Published in European Journal of Operational Research 240(2) (2015) 307-314
doi.org/10.1016/j.ejor.2014.05.011

On some applications of the selective graph coloring problem

Marc Demange[a,d], Tınaz Ekim[b,*], Bernard Ries[c,d], Cerasela Tanasescu[a,e]

[a] ESSEC Business School, Paris, France
[b] Bogazici University, Istanbul, Turkey
[c] PSL, Université Paris-Dauphine, 75775 Paris Cedex 16, France
[d] CNRS, LAMSADE UMR 7243, France
[e] CEREGMIA, University of Antilles-Guyane, Schoelcher, Martinique, France

ABSTRACT

In this paper we present the *Selective Graph Coloring Problem*, a generalization of the standard graph coloring problem as well as several of its possible applications. Given a graph with a partition of its vertex set into several *clusters*, we want to select one vertex per cluster such that the chromatic number of the subgraph induced by the selected vertices is minimum. This problem appeared in the literature under different names for specific models and its complexity has recently been studied for different classes of graphs. Here, we describe different models – some already discussed in previous papers and some new ones – in very different contexts under a unified framework based on this graph problem. We point out similarities between these models, offering a new approach to solve them, and show some generic situations where the selective graph coloring problem may be used. We focus on specific graph classes motivated by each model, and we briefly discuss the complexity of the selective graph coloring problem in each one of these graph classes and point out interesting future research directions.

1. Introduction

1.1. Definitions and motivation

Graph coloring is one of the most studied optimization problems in graph theory. Given a simple graph, it consists in assigning one color to each vertex such that two vertices linked by an edge get different colors and the total number of colors used is minimized. It is widely used to model several types of real applications such as scheduling, timetabling, frequency allocation, wavelength routing and many more (see for instance Dandashi & Al-Mouhamed, 2010; Gamache, Hertz, & Ouellet, 2007; Giaro, Kubale, & Obszarski, 2009; Qu, Burke, & McCollum, 2009; Talaván & Yáñez, 2008). Several generalizations of the classical graph coloring problem are also considered in the literature to cover an even wider range of applications with various constraints (see for instance Demange, Ekim, & de Werra, 2009; Sotskov, Dolgui, & Werner, 2001). In this paper, we motivate a further generalization of the usual graph coloring problem which introduces more flexibility to the applications by offering the possibility of choosing among several predefined strategies.

In this paper, all graphs $G = (V, E)$ are undirected and simple. A *stable set* is a set of vertices that are pairwise non-adjacent and a *clique* is a set of vertices that are pairwise adjacent. We denote by $\alpha(G)$ and $\omega(G)$ the size of a maximum stable set and the size of a maximum clique in G, respectively. The complementary graph $\bar{G} = (V, \bar{E})$ of $G = (V, E)$ is defined by $uv \in \bar{E} \iff u \neq v$ and $uv \notin E$. An induced path on k vertices is denoted by P_k. The graph obtained by taking k disjoint copies of G (with no edges between any two copies) is referred to as kG. For $V' \subseteq V, G[V']$ denotes the subgraph induced by V' in G.

A *k-coloring* of G is a mapping $c: V \to \{1, \ldots, k\}$ ($c(u)$ is called the *color* of vertex u) such that $c(u) \neq c(v)$ for all $uv \in E$ (each color class defined by the set of vertices with a same color forms a stable set). The smallest integer k such that G admits a k-coloring is called the *chromatic number* of G and is denoted by $\chi(G)$. Given a graph G, the problem of deciding whether G is k-colorable or not is called k-Colorability. Consider now a partition $\mathcal{V} = (V_1, V_2, \ldots, V_p)$ of the vertex set V of G. The sets $V_1, \ldots, V_p$ are called *clusters* and $\mathcal{V}$ is called a *clustering*. A *selection* is a subset of vertices $V' \subseteq V$ such that $|V' \cap V_i| = 1$ for all $i \in \{1, \ldots, p\}$. A *selective k-coloring* of G, also called *partition coloring* in the literature, with respect to $\mathcal{V}$ is defined by (V', c) where V' is a selection and c is a k-coloring of $G[V']$. We may define the following two problems:

SEL-COL
Input: An undirected graph $G = (V, E)$ and a clustering $\mathcal{V} = (V_1, \ldots, V_p)$ of V.
Output: A selection V^* such that $\chi(G[V^*])$ is minimum.

