On the hardness of Edge Coloring Problems Associated with Scheduling

Michele Barbato[1], Diego Delle Donne[2], and Emiliano Lancini[3]✉

[1] Dipartimento di Informatica, Università degli Studi di Milano, Milan, Italy
michele.barbato@unimi.it,
[2] IDS Department, ESSEC Business School, Cergy-Pontoise, France
delledonne@essec.edu
[3] Université Paris Dauphine - PSL, LAMSADE, CNRS, UMR 7243, Paris, France
emiliano.lancini@lamsade.dauphine.fr.

Abstract. In this work, we introduce several Edge Coloring problems related with scheduling and we study their computational complexity. In particular, we prove that the Concurrent Open Shop Coloring is NP-hard. This problem can be summarized as a unitary-time Open Shop Problem with a hard time horizon constraint, in which the goal is to minimize the total processing time of the tasks. We prove that this problem is NP-hard by reducing it to a new variant of the edge coloring problem, named the Mono-Polychromatic Edge Coloring. We study feasibility and hardness of this problem, both for the general case and when the underlying graph is bipartite. We show that the latter case is equivalent to the Vertex Coloring Problem.

Keywords: edge coloring, open shop, scheduling

This preprint has not undergone peer review or any post- submission improvements or corrections. The Version of Record of this contribution is published in [insert volume title], and is available online at https://doi.org/[insert DOI]

1 Introduction

The *Open Shop Scheduling* problems form a class of optimization problems in which a set of jobs, each composed by tasks, must be scheduled on a set of machines without observing any prescribed order on the execution of the tasks. Depending on the specific problem, several objectives are considered, such as the minimization of the makespan, of the total tardiness, of the number of late jobs, or other criteria commonly adopted in scheduling. Regardless of the objective function, these problems demand each task to be executed each on a pre-specified machine, and each machine can execute at most one task at once. The *Concurrent Open Shop Scheduling* problems allow different tasks of a job to be executed in parallel. When makespan minimization is considered as the objective function, both these problems are known to be NP-hard, as proved by

Williamson [9] for the Open Shop Scheduling, and by Roemer [6] for the Concurrent Open Shop Scheduling. The Open Shop Scheduling and Concurrent Open Shop Scheduling problems are motivated by a wide range of applications, including product design and product assembly [1], and airplane maintenance [8]. We refer to the monograph of Kubiak [4] for a more extensive discussion. A natural approach to deal with scheduling problems is to transform them into coloring problems on graphs. In particular, this approach has been used to investigate the interactions between Concurrent Open Shop Scheduling and vertex coloring in Ilani, Grinshpoun, and Shufan [2]. Another natural equivalence exists between Open Shop Scheduling with Unit-time Tasks and the Edge Coloring Problem on bipartite graphs [4].

In our work, we consider the *Concurrent Open Shop Coloring* (COSC), a variant of the Concurrent Open Shop Scheduling problem where tasks have unitary duration. We are given a time horizon, a set of jobs and a set of processors. The unit-time tasks composing each job have to be processed on any order, each task on a pre-specified processor. Processor can interact once in the time horizon; this operation costs exactly one unit of time that the processors do not use for executing tasks. Our goal is to minimize the sum of the processing times of each job.

We show that COSC is NP-hard, by a reduction from a novel type of edge-coloring problem on bipartite graphs, for which we first give a proof of its NP-hardness.

Outline. In Section 2 we define the Concurrent Open Shop Coloring both in graph-theoretical terms and as a binary linear program. We discuss basic feasibility results and bounds on the objective function. In Section 3 we introduce and study a novel edge-coloring problem which is used to prove that the Concurrent Open Shop Coloring is NP-hard and we characterize its feasibility. To conclude, we study the computational complexity and feasibility characteristics of this problem on a particular subfamily of instances, namely, the case when the underlying graph is bipartite and simple.

