists a colored path p between s and v such that $\chi(p) = \ell$.

Lemma 5. *Algorithm 2 returns* `true` *if and only if there is a solution for k-*CP *and runs in $O(\binom{k}{C} \cdot n)$ time.*

Proof. First, note that a vertex u can be added to V' by a vertex v if L' contains a set of colors of size at most k that is not dominated by a set of colors of $L'(u)$. Hence, we have $|L(u)| \leq \binom{k}{C}$ and u can be added to V' at most $|L(u)|$ times. It follows that the while loop is not endless and thus, the algorithm stops.

Further, we show that Algorithm 2 returns true if and only it exists a colored path π with $\lambda_{\#}(\pi) \leq k$.

"⇒" Suppose that the algorithm returns true. We show by induction that for any vertex u and for any color set $\ell \in L(u)$, there is a colored path π between s and u with $\lambda(\pi) = \ell$. First, it is clear that there is an empty path from s to itself using no color. Later, let u be any vertex and $\ell' \in L(u)$. Let v be the vertex considered by the while loop when ℓ' is added to $L(u)$ (line 17). Then, it exists a color set $\ell \in L(v)$ and a color $c \in \chi(v, u)$ such that $\ell \cup \{c\} = \ell'$. By induction hypothesis, there is a colored path π between s and v with $\lambda(\pi) = \ell$. Let $\pi' = \pi \cup \{(v, u)\}$ with $\lambda(v, u) = c$, we have $\lambda(\pi') = \lambda(\pi) \cup \{c\}$. Thus, π' is a colored path from s to u with $\lambda(\pi') = \ell'$. Finally, the algorithm

Algorithm 2 FPT Algorithm

```
Require: A list arc-colored graph (G, χ), two vertices s and t and an integer k. V' ←
  {t}
  for all v ∈ V(G) do
      L(v) = ∅
  end for
  while V' ≠ ∅ do
      if v == t then
          return true;
      end if
  end while
  v ← any element of V';
  V' ← V' − v;
  for all u ∈ N⁺(v) do
      L' ← {ℓ ∪ {c} | ℓ ∈ L(v), c ∈ χ(v, u)};
      for all ℓ' ∈ L' do
          if |ℓ'| > k or ∃ℓ ∈ L(u), ℓ ⊆ ℓ' then
              L' ← L' − ℓ';
          end if
      end for
      if L' ≠ ∅ then
          L(u) ← L(u) ∪ L';
          V' ← V' ∪ {u};
      end if
      return false;
  end for
```

returns true the first time t is considered by the while loop. In that case, $L(t)$ is non-empty. Hence, it exists a color set $\ell \in L(t)$ with $|\ell| \leq k$ and by the above property, there is a colored path between s and t using less than $|\ell|$ colors.

"$\Leftarrow$" Let $\pi = (v_1, \ldots, v_i)$ be a colored path such that $v_1 = s$ and $v_i = t$. We show by induction that, if the algorithm does not return true before, for each vertex $v_j \in \pi$, there is a state in which $L(v_j)$ contains a color set ℓ that dominates $\lambda(v_1, \ldots, v_j)$. Clearly, the property is true for v_1. Let $v_j \in \pi$. By induction hypothesis, there $L(v_{j-1})$ contains a color set ℓ that dominates $\lambda(v_1, \ldots, v_{j-1})$. Moreover, when ℓ is added to $L(v_{j-1})$ (line 17), v_{j-1} is added to V' (line 18) in the same loop iteration. Hence, v_{j-1} is considered by the while loop when $\ell \in L(v_{j-1})$. Further, let $\ell' = \ell \cup \lambda(v_{j-1}, v_j)$, when v_{j-1} is considered, two cases can happen: either ℓ' is added to $L(v_j)$ or $L(v_j)$ already contains a color set that dominates ℓ'. In any case, $L(v_j)$ contains a color set that dominates $\lambda(v_1, \ldots, v_j)$. Hence, it exists a state in which $L(v_i)$ contains a color set ℓ that dominates $\lambda(\pi)$. Moreover, since $L(t)$ is not empty, t is considered by the while loop and in that case, the algorithm returns true (line 5).

9 Conclusion

In this paper, we tackle the trip sharing problem in complexity and approximation viewpoints. We show that in the case of each input colored graph, for a fixed color, has a length at most two, the problem is polynomial whereas the problem becomes NP-complete for each colored path has length three. The complexity results are supplemented by hardness results according to topology (planar, bipartite and bipartite planar). On positive side, we develop a polynomial-time approximation algorithm a ratio at most $\lceil \frac{l}{2} \rceil$ with l the length of each input colored path. We also propose some lower bounds according to ETH, and parameter complexity viewpoint. Finally, we present a FPT algorithm running in time in $O(\binom{k}{C} \cdot n)$. The challenge consists in finding a $O(\log n)$-approximation for the problem of minimizing the number of colors. In such case k-COLORED PATH would be in Log-APX-complete.