Let $k \geqslant 1$ be a fixed integer.

k-DSEL-COL
Input: An undirected graph $G = (V, E)$ and a clustering $\mathcal{V} = (V_1, \ldots, V_p)$ of V.
Question: Does there exist a selection V' such that $\chi(G[V']) \leqslant k$?

The smallest integer k for which a graph G admits a selective k-coloring with respect to $\mathcal{V}$ is called the *selective chromatic number* of G and is denoted by $\chi_{SEL}(G, \mathcal{V})$. It is obvious to see that $\chi_{SEL}(G, \mathcal{V}) \leqslant \chi(G)$ for every clustering $\mathcal{V}$ of V. For both problems above, adding or/and deleting edges between vertices of a same cluster does not affect the solution. However, when considering specific classes of graphs, adding or deleting edges in a cluster may not be allowed if the graph class we are considering is not stable under such graph operations. In this case, it is sometimes interesting to consider instances for which all clusters are cliques. We call this case *compact clustering* (Demange, Monnot, Petrica, & Ries, 2013).

The selective graph coloring problem has been considered by several authors. Some complexity results for restrictive classes of graphs can be found in (Demange et al., 2013; Erlebach & Jansen, 2001). In (Guangzhi & Simha, 2000), some constructive heuristics are proposed whereas a tabu search algorithm for SEL-COL is developed in (Noronha & Ribeiro, 2006). Concerning exact algorithms, a branch-and-cut algorithm and a branch-and-price algorithm are given along with computational results in respectively (Frota, Maculan, Noronha, & Ribeiro, 2010; Hoshino, Frota, & de Souza, 2011). More graph theoretical results concerning the selective graph coloring problem can be found in (Bonomo, Cornaz, Ekim, & Ries, 2013).

To the best of our knowledge (Guangzhi & Simha, 2000; Noronha & Ribeiro, 2006) are the only papers where the selective graph coloring problem is explicitly mentioned to model a real life problem which is called routing and wavelength assignment. However, we notice that the selective graph coloring problem has a huge potential to extend the use of the standard coloring problem to many other real life applications. The aim of this paper is to emphasize how several real life problems can be modeled using the selective graph coloring model whereas the usual graph coloring model would not be able to handle them. Section 2 presents examples such as routing and wavelength assignment, dichotomy-based constraint encoding, frequency assignment, timetabling, quality test scheduling, berth allocation and multiple stuck TSP. Each model motivates the study of SEL-COL in some specific class of graphs and rises in particular questions about complexity. These questions are studied in (Demange et al., 2014) and in this paper we will only mention the results without the proofs. We first start with a few remarks about the complexity of the selective graph coloring problem.

1.2. Some remarks on the complexity of the selective graph coloring problem

As mentioned above, a solution for SEL-COL consists in a selection V^* such that $\chi(G[V^*])$ is minimum. However, the value associated with V^*, namely $\chi(G[V^*])$, may be hard to determine even if V^* is known. In other words, when we can compute efficiently an optimal solution for SEL-COL for an instance $(G, \mathcal{V})$, this does not necessarily imply that its value $\chi_{SEL}(G, \mathcal{V})$ can be computed in polynomial time. More precisely, it is still NP-hard to compute the selective chromatic number in general even when an optimal selection V^* is known. For this reason, we will sometimes distinguish between the hardness of SEL-COL (selection process) and the hardness of deciding whether a given selection induces a k-colorable graph.

The following examples of reductions point out this difference. First note that, for any $k \geqslant 1$, k-DSEL-COL is in NP in general graphs. Indeed, given an instance $(G, \mathcal{V})$ with a set $V' \subseteq V$ and a coloring of $G[V']$, in order to test whether $\chi(G[V'])) \leqslant k$, it is enough to verify whether V' meets each cluster exactly once and the color assignment defines a k-coloring of the graph $G[V']$. This can clearly be done in polynomial time. Furthermore, 3-DSEL-COL generalizes the usual 3-coloring problem (i.e, given a graph $G = (V, E)$, we want to know if G admits a coloring of V using at most 3 colors) for which each vertex is a cluster on its own, and hence it is NP-complete. However determining an optimal selection is easy since there is only one solution: all vertices must be selected since each vertex represents a cluster.