2 Definitions and Preliminary Results

In the Concurrent Open Shop Coloring (COSC) problem, we are given a set of jobs, a set of processors, and a time horizon of length $k \in \mathbb{Z}_{>0}$. Each job is composed by a set of distinct unit-time tasks each of which must be executed on a pre-specified processor. No prescribed order is given on the execution of the tasks, and the tasks can be executed in parallel, as long as these are executed in different processors. Processors execute at most one task at the time. Moreover, we allow processor to interact one with another once during the time span. When present, this interaction takes one time unit during which the two processors cannot execute any task. The *processing time* of a job is the number of different (unitary) time slots in which at least one task of the job is being processed. The goal of COSC is to minimize the sum of the processing times of each job.

The following gives a model for COSC stating it as a graph coloring problem. Let V be a set of vertices partitioned into subsets J and P representing the jobs and the machines, respectively. Let $G = (V, E)$ be the graph whose edge-set E is such that:

– for $j \in J$ and $p \in P$, edge jp exists if and only if the job represented by j has a task that must be executed on the processor represented by p;
– for $p, q \in P$, edge pq exists if and only if the two corresponding processors must interact.

Representing each time slot with a color in the set $[k] := \{1, 2 \ldots, k\}$, we want to find an assignment of colors to the edges in E such that no two edges incident to a same vertex in P receive the same color. The objective is to minimize the sum over all vertices $j \in J$ of the number of distinct colors used on the edges of $\delta(j) := \{e \in E \colon e \text{ is incident to } j\}$.

Using the above graph-theoretical representation, COSC can be formalized as a binary linear program as follows. Let x_{ec} be a binary variable equal to 1 if and only if color $c \in [k]$ is assigned to edge $e \in E$, and, for all $j \in J$, let y_{jc} be a binary variable equal to 1 if and only if color c is assigned to at least one edge of $\delta(j)$. With these definitions, COSC amounts to solve:

$$
\begin{aligned}
\min \quad & \sum_{j \in J} \sum_{c \in [k]} y_{jc} & & (1)\\
\text{s.t.} \quad & \sum_{c \in [k]} x_{ec} = 1 & \forall e \in E, & \quad (2)\\
& x_{ec} \leq y_{jc} & \forall e \in \delta(j), \forall j \in J, \forall c \in [k], & \quad (3)\\
& \sum_{e \in \delta(p)} x_{ec} \leq 1 & \forall p \in P, \forall c \in [k], & \quad (4)\\
& y_{jc} \in \{0, 1\} & \forall j \in J, \forall c \in [k], & \quad (5)\\
& x_{ec} \in \{0, 1\} & \forall e \in E, \forall c \in [k]. & \quad (6)
\end{aligned}
$$

A straightforward lower bound for COSC is $|J|$. Moreover, the problem is clearly NP-hard, as it contains as a special case the edge-coloring problem (when $J = \emptyset$). This link motivates the following observation.

Proposition 1. *The following hold:*

1. *COSC admits a feasible solution whenever $k \geq \Delta(P)+1$, where $\Delta(P)$ denotes the maximum degree of the vertices in P.*
2. *If every vertex in P is adjacent to at least one vertex in J, then COSC admits a feasible solution if and only if $k \geq \Delta(P)$, and we can build such a solution in polynomial time.*

Proof. We first observe that if $k < \Delta(P)$, then COSC is infeasible.

Let G be the graph underlying COSC, and let G' be the graph built by removing each vertex j in J, adding one vertex t for each task t of the job associated to j, and joining this vertex with the vertex associated to the processor that must execute that task. Then, finding a feasible solution to COSC on G is equivalent to finding a proper edge-coloring in G'. By Vizing's theorem [7], we need at most $\Delta(G') + 1$ colors to do so. Since all the vertices outside P in G' have degree 1, $\Delta(G') = \Delta(P)$, and this proves 1.