Acknowledgment

This work was carried out and funded in the framework of the Labex MS2T. It was supported by the French Government, through the program "Investments for the future" managed by the National Agency for Research (Reference ANR-11-IDEX-0004-02).

Bibliography

[1] K. Böhmová, L. Häfliger, M. Mihalák, T. Pröger, G. Sacomoto and M. Sagot, Computing and listing st-paths in public transportation networks, *Theory of Computing System* **62**(3) (2018) 600–621.

[2] H. Broersma, X. Li, G. J. Woeginger and S. Zhang, Paths and cycles in colored graphs, *Australasian Journal of Combinatorics* **31** (2005) 299–312.

[3] Y. Cao, G. Chen, G. Jing, M. Stiebitz and B. Toft, Graph edge coloring: A survey, *Graphs and Combinatorics* **35**(1) (2019) 33–66.

[4] M. E. Captivo, J. C. Clímaco and M. M. Pascoal, A mixed integer linear formulation for the minimum label spanning tree problem, *Computers & Operations Research* **36**(11) (2009) 3082–3085.

[5] A. M. Chwatal and G. R. Raidl, Solving the minimum label spanning tree problem by mathematical programming techniques, *Advances in Operations Research* **2011** (2011).

[6] J.-F. Cordeau and G. Laporte, The dial-a-ride problem (darp): Variants, modeling issues and algorithms, *Quarterly Journal of the Belgian, French and Italian Operations Research Societies* **1**(2) (2003) 89–101.

[7] J.-F. Cordeau and G. Laporte, The dial-a-ride problem: models and algorithms, *Annals of Operations Research* **153** (Sep 2007) 29–46.

[8] M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk and S. Saurabh, *Parameterized Algorithms* (Springer, 2015).

[9] M. R. Fellows, J. Guo and I. Kanj, The parameterized complexity of some minimum label problems, *Journal of Computer and System Sciences* **76**(8) (2010) 727–740.

[10] M. L. Fredman and R. E. Tarjan, Fibonacci heaps and their uses in improved network optimization algorithms, *25th Annual Symposium onFoundations of Computer Science, 1984.*, (1984), pp. 338–346.

[11] M. Furuhata, M. Dessouky, F. Ordonez, M.-E. Brunet, X. Wang and S. Koenig, Ridesharing: The state-of-the-art and future directions, *Transportation Research Part B: Methodological* **57** (2013) 28 – 46.

[12] R. Hassin, J. Monnot and D. Segev, Approximation algorithms and hardness results for labeled connectivity problems, *Journal of Combinatorial Optimization* **14**(4) (2007) 437–453.

[13] S. C. Ho, W. Szeto, Y.-H. Kuo, J. M. Leung, M. Petering and T. W. Tou, A survey of dial-a-ride problems: Literature review and recent developments, *Transportation Research Part B: Methodological* **111** (2018) 395 – 421.

[14] R. Impagliazzo and R. Paturi, On the complexity of k-sat, *Journal of Computer and System Sciences* **62**(2) (2001) 367–375.

[15] R. Impagliazzo, R. Paturi and F. Zane, Which problems have strongly exponential complexity?, *Journal of Computer and System Sciences* **63**(4) (2001) 512–530.

[16] S. O. Krumke and H. Wirth, On the minimum label spanning tree problem, *Information Processing Letters* **66**(2) (1998) 81–85.

[17] D. Lokshtanov, D. Marx and S. Saurabh, Lower bounds based on the Exponential Time Hypothesis, *Bulletin of the EATCS* **105** (2011) 41–72.

[18] C. Lund and M. Yannakakis, On the hardness of approximating minimization problems, *Journal of ACM* **41**(5) (1994) 960–981.

[19] A. Mourad, J. Puchinger and C. Chu, A survey of models and algorithms for optimizing shared mobility, *Transportation Research Part B: Methodological* **123** (2019) 323 – 346.

[20] A. Niels, H. Agatz, A. L. Erera, M. W. P. Savelsbergh and X. Wang, Optimization for dynamic ride-sharing: A review, *European Journal of Operational Research* **223**(2) (2012) 295–303.

[21] G. J. Woeginger, Exact Algorithms for NP-Hard Problems: A Survey, *Combinatorial Optimization*, (2001), pp. 185–208.

[22] Y. Xiongm, B. Golden, E. Wasil and S. Chen, *The Label-Constrained Minimum Spanning Tree Problem*, *Telecommunications Modeling, Policy, and Technology* eds. S. Raghavan, B. Golden and E. Wasil (Springer US, Boston, MA, 2008), Boston, MA, pp. 39–58.