Showing the NP-hardness of SEL-COL requires another reduction. Consider an instance $G = (V, E)$ of 3-COLORABILITY. Let $G' = (V', E')$ be a graph composed of 3 independent copies of G and define the following clustering $\mathcal{V}$ of V': each cluster contains exactly the 3 copies of a same vertex in G. Then, G is 3-colorable if and only if an (optimal) selection for the instance $(G', \mathcal{V})$ is a stable set (i.e., $\chi_{SEL}(G', \mathcal{V}) = 1$). Indeed, if there is a selection which is a stable set, then a 3-coloring of G can be found by coloring a vertex with color $i \in \{1, 2, 3\}$ if its ith copy in G' has been selected. Conversely, a 3-coloring of G allows to find a selection which is a stable set by choosing the ith copy of a vertex in G' if it is colored with color $i \in \{1, 2, 3\}$. Note that deciding whether a given selection is 1-colorable is trivially polynomial.

The two reductions above show that 3-DSEL-COL is NP-complete even with clusters of size 1 while 1-DSEL-COL (resp. SEL-COL) is NP-complete (resp. NP-hard) even if clusters are of size 3. However, in the first case the selection is trivial but evaluating the corresponding selective chromatic number is hard while in the second case finding the required selection is hard but it is trivial to decide whether the chromatic number of the graph induced by any selection is 1. *More generally, any hardness result for k-DSEL-COL, with $k \leqslant 2$, or for SEL-COL in graph classes for which graph coloring is polynomial, points out the hardness of the selection process.* Note that in many cases both steps – finding a selection and evaluating its value – can be hard.

Finally, note that the last reduction above can be adapted to show that SEL-COL even generalizes the k-List-coloring problem. Here, with every vertex v of the instance graph $G = (V, E)$, we associate a list $L(v) \subseteq \{1, \ldots, k\}$ of allowed colors and the problem consists in deciding whether there is a k-coloring of G such that each vertex gets a color from its list. We construct a graph $G' = (V', E')$ as follows. For every vertex $v \in V$ and every color $c \in L(v)$ we define a vertex (v, c) of G'. We then link (v, c) and (v', c') if $c = c'$ and v, v' are linked in G. The clustering $\mathcal{V}$ of V' is obtained by putting all vertices $(v, c), c \in L(v)$ in a same cluster. Clearly, G admits a k-list coloring if and only if $\chi_{SEL}(G', \mathcal{V}) = 1$.

2. Some models and related classes of graphs

In this section, we present five different types of applications each of which motivates SEL-COL in a particular class of graphs. For each of these classes, we mention the related computational complexity for solving the problem.

2.1. Routing and wavelength assignment

Optical networks appear in a large number of applications including among others high performance computing and video-conferences. In the technology of optical networks, the information is transmitted along the optical fibers as optical signals. An all-optical point-to-point connection between two end nodes is called a *light-path*. A light-path request consists of two nodes, the input and the output nodes. A related routing specifies one light-path between these two nodes. Two light-paths may use the same wavelength, provided they do not share any common link.

In the Routing and Wavelength Assignment (RWA) problem (see e.g. Zang, Jue, & Mukherjee, 2000), sometimes refereed to as *Path Coloring Problem* (see e.g. Erlebach & Jansen, 2001), light-path requests are given and the two required operations are (1) to generate a set of light-paths for each request (called *routing*) and (2) to select a light-path per request and assign wavelengths to these selected paths (called *wavelength assignment*). The set of light-paths should be generated and the selection of one light-path per request should be made among the generated light-paths such that the number of wavelengths needed is minimized.

An optical network can be transformed into a host graph by associating vertices, edges and paths in the host graph with respectively nodes, links and light-paths in the optical network. For some host graphs the number of possible paths between two vertices is limited by the size of the network and then all of them can be considered before solving the wavelength assignment problem. Nevertheless, some real applications of RWA involve relatively dense graphs for which the number of possible paths satisfying a given request may be huge, sometimes even exponential with respect to the size of the network. Then, the problem is usually decomposed into two parts which are sequentially solved (see for instance Noronha & Ribeiro, 2006): first a set of light-paths is generated for each request, then taking this set as input for the second part, one light-path per request is selected in such a way that the number of wavelengths needed for the selected light-paths is minimized. In what follows, we will only focus on the second part, that is the wavelength assignment problem.

