

Robotics and Chromatic Scheduling

Dominique de Werra

EPFL, 1015 Lausanne, Switzerland, dewerra.ima@epfl.ch

Marc Demange

ESSEC, F-95021 Cergy Pontoise, France, demange@essec.fr

Tinaz Ekim

EPFL, 1015 Lausanne, Switzerland, tinaz.ekim@epfl.ch

1 Introduction

A classical problem in robotics consists in organizing the moves of a single robot (or of several robots) which have to pick up a collection of items with different sizes along a storage line. A robot may pick up several items during a trip and it has to pile them, which implies that larger items have to be placed below smaller ones: in other words, during a trip a robot has to pick up items in decreasing order of their sizes.

Different hypotheses made on the possible moves of a robot along the storage line (or corridor) or assumptions on the location of the entry/exit points of the corridor will lead to different problems. It turns out that these will correspond to some graph coloring models; it is our objective to derive some of these generalized coloring models and to present some basic results on their complexity.

2 Preliminaries

For graph theoretical definitions not given here, the reader is referred to [1]. In this paper, all graphs are simple and loopless. Given a graph $G = (V, E)$ where $|V| = n$ and $|E| = m$, a subset of pairwise non-adjacent vertices is called a *stable set*, and a subset of pairwise adjacent vertices is called a *clique*.

A problem which is extensively studied in graph theory is *Min Coloring*: it is the problem of minimizing the number of stable sets covering all the vertices of G . Min Coloring has a wide range of applications in scheduling, telecommunication, production systems, etc. In the literature, several generalizations of this problem have also been studied to cover an even wider field of applications. One way of generalizing this usual coloring problem is to use not only stable sets but also cliques to cover all the vertices of a graph. In this paper, we use such generalizations in order to model some problems occurring in automated storage systems. We say that G is (p, k) -colorable if its vertex set V can be partitioned into p cliques and k stable sets. Given a graph G , *Min Cocoloring* consists in minimizing the number $p+k$ such that G is (p, k) -colorable. The minimum value is called *cochromatic number* and is denoted by $z(G)$. Another problem, called *Min Split-coloring*, is the one where for a given graph G , we minimize the number $\max(p, k)$ such that G is (p, k) -colorable. The optimal value, called *split-chromatic number*, is denoted by $\chi_S(G)$. The reason why we call this problem split-coloring is that it amounts to partitioning the vertex set of a given graph into a minimum number of split graphs; a graph is a *split graph* if its vertex set admits a partition (S, K) , called *split partition*, where S is a stable set and K a clique. Min Split-coloring was first introduced in [8] and studied in [5, 6, 4] which give polynomial time algorithms in restricted classes of graphs and provide approximation algorithms for other cases.

A graph G is a *threshold graph* if it is a split graph with partition (S, K) and the neighborhoods of vertices i in S , denoted by $N(i)$, are nested, i.e., one can label vertices in S with integers such that if $i < j$ then $N(i) \subseteq N(j)$.

In this work, we handle the above generalized coloring problems in permutation graphs; given a permutation π of n numbers, the corresponding *permutation graph* $G(\pi)$ is obtained by representing each number by a vertex and linking vertices i and j with an edge whenever we have $i > j$ and $\pi^{-1}(i) < \pi^{-1}(j)$, $\pi^{-1}(i)$ being the position of number i in π . Note that, in permutation graphs, stable sets correspond to increasing sequences and cliques to decreasing sequences. Partitioning permutations into increasing and decreasing sequences has been extensively studied [14, 12, 2]. Our work will extend this research and raise some related questions. Let us first mention the preliminary results on the complexity of Min Split-coloring.

Theorem 2.1 ([4]) *Let \mathcal{G} be a class of graphs closed under addition of cliques without link to the rest of the graph and under addition of stable sets completely linked to the rest of the graph; then for any graph in \mathcal{G} Min Cocoloring reduces to Min Split-coloring.*

Corollary 2.2 *Min Split-coloring is NP-hard in permutation graphs.*

The corollary follows from Theorem 2.1 and the fact that Min Cocoloring is NP-hard in permutation graphs [14].