For 2, let G'' be the subgraph of G' induced by P. Since every vertex in P is adjacent to at least one vertex outside P, we have that $\Delta(G') \geq \Delta(G'') + 1$. We can properly edge-color G'' with $\Delta(G'')+1$ colors in polynomial time using Misra and Gries algorithm [5]. Then, we can extend such coloring to G' by greedily coloring the remaining edges. Since each of these edges is incident to a vertex of degree 1, the resulting edge-coloring uses exactly $\Delta(G') = \Delta(P)$ colors. □

Proposition 1 states that a feasible solution to COSC can be found in polynomial time if either $k \geq \Delta(P)+1$ or if each processor executes at least one task. Consequently, one might initially surmise that COSC can be solved in polynomial time under these circumstances. However, contrary to this expectation, we demonstrate that COSC remains NP-hard, even when these conditions are met. This holds true even when there are no edges connecting vertices in P, in which case the underlying graph is bipartite.

3 The Mono-Polychromatic Edge Coloring Problem

We prove that the COSC with $k \geq \Delta(P) + 1$ is NP-hard by considering an auxiliary edge-coloring problem defined as follows.

Let $G = (V, E)$ be graph, with V partitioned into two subset of vertices, M and P. The *Mono-Polychromatic Edge Coloring* (MPEC) asks to minimize the number of colors assigned to the edges of G so that for every vertex $v \in P$, all edges incident to v are assigned distinct colors, and for every vertex $w \in M$, all edges incident to w receive the same color. The vertices in P are called *polychromatic*, those in M are called *monochromatic*. To simplify the proofs below we assume that E may contain parallel edges.

A graph is said to be *k-MP-colorable* if there exists a solution to MPEC that uses at most k colors. Even for k arbitrarily large, not all graphs admit such a coloration, as it is depicted with the example in Figure 1: edges w_1v and w_2v must have different colors, however both edges must have the same color as the edge w_1w_2, which is a contradiction. We say that a graph is *MP-colorable* if it is k-MP-colorable for some $k > 0$.

To characterize MP-colorable graphs we define the following operation: an *M-contraction* consists in contracting an edge e incident to two vertices in M, and assigning to M the resulting vertex. This operation possibly produces parallel edges.

Observation 2 *Deleting edges between vertices in P and/or M-contracting edges do not affect MP-colorability.*

From this observation we reach the following result.

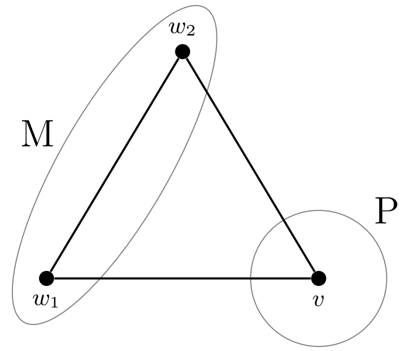

Fig. 1. A graph that does not admit a k-MP-coloring, for any $k \in \mathbb{Z}$.

Proposition 3. *A graph G is MP-colorable if and only if it has no circuit containing exactly one polychromatic vertex.*

Proof. First suppose that G is MP-colorable and has a circuit C composed by only monochromatic vertices except for one polychromatic vertex v. Then, by repeatedly contracting all edges connecting monochromatic vertices in C, we obtain a circuit of length 2 $\{v, w\}$, with $v \in P$ and $w \in M$. By the condition on the polychromatic vertex v, the two parallel edges of the M-contracted circuit must have different colors, but this contradicts the condition on the monochromatic vertex w. Therefore, this contracted instance is not MP-colorable and, by Observation 2, neither is G.

On the other hand, suppose that G has no circuit containing exactly one polychromatic vertex. Let G' be the graph obtained by applying all possible M-contractions and deleting all edges between polychromatic vertices. By Observation 2, if G' is MP-colorable, then so is G. We have that G' is bipartite, as there are no edges between vertices in M (resp. in P).

We have that G is MP-Colorable if and only if G' is simple, as we now show. If G' has two parallel edges, then it is not MP-colorable, hence the same holds for G. If G' is simple, let $v_1, v_2, \ldots, v_m$ be the monochromatic vertices of G'; assigning color i to all edges in $\delta(v_i)$, for all $i = 1, 2, \ldots, m$ yields an MP-coloring that uses m colors.