From the host graph associated with the optical network we construct another graph as follows. Its vertex set corresponds to the preselected set of paths on the host graph (which in turn correspond to possible light-paths between inputs and outputs associated with requests) and there is an edge between two vertices if the two corresponding paths share at least one common edge. Then we associate a cluster with each request, i.e., a cluster contains all vertices corresponding to preselected light-paths satisfying a same request. A selection defines a specific path for each request and the number of necessary wavelengths corresponds to the chromatic number of the graph induced by the vertices of this selection. This model was originally described in (Guangzhi & Simha, 2000) (see also Noronha & Ribeiro, 2006) under the name of *Partition Coloring Problem*. In Fig. 1, one can observe on the left side, a host graph with light-paths 1, 2, 3 between nodes a and b, 4, 5 between nodes a and c, and 6, 7 between d and e. The related selective coloring model (clustered graph) where each cluster corresponds to a light-path request is on the right side. A best solution consists for example in selecting light-paths 1, 4, and 6. In this case one wavelength (color) is sufficient.

The above application motivates the selective graph coloring problem in the so-called edge intersection graphs of paths in different kinds of host graphs. Natural classes of host graphs for this problem are trees, rings, tori, Meshes, hypercubes or stars (Beauquier et al., 1997). Related complexity and approximation results can be found in (Erlebach & Jansen, 2001). In particular the problem is shown to be NP-hard in star networks and in rings (cyclic networks). Note that in the case of a star network and more generally in a tree network, there is only one single path per request. In the related Sel-Col instance, each cluster contains exactly one vertex and thus, the problem is a usual coloring problem without any selection process. In the case where the host graph is a cycle (ring), if only one path per request is considered, then the related Sel-Col problem reduces to a coloring problem in a *circular arc graph* (CAG), defined as the intersection graph of arcs on a circle. This problem is known to be NP-hard (Garey, Johnson, Miller, & Papadimitriou, 1980). The result in (Erlebach & Jansen, 2001) concerning rings corresponds to the case where the two possible paths satisfying a request are considered. The problem corresponds then to the selective graph coloring problem in circular arc graphs with clusters containing two vertices corresponding to two arcs the union of which is the whole cycle.

One can note that edge intersection graphs of trees in a graph also deserve investigation since they can model the multicast routing and wavelength assignment problem in which requests are given between possibly more than two nodes and therefore trees linking all end nodes related to a request should be considered instead of paths (Zhou & Poo, 2005).

Finally, notice that whenever the considered class of host graphs is stable under the addition of a pending vertex (i.e., a vertex of degree one), then the general case is reduced to the particular case where all clusters are cliques (called compact clustering). Indeed, if for each request between two vertices x and y we add a pendant vertex z to one of the vertices, say y, and extend every path between x and y to a path between x and z by adding the edge yz, then all vertices belonging to a same cluster will be pairwise adjacent.

2.2. Dichotomy-based constraint encoding

Originally introduced by Tracey 50 years ago (Tracey, 1966), dichotomy-based constrained encoding can be used to solve several problems in computer science and in particular to generate asynchronous implementations of Finite State Machines.

Let $S = \{s_1, \ldots, s_n\}$ be a set of n states. An *encoding* of length k is a mapping from S into $\{0, 1\}^k$. For $j = 1, \ldots, k$, the jth component of the vector associated with a state $s \in S$ in an encoding will simply be referred to as jth component of state s. A *dichotomy* in S is an

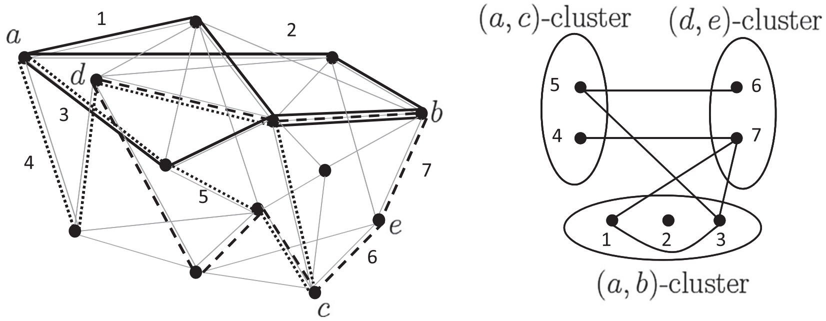

Fig. 1. An example of routing and wavelength assignment problem and its selective graph coloring model.