3 Various models of pick up robots in an automated storage system

We shall consider several situations where a robot has to pick up a collection of items with different sizes which are located in a storage shelf. One may see the storage shelf as a predefined trajectory that the robot can cross in both directions. For simplicity purposes we will assume that the shelf has just one level so that the items (characterized by their size) are located in line along a storage corridor.

In addition, we assume that the items are labelled: $1, 2, \dots, n$ in the order of their decreasing sizes. So their positions along the corridor are given by a permutation π of $\{1, 2, \dots, n\}$. The robot is able to pick up several items but these should be arranged in a pile. For obvious stability reasons when a pile is constructed one is never allowed to put an item j on top of an item i if $j < i$, i.e., if j has a larger size than i . So the robot picks up items in increasing order of their labels. Note that, in a more general framework, the labels may correspond to any order, due to some precedence constraints, in which some tasks should be executed by the robot. In this paper, all the models are illustrated for the case where the order is determined by the requirement of decreasing sizes of the items to be collected.

We shall describe six situations which will lead to different graph theoretical models. The storage corridor will have a left end called L and a right end called R . In all of these models, the unloading operations performed at an Entry/Exit station E cause some idle time for the robots. For this reason, we want to minimize the number of times the robot has to be unloaded.

In each model, continuous arrows represent the possible exit trajectories from the Entry/Exit station E , whereas dashed arrows represent the possible entrance trajectories to E . We also show the possible moves of the robot along the corridor by (broken) arrows depicted under the corridor.

3.1 One-directional system

Here there is an Entry/Exit station E which is outside of the corridor; all robot trips are from L to R then to E and back to L for a new trip.

If we call $G(\pi)$ or simply G the permutation graph associated to π , then during a trip the robot will be able to collect some items in increasing order of their labels; the vertices corresponding to these items will form a stable set in G . So, minimizing the number of trips made by a robot for collecting all items amounts to finding a usual coloring of a permutation graph G . In other words one has to cover π with a minimum number of increasing sequences.

3.2 Bi-directional system with a unique E

The robot is now allowed to have trips from L to R and then to E as well as trips from R to L and then to E . In other words, whenever a robot is at E , it may go to L or to R to start its trip along the storage corridor.

Each trip of the robot will be represented by an increasing or a decreasing sequence in π . Minimizing the number of trips required to pick up all items corresponds to finding an optimum cocoloring of G , i.e., a partition into a minimum number of stable sets and cliques: each trip corresponds in π to a monotone sequence, that is either an increasing sequence (trip from L to R) or a decreasing sequence (trip from R to L), i.e., to a stable set or to a clique.

3.3 Bi-directional system with E at both ends

In this case we have at each end L and R an Entry/Exit station E . The robot may start from one end, pick up items while moving to the other end, then unload its pile and possibly start a trip back to the end where it entered the corridor. The items collected during each trip from L to R will correspond to a stable set in G (increasing sequence) and for each trip from R to L the items collected will define a clique. So all the items collected during a trip from L to R and then from R to L (or the inverse) form a split graph. We may assume that after the last trip, the robot may possibly make a non-collecting trip to get back to the station where it started. Hence minimizing the number of two-way trips of the robot will be a split-coloring problem; it will consist of partitioning the vertex set of G into a minimum number of split graphs.

3.4 Two-way trips from L

We shall assume now that at end L of the storage corridor there is an Entry/Exit station. The robot may now enter the corridor at L , then it will move to the other end R and come back where it started so that it can unload the collected items.

In the permutation π , each trip (starting from L) will be represented by an increasing sequence $a_1 < a_2 < \dots < a_p$ (trip from L to R) followed by an increasing sequence $a_{p+1} < \dots < a_r$ with $a_{p+1} > a_p$ (since this last part corresponds to the trip from R to L , it will be in fact decreasing in π and it will be represented by a clique in G). The items collected in such a two-way trip will form a threshold graph in G ; this can be seen as follows: if j is collected after i during the trip back to L , then j comes before i in π and $j > i$, so j will be linked to all neighbors of i which are after i in π . Hence $N(j) \supseteq N(i)$.