Therefore to conclude the proof, we suppose by contradiction that there are two parallel edges e and f in G' incident to a vertex $v \in P$. Graph G has no parallel edges between a monochromatic and a polychromatic vertex as otherwise it would be a circuit with exactly one polychromatic vertex. Hence, there are two vertices in M, say u and w, that are endpoints of e and f respectively. Since u and w are identified in the same vertex in G', and this vertex represents a connected subgraph of monochromatic vertices in G. Hence, in this subgraph there is a path $u, v_1, \ldots, v_h, w$ in G such that all v_i are monochromatic vertices. But then, $v, u, v_1, \ldots, v_h, w, v$ is a circuit with exactly one vertex in P, a contradiction. □

As byproduct of the proof of Proposition 3, we have that bipartite simple graphs with shores M and P where M (resp. P) contains precisely the monochromatic (resp. polychromatic) vertices are always MP-colorable. Nonetheless we now show that finding the minimum k for which such a graph is k-MP-colorable is theoretically difficult. In the following we denote by b-MPEC the latter problem.

In our proof we show that the b-MPEC is equivalent to the *vertex coloring problem* (VCP) on general graphs, that is, the problem of assigning the minimum number of colors to the vertices of a graph so that adjacent vertices are assigned distinct colors.

Proposition 4. *The two following reductions hold true:*

1. *Every instance $\mathcal{V}$ of the VCP admits one instance $\mathcal{M}$ of the b-MPEC such that $\mathcal{M}$ and $\mathcal{V}$ have the same value.*
2. *Every instance $\mathcal{M}$ of the b-MPEC admits one instance $\mathcal{V}$ of the VCP such that $\mathcal{M}$ and $\mathcal{V}$ have the same value.*

Moreover, in each reduction the solution of one problem can be constructed from the solution of the other problem in polynomial time with respect to the size of the instance.

Proof. We first prove point 1. Let $G = (V, E)$ be the graph defining instance $\mathcal{V}$ of the VCP. We subdvide every edge $e \in E$ with a new vertex. Let P and F respectively be the sets of new vertices and edges generated by the subdivision. Let us consider the instance $\mathcal{M}$ of the b-MPEC defined over $B = (P \cup V, F)$. An optimal solution to $\mathcal{V}$ gives a feasible solution for $\mathcal{M}$: for every $v \in V$ assign to every edge $f \in \delta(v)$ the color assigned to v in the optimal solution to $\mathcal{V}$. This solution is feasible because a vertex $w \in P$ is incident to precisely the two edges $f_1, f_2 \in F$ obtained by subdividing $e \in E$; since the endpoints of e have distinct colors, also f_1 and f_2 have distinct colors and the condition on vertices in P is satisfied. The condition on the vertices of V (which are monochromatic) is also fulfilled by definition. Conversely, an optimal solution to $\mathcal{M}$ is feasible for $\mathcal{V}$ by assigning to each vertex $v \in V$ the color assigned to its incident edges in that optimal solution. Such a color is unique because the vertices in V are monochromatic, by definition of B, and thus satisfy the condition on monochromatic vertices. Moreover, by the condition on P, adjacent vertices of G receive distinct colors. Therefore point 1 holds.

Point 2 can be proved similarly after replacing with an edge (v_1, v_2) each 2-path (v_1, w, v_2) linking two polychromatic vertices $v_1, v_2 \in P$ of the bipartite graph $B = (P, M; E)$. These operations transform an instance of the b-MPEC into an instance of the VCP. ☐

Proposition 4 implies that the Mono-Polychromatic Edge Coloring is NP-hard even when $k \geq \Delta(P)+1$, or when the underlying graph is bipartite: indeed, we can reduce a vertex coloring problem on a graph with large chromatic number to an instance of b-MPEC where $\Delta(P) = 2$. These facts contrast with classical results on edge coloring problems such as Kőnig's [3], or Vizing's [7].

Finally, we prove that the COSC is also NP-hard as it implies the b-MPEC.