unordered pair $\{P, Q\}$ of disjoint subsets of S, with $P \cup Q \neq \emptyset$. We say that an encoding of S *satisfies* a dichotomy $\{P, Q\}$ if there is at least one component that takes value 0 for all states in P and value 1 for all states in Q, or vice versa. Let $C = \{D_1, \ldots, D_p\}$ be a set of dichotomies in S with $D_i = \{P_i, Q_i\}, i = 1, \ldots, p$. The Constrained Encoding problem consists in finding an encoding of S of minimum length which satisfies all the dichotomies in C.

In (Couderc, 1998), the decision version of Constrained Encoding is reduced to a special coloring problem in a graph G described as follows. With each dichotomy $D_i \in C$, we associate two vertices (P_i, Q_i) and (Q_i, P_i) corresponding to the two possible oriented pairs. Two vertices (P, Q) and (P', Q') are linked by an edge if and only if $P \cap Q' \neq \emptyset$ or $P' \cap Q \neq \emptyset$. In particular (P, Q) and (Q, P) are linked by an edge. Now let $k \geqslant 1$ be a fixed integer. We will show that there exists an encoding of S of length at most k satisfying all dichotomies in C if and only if there exists a k-coloring of a subset of vertices in G such that for each dichotomy $\{P, Q\}$, either vertex (P, Q) or vertex (Q, P) is colored. Clearly, this coloring problem mentioned in (Couderc, 1998) corresponds exactly to deciding whether there exists a selective k-coloring of G with clustering $\mathcal{V} = (V_1, \ldots, V_p)$ defined as $V_i = \{(P_i, Q_i), (Q_i, P_i)\}$, for $i = 1, \ldots, p$, i.e., it corresponds to k-Dsel-Col in $(G, \mathcal{V})$. The following proposition gives the formal proof of the equivalence (mentioned in Couderc, 1998) between Constrained Encoding and Sel-Col in the related graph G.

Proposition 2.1. *Consider an instance* (S, C) *of* Constrained Encoding *and the related clustered graph* $(G, \mathcal{V})$ *as defined above. For any* $k \geqslant 1$, *there is an encoding of length* k *satisfying all dichotomies if and only if* $(G, \mathcal{V})$ *is selective* k*-colorable.*

Proof. Without loss of generality, we may assume that every state is involved in at least one dichotomy. Consider a selective k-coloring of G with respect to $\mathcal{V}$. We define an encoding of S of length k satisfying all dichotomies as follows: for $i = 1, \ldots, k$, consider all vertices being colored with color i; let (P, Q) be such a vertex; then we set the ith component of all states in P to 0 and the ith component of all states in Q to 1; let (P', Q') be a vertex not colored with color i; then we set the ith component of all states in $P' \cup Q'$ to an arbitrary value in $\{0, 1\}$. This clearly gives us an encoding of S of length k satisfying all dichotomies.

Conversely, suppose now that there exists an encoding of S of length k satisfying all dichotomies in C. Then we obtain a selective k-coloring with respect to $\mathcal{V}$ as follows: for $i = 1, \ldots, k$, consider the dichotomies $\{P, Q\}$ such that all states in P have their ith component set to 0 (resp. 1) and all states in Q have their ith component set to 1 (resp. 0); then we color the vertices (P, Q) (resp. (Q, P)) with color i. Now consider two vertices having color i, say (P, Q) and (P', Q'), and suppose they are adjacent. By the construction of G this means that $P \cap Q' \neq \emptyset$ or $P' \cap Q \neq \emptyset$. Without loss of generality, we may assume that there is a state $s_j \in P \cap Q'$. But this means that the encoding of S sets the ith component of s_j to 0 (because $s_j \in P$) and also it sets the ith component of s_j to 1 (since $s_j \in Q'$), a contradiction. Thus we obtain a selective k-coloring with respect to $\mathcal{V}$. □

The graph $G = (V, E)$ constructed above is called a *twin graph*. It has the property that the edges $(P_i, Q_i)(Q_i, P_i), i = 1, \ldots, p$, form a *perfect matching*, i.e., they form a set of pairwise non-adjacent edges such that each vertex is incident to one such edge. In (Demange et al., 2014), it is shown that 2-Dsel-Col is NP-complete in twin graphs even if all clusters consist in two adjacent vertices, and as a corollary that deciding whether there exists an encoding of length at most 2 satisfying all dichotomies of a given set is NP-complete.

Note that 1-Dsel-Col is polynomial-time solvable in twin graphs: due to their symmetry, there exists a stable set intersecting all clusters exactly once if and only if the twin graph is bipartite. Equivalently, deciding whether there exists an encoding of length one satisfying all dichotomies of a given set can be solved in polynomial time.

2.3. Antenna positioning and frequency assignment problem

The Frequency Assignment Problem (FAP) corresponds to a well known application of the graph coloring problem and some of its generalizations (see e.g. Eisenblätter, Grötschel, & Koster, 2002). Given a network of transmitter/receiver units (TRXs for short) spread over a country, the problem consists in assigning a frequency to each TRX taking into account two main constraints. The first constraint is that devices that are close to each other should not get frequencies that are too close to each other in order to avoid interferences. Indeed, interferences may occur on a receptor (e.g. a mobile phone) if there are two TRXs with the same frequency sufficiently close to this receptor. The second constraint is that the spectrum of available frequencies is narrow and consequently a very popular objective is to minimize the total number of frequencies used in the network.

The most basic model involves an interference graph in which the vertices correspond to TRXs and there is an edge between two vertices if the corresponding TRXs are too close so that interferences may occur. Considering a finite number of different frequencies, a feasible assignment using k frequencies corresponds to a k-coloring of the interference graph. Disk graphs (DG) and unit disk graphs (UDG) are popular models of possible interference graphs. A disk graph is the intersection graph of disks in the Euclidean plane and a unit disk graph corresponds to the particular case where all disks have the same radius. In such graphs, the TRXs are associated with the centers of the disks and the radii correspond to the signal power of the TRXs. Thus, the disks represent the areas in which the signals are significantly high; interferences may occur for a receptor at the intersection of two such disks if the corresponding TRXs are given frequencies that are too close to each other (or simply if they get the same frequency). FAP where interferences are allowed up to some level can be handled with graph coloring models where conflicts may occur (Archetti, Bianchessi, Hertz, Colombet, & Gagnon, 2013). This gives the so-called improper coloring notion that can be easily adapted to the selective framework.

The antenna positioning problem (APP) is a second important problem in this domain. Its goal is to select some TRXs positions among a set of possible locations such that several constraints are satisfied and in particular such that the positions allow a good overall covering. The map is therefore divided into several elementary cells and the global APP–FAP consists in positioning a fixed number of TRXs respecting some constraints, allocating each cell to one TRX and assigning frequencies to TRX. The global objective is then to minimize the total number of frequencies used.

The traditional wireless network design process examines these two major issues sequentially. However, in order to achieve better global solutions, one considers processes able to integrate both problems – the positioning and the frequency assignment – and to solve them simultaneously (see e.g. Touhami, Bourjolly, & Laporte, 2009). To our knowledge, only a partial integration has been achieved due to the huge complexity of the whole problem. In particular, many constraints have to be taken into account for APP: coverage rate, physical constraints due to the topography, law constraints, etc.

The following procedure in two phases could be considered to achieve a partial integration. The map is first divided into several areas satisfying topography and population constraints as well as constraints due to law limitations, the quality of service and the expected traffic. Areas are supposed to be connected and define a partition of all elementary cells. Then, in a second phase, exactly one TRX will be installed in each area. To this aim, a finite number of possible locations of this TRX are identified taking into account technical constraints. Thus the problem consists in selecting one position per area such that the chromatic number of the resulting interference graph is as small as possible in order to minimize the number of frequencies used.

This model motivates the study of Sel-Col in DGs and UDGs, in particular with compact clustering as illustrated below. We define a cluster as the set of vertices corresponding to all possible locations for the TRX in a given area. The problem corresponds then exactly to Sel-Col in the related interference graph (a DG or an UDG) with this clustering. It is very natural to assume that for every possible location of a TRX in a given area, the whole area is covered. This means that the area will be completely included in the disk centered on any possible location of the TRX in that area. As a consequence, the vertices corresponding to all possible locations in a same area are pairwise adjacent in the interference graph, which means that under this assumption, the clustering is compact.

When dealing with the complexity of problems in DGs and UDGs, it is worth to remind that deciding whether a given graph is a UDG is NP-complete (Breu & Kirkpatrick, 1998); so it is important to specify whether an intersection model is given in the instance or not. For most of the applications of DG/UDG's, it is natural to suppose that such a model is given and hardness results become stronger under this assumption. Following the general remarks of Section 1.2, k-Dsel-Col is NP-complete in UDGs for any fixed $k \geqslant 3$ even if all clusters contain a single vertex and if an intersection model is known. This is due to the fact that the k-Colorability is NP-complete in planar unit disk graphs for $k = 3$ (Clark, Colbourn, & Johnson, 1990; Peeters, 1991) and in unit disk graphs for $k > 3$ (Gräf, Stumpf, & Weißenfels, 1998) even if an intersection model is known. As already mentioned before, this argument does not prove that the selection process is hard. In (Demange et al., 2013) it is shown that 1-Dsel-Col is NP-complete in graphs isomorphic to nP_3, even if clusters are of size either 2 or 3. This class is trivially included in planar UDGs and consequently, 1-Dsel-Col is NP-complete in planar UDGs with clusters of size 2 or 3, even if an intersection model is known. This implies that the selection process is hard but leaves open the case of compact clustering. Indeed, the class of UDGs being not stable under the addition of an edge, a hardness result in this class does not follow from the above discussion. This question is answered in (Demange et al., 2014) where it is shown that 1-Dsel-Col is NP-complete in planar UDGs of maximum degree 3 with compact clustering and clusters containing either 2 or 3 vertices, even if an intersection model is known.

In Sections 2.4.1 and 2.4.2 we will mention some polynomial cases of Sel-Col in interval graphs (IG) and unit interval graphs (UIG) (definitions will be given in these sections). These particular cases can also be seen as special cases of the present problem in DGs and UDGs which contain respectively IGs and UIGs.

Let us conclude this part by pointing out several remarks on the above mentioned model. Besides compact clustering, another assumption on APP–FAP is natural. If each area is completely covered by a TRX, two vertices corresponding to TRXs located in two adjacent areas are adjacent in the interference graph. This means that the vertices representing locations in two adjacent areas induce a clique in the interference graph. Denote by A the set of edges inside clusters, by B the set of edges between vertices corresponding to locations in two different adjacent areas and by C the set of other edges between vertices corresponding to locations in different areas that are not adjacent. If $C = \emptyset$ then any selection induces a planar graph. Since these graphs have chromatic number at most 4, any solution forms a 4-approximation. Moreover, since every two adjacent areas correspond to a clique in the interference graph, all possible selections give the same planar graph, that is the adjacency graph of the different areas. So, in this case any selection is optimal even if it is hard to evaluate its value since the NP-completeness of 3-Dsel-Col in planar unit disk graphs applies to this case.

2.4. Scheduling problems

2.4.1. Timetabling problems

Let us consider the following timetabling problem. In order to schedule the talks at a conference, each speaker gives his/her available time periods to give his/her talk. The organizer should assign rooms to speakers given the constraint that two talks having intersecting time intervals can not be held in the same room. In order to minimize the costs related to the booking of the rooms, the organizer wants to minimize the number of rooms needed to schedule all the talks. As the speakers agree to give their talk in any available period they specified, the organizer faces the following question: among all possible choices of available periods, how to choose an available period for each speaker such that a scheduling of the talks can be accomplished using a minimum number of rooms? This problem generalizes the scheduling applications of the usual graph coloring problem by allowing some flexibility to the users (in our case the speakers) who can express their availabilities. When each period is represented by a vertex and two vertices are adjacent if they represent intersecting time intervals, we obtain an *interval graph* (IG). Notice that if two speakers are available during a same time period, we will of course introduce two vertices each representing the same time period but for different speakers. We define a cluster as the set of vertices representing available periods of a same speaker. Then the above described scheduling problem corresponds to Sel-Col in an interval graph. Now, if the time