So in this case, if one can pick up all items with l two-way trips all from L , then there are l threshold graphs which cover all vertices of G . The links of this problem with graphs will be discussed in more details in Section 5. Note that a symmetrical model is obtained if the Entry/Exit station is located at R instead of L .

3.5 Zigzag trips

It is reasonable to extend the above type of models by assuming that the robot may move back and forth several times in the corridor. This can be allowed, for instance, by the fact that the robot receives some fixed energy impulse at each visit of the Entry/Exit station which is sufficient to execute such a multiple trip. If the robot makes l trips (alternating trips from L to R and from R to L) before leaving the corridor, the items collected will be represented by a so-called $(l-1)$ -modal sequence; it can be seen as a trip where the robot changes direction $l-1$ times.

3.6 Two-way trips from either end

Here, we focus on a special case of model 3.5. The robot collects items with two-way trips (meaning that the items are unloaded only at the end of a back and forth trip) from either L or R . The possibility of doing two-way trips is provided by the energy impulse given at each visit of the Entry/Exit station. The location of the Entry/Exit station allows the robot to execute two-way trips that can start from either end; such a trip in this model is called *feasible trip*. It is easy to see that a feasible trip is obtained by merging a sequence $a_1 < a_2 < \dots < a_s$ and $b_1 > b_2 > \dots > b_t$ with either $a_s < b_t$ (robot starting from L), or $b_1 < a_1$ (robot starting from R). It follows from the explanations given in model 3.4 that the items collected during each feasible trip in this model induces a threshold graph in the corresponding permutation graph G . The problem of covering a given permutation π (or equivalently collecting all the items represented by π) by a minimum number of feasible trips will be called *Min Ordered Collecting*, and its optimal value will be denoted by $\rho(\pi)$. Also, if we denote by $\rho_l(\pi)$ (respectively $\rho_r(\pi)$) the values of optimal ordered collections where all feasible trips start from left (respectively right) as described in model 3.4, then clearly $\rho(\pi) \leq \min(\rho_l(\pi), \rho_r(\pi))$. The links of Min Ordered Collecting with graphs will be discussed in Section 5.

4 Some results on related problems

We first introduce the necessary definitions and tools to study the models described above. Then, polynomial time algorithms will be derived for some special cases.

4.1 Terminology and NP-hardness

Before mentioning the complexity status of Min Ordered Collecting, let us define the notion of l -modal sequences, introduced in [13].

We call *internal extremum* of a permutation π , a number $\pi(i)$ such that $2 \leq i \leq n-1$ and we have either $\pi(i) < \pi(i-1)$ and $\pi(i) < \pi(i+1)$, or $\pi(i) > \pi(i-1)$ and $\pi(i) > \pi(i+1)$. A sequence is *l-modal* if it has at most l internal extrema, the first being of either type. If the first extremum is a maximum then the sequence is called *upper l-modal* and otherwise *lower l-modal*. In the case $l=1$, we say that a sequence is (upper or lower) *unimodal*. Finally, if a permutation can be partitioned into p decreasing and k increasing subsequences then it is called *p-decreasing k-increasing*.

Let $\pi = (\pi(1), \dots, \pi(n))$ be a permutation, the *reversal* of π , denoted by $\bar{\pi}$, is the permutation $(\pi(n), \dots, \pi(1))$. The *inverse permutation*, denoted by π^{-1} , is given by $(\pi^{-1}(1), \dots, \pi^{-1}(n))$. Finally, we call the *symmetric* of π , the permutation $\pi_s = (n+1-\pi(1), \dots, n+1-\pi(n))$. Clearly, $\rho_l(\pi) = \rho_r(\bar{\pi})$ and $\rho_r(\pi) = \rho_l(\bar{\pi})$. One can notice that ρ is an invariant with respect to the symmetric and the reversal operations, i.e., $\rho(\pi) = \rho(\bar{\pi}) = \rho(\pi_s)$, since both interchange increasing and decreasing subsequences. Note also that $\rho_l(\pi) = 1$ if and only if π^{-1} is upper unimodal, and $\rho_r(\pi) = 1$ if and only if π^{-1} is lower unimodal; moreover $(\pi^{-1})^{-1} = \pi$. Finally, it can be easily

checked that π is upper l -modal if and only if π_s is lower l -modal and vice versa. To illustrate, consider the permutation $\pi = (1, 5, 2, 4, 3)$ which is a feasible trip from left; then $\pi^{-1} = (1, 3, 5, 4, 2)$ is upper unimodal and $(\pi^{-1})_s = (5, 3, 1, 2, 4)$ is lower unimodal.

Using the above terminology, one can state that minimizing feasible trips covering a given permutation π is equivalent to finding the minimum k such that there are at most k unimodal subsequences covering π^{-1} .

Now, we assume that for reasons of autonomy the robot is constrained to make at most $l + 1$ trips back and forth along the corridor before going to the Entry/Exit station, as introduced in model 3.5. It is therefore reasonable to try to minimize the number of times the robot has to make a trajectory with (at most) $l + 1$ trips back and forth. This amounts to covering the permutation π representing the items in the corridor by a minimum number of l -modal sequences.

Let us call *Min l -modal*, *Min upper l -modal* and *Min unimodal* the problems of covering a given permutation with a minimum number of respectively l -modal, upper l -modal and unimodal subsequences, then the following theorem of [13] establishes the *NP*-hardness of all these problems.

Theorem 4.1 ([13]) *Min l -modal, Min upper l -modal and Min unimodal are NP-hard even for fixed l , in particular for $l = 1$. \square*

Corollary 4.2 *Given a permutation π , it is NP-hard to find $\rho(\pi)$, $\rho_l(\pi)$ and $\rho_r(\pi)$.*

Proof 4.1 *Given a permutation π , feasible trips from left (respectively right) in π become upper (respectively lower) unimodal sequences in π^{-1} ; moreover the inverse permutation is obtained in polynomial time. \square*

In what follows, we formulate our results in terms of Min (upper/lower) l -modal problem rather than feasible trips. A sequence that can be covered by k (upper/lower) l -modal subsequences will be called k -(upper/lower) l -modal and similar notations hold for the unimodal case.

4.2 Recognition of 2-lower unimodal permutations

This would correspond to the case where all items could be picked up by two trips of the robot starting from R and making two-way trips (See model 3.4).

The following result simply states that there is a polynomial time algorithm to decide whether the items in the corridor can be picked up by two trips of a robot starting from R and making two-way trips.

Theorem 4.3 *There is an algorithm that decides in time $O(m + n \log n)$ whether a given permutation π is 2-lower unimodal or not.*

By symmetry, if the robot starts from L , we can clearly state:

Corollary 4.4 *Given a permutation π , it can be decided in time $O(m + n \log n)$ whether it is 2-upper unimodal or not. \square*

4.3 Maximum l -modal subsequence

Consider a permutation π which is upper or lower l -modal. Then, it can be easily checked that π^{-1} corresponds to an increasing subsequence of labels representing items collected during $l + 1$ trips back and forth of a robot (starting from left and right, respectively) which does not unload the collected items until the end of $l + 1$ trips and which obeys the constraint of increasing labels during its whole trip (see model 3.5). In other words, a robot can collect all the items ordered

according to π^{-1} with $l + 1$ back and forth trips respecting the constraint of increasing labels. In what follows, given a permutation π representing the sizes of the items, we show how to collect a maximum number of items by $l + 1$ trips of a robot. This will be explained in terms of maximum l -modal subsequence of a given permutation.

We construct from a permutation π a permutation π^l such that there is a one-to-one correspondence between the upper l -modal subsequences in π and the increasing subsequences in π^l . Since a maximum increasing subsequence in π^l can be found in time $O((l + 1)n \log((l + 1)n))$ we get:

Theorem 4.5 *Given a permutation π , a maximum upper or lower l -modal subsequence of π can be found in time $O(n \log n)$ for fixed l , and in time $O(n^2 \log n)$ for arbitrary l .*

5 Min Ordered Collecting vs. Min Threshold-coloring

In this section, we explore the links between feasible trips (or unimodal subsequences) and split graphs. We recall that feasible trips correspond in fact to a subclass of split graphs, namely the threshold graphs. Nevertheless, as will be explained below, a minimum ordered collecting ρ is a parameter defined in permutations and not in permutation graphs. These results give birth to the idea of considering the problem of covering the vertices of a given graph with a minimum number of threshold graphs.

The following notion is given in [11].

Definition 5.1 (shuffle product) *Let σ and τ be two sequences. The shuffle product of σ and τ is defined as follows:*

$$\sigma \sqcup \tau = \{\sigma_1 \tau_1 \dots \sigma_k \tau_k : \sigma = \sigma_1 \dots \sigma_k \text{ and } \tau = \tau_1 \dots \tau_k\}$$

where σ_i and τ_i are subsequences, k ranges over all integers and the juxtaposition means concatenation.

Now, let us observe the following fact.

Proposition 5.1 *Let π be a permutation, then:*

$$\begin{aligned} \rho_l(\pi) = 1 &\Leftrightarrow \pi \text{ is a shuffle product of type 1: } [1, \dots, p] \sqcup [n, \dots, p + 1], 1 \leq p < n; \\ \rho_r(\pi) = 1 &\Leftrightarrow \pi \text{ is a shuffle product of type 2: } [p + 1, \dots, n] \sqcup [p, \dots, 1], 1 \leq p < n; \end{aligned}$$

where shuffle products can also have only one term.

The relationship between threshold graphs and feasible trips is derived from the following theorem of [10].

Theorem 5.2 ([10]) *The threshold graphs are precisely those permutation graphs corresponding to permutations contained in*

$$[1, 2, \dots, p] \sqcup [n, n - 1, \dots, p + 1]$$

where p and n are positive integers. \square

A natural extension of Min Ordered Collecting in graphs appears as the following problem. These extensions may have no obvious interpretation in terms of item collection by robots along a storage corridor; we give them anyway for their mathematical interest.

Definition 5.3 (Min Threshold-coloring) *Min Threshold-coloring is the problem of covering the vertices of a given graph G by a minimum number of threshold graphs. The optimal value, called threshold-chromatic number, is denoted by $\chi_T(G)$.*

Theorem 5.4 *Let \mathcal{G} be a class of graphs closed under addition of cliques without link to the rest of the graph and under addition of stable sets completely linked to the rest of the graph, then for any graph in \mathcal{G} Min Cocoloring reduces in polynomial time to Min Threshold-coloring.*

Corollary 5.5 *Min Threshold-coloring is NP-hard in permutation graphs.*

6 Open questions and discussion

There are several questions to explore on the links between Min Ordered Collecting and Min Threshold-coloring. In view of Theorem 5.2 and the following discussion, we have clearly $\rho(\pi) \geq \chi_T(G(\pi))$ for any permutation π . It would be interesting to know, given a permutation graph G , whether there is a permutation π such that $G(\pi)$ is isomorphic to G and $\rho(\pi) = \chi_T(G)$. If this is the case, then an optimal threshold coloring of G would give an optimal ordered collecting of the items forming the corresponding permutation π . Also, one could study, for a permutation graph G , how large can be the quantity $\max_{\pi_i}(\rho(\pi_i) - \chi_T(G))$, where $G(\pi_i)$ is isomorphic to G .

Another interesting topic concerns permutations for which an optimal ordered collecting is a solution where the robot departs always from the same end. In other words, for such permutations, locating the Entry/Exit station outside of the corridor does not provide any advantage over the more constrained model where the Entry/Exit station is fixed to one end of the corridor. This notion yields open questions of high mathematical interest.

Definition 6.1 (ideal permutation) *A permutation π is called ideal if we have $\rho(\pi) = \min(\rho_l(\pi), \rho_r(\pi))$.*

Definition 6.2 (minimal obstruction) *A permutation π is a minimal obstruction if it verifies $\rho(\pi) < \min(\rho_l(\pi), \rho_r(\pi))$ and all subpermutations $\pi' \subset \pi$ are ideal, i.e., $\rho(\pi') = \min(\rho_l(\pi'), \rho_r(\pi'))$.*

In [9], lower and upper bounds on the size of minimal obstructions for permutations having $\rho(\pi) = 2$ is studied. It can be easily shown by case enumeration that all permutations of size less than or equal to 7 are ideal; smallest minimal obstructions that are detected have size 8. For instance, $\pi = (1, 8, 4, 3, 6, 2, 7, 5)$ is a minimal obstruction; $\rho(\pi) = 2$ where the robot collects the items with labels (1, 5, 8) by starting from left, and the items with labels (4, 3, 6, 2, 7) by starting from right. However, it can be easily checked that $\rho_l(\pi) = \rho_r(\pi) = 3$ and that for all subpermutations $\pi' \subset \pi$, we have $\rho(\pi') = \min(\rho_l(\pi'), \rho_r(\pi'))$. On the other hand, it is also shown by computer enumeration that obstructions of size 9 always contain an obstruction of size 8, that is, there is no minimal obstruction of size 9. These results allow us to state the following conjecture.

Conjecture 6.3 *All minimal obstructions π with $\rho(\pi) = 2$ are of size 8.*

A natural extension of the robotics problems considered in this paper is the case where the arrangement of the items on the storage corridor is not known in advance, but it is revealed in time. The study of such on-line models would be of great use in practice; in fact it is natural to think that items are delivered at different times depending on the rest of the production system, and that one has to decide the moves of the robot each time an item is delivered to the storage system. We can already find in [13] and [7] some results on the on-line version of the model 3.2; further research is needed on other models.

References

- [1] C. Berge (1976), *Graphs and Hypergraphs*. North Holland Publishing Company.
- [2] A. Brandstädt and D. Kratsch (1986), On partitions of permutations into increasing and decreasing subsequences. *Journal of Information Processing and Cybernetics* **22**(5/6), 263 – 273.
- [3] V. Chvátal (1984), Perfectly ordered graphs. In *Topics on Perfect Graphs (C. Berge and V. Chvátal eds.)* **21**, *Annals of Discrete Mathematics*, 63 – 65.
- [4] M. Demange, T. Ekim, and D. de Werra (in press), On the approximation of Min Split-coloring and Min Cocoloring. *Journal of Graph Algorithms and Applications*.
- [5] M. Demange, T. Ekim, and D. de Werra (2005), Partitioning cographs into cliques and stable sets. *Discrete Optimization* **2**, 145 – 153.
- [6] M. Demange, T. Ekim, and D. de Werra (2005), (p, k) -coloring problems in line graphs. *Theoretical Computer Science* **349**(3), 462 – 474.
- [7] M. Demange and B. Leroy-Beaulieu (2006), Online Coloring of Comparability Graphs: some results. <http://infoscience.epfl.ch/search.py?recid=102343>, *Ecole Polytechnique Fédérale de Lausanne (ROSE)*.
- [8] T. Ekim and D. de Werra (2005), On split-coloring problems, *Journal of Combinatorial Optimization* **10**, 211 – 225.
- [9] F. Gilliéron (2006), Collecte d'objets par ordre décroissant de leur taille, Semester project, *Ecole Polytechnique Fédérale de Lausanne (ROSE)*.
- [10] M. C. Golumbic (1978), Threshold graphs and synchronizing parallel processes, **18** of *Combinatorics*, Colloq. Math. Soc. Janos Bolyai, North Holland, Budapest, 419 – 428.
- [11] M. C. Golumbic (1980), *Algorithmic graph theory and perfect graphs*, Computer Science and Applied Mathematics, Academic Press.
- [12] A.E. Kézdy, H.S. Snevily, and C. Wang(1996), Partitioning permutations into increasing and decreasing subsequences, *Journal of Combinatorial Theory Series A* **73**(2), 353 – 359.
- [13] G. Di Stefano, S. Krause, M.E. Lübbecke, and U.T. Zimmermann (2006), On minimum k -modal partitions of permutations, In *LATIN 2006: Theoretical informatics* **3887** of *Lecture Notes in Comput. Sci.* Springer, Berlin, 374 – 385.
- [14] K. Wagner (1984), Monotonic coverings of finite sets, *Elektron. Inf. Kybern. EIK* **20**(12), 633 – 694.