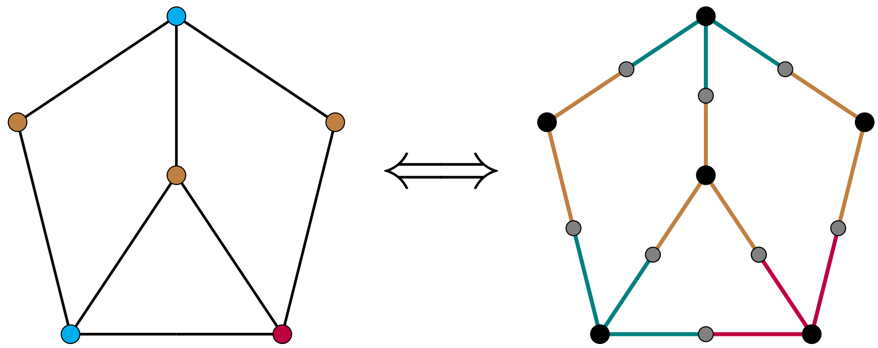

Fig. 2. Equivalence of vertex coloring problem (left-hand graph) and b-MPEC (right-hand graph) via edge subdivision. The gray vertices in the right-hand graph are polychromatic, the black vertices are monochromatic.

Proposition 5. *The COSC is NP-hard, even when there exists no edge between processors.*

Proof. It is enough to prove the result on the case where there exists no edge between processors. In particular, the underlying graph is bipartite. Let $B = (M, P; E)$ be a bipartite graph, and let $k \geq 0$ be an integer. If B is k-MP-colorable, then the corresponding coloring is also a solution of value exactly $|M|$ to the COSC instance defined on B and k. Since this is the minimum possible value of a solution to COSC, then that solution is optimal.

On the other hand, let x be an optimal solution to COSC on the graph B with at most k colors. If x has value exactly $|M|$, then B is k-MP-colorable, since the edges incident to each vertex $v \in M$ use exactly one color, and hence v is monochromatic. This proves that the problem of deciding whether MPEC admits a solution on $B = (M, P; E)$ of value less or equal than k is equivalent to the problem of deciding whether the COSC defined by $B = (M, P; E)$ and k admits a solution of value exactly $|M|$. Therefore, as MPEC is NP-hard, then also COSC is. □

To conclude we remark that the previous proof states that COSC is NP-hard also when there are no interactions between processors, hence we have this concluding result:

Corollary 1. *Minimizing the sum of the processing times of the jobs in the Concurrent Open Shop Problem with tasks of unitary duration, is NP-Hard.*

Fundings

This research was partially supported by the following projects :

- SERICS (PE00000014) under the NRRP MUR program funded by the EU - NGEU;
- OECP, funded by INS2I and CNRS.

References

1. Ahmadi, R., Bagchi, U., and Roemer, T. A. Coordinated scheduling of customer orders for quick response. *Naval Research Logistics 52*, 6 (2005), 493–512.
2. Ilani, H., Shufan, E., and Grinshpoun, T. Partially concurrent open shop scheduling and graph colourings. In *Proceedings of the 11th international conference of the practice and theory of automated timetabling (PATAT)* (2016), pp. 195–205.
3. Kőnig, D. Über graphen und ihre anwendung auf determinantentheorie und mengenlehre. *Mathematische Annalen 77*, 4 (1916), 453–465.
4. Kubiak, W. *A Book of Open Shop Scheduling: Algorithms, Complexity and Applications*, vol. 325 of *International Series in Operations Research & Management Science*. Springer International Publishing, 2022.
5. Misra, J., and Gries, D. A constructive proof of Vizing's theorem. *Information Processing Letters 41*, 3 (1992), 131–133.
6. Roemer, T. A. A note on the complexity of the concurrent open shop problem. *Journal of Scheduling 9*, 4 (2006), 389–396.
7. Vizing, V. G. On an estimate of the chromatic class of a p-graph. *Diskret. Analiz*, 3 (1964), 25–30.
8. Wagneur, E., and Sriskandarajah, C. Open shops with jobs overlap. *European Journal of Operational Research 71*, 3 (1993), 366–378.
9. Williamson, D. P., Hall, L. A., Hoogeveen, J. A., Hurkens, C. A. J., Lenstra, J. K., Sevast'janov, S. V., and Shmoys, D. B. Short shop schedules. *Operations Research 45*, 2 (1997), 288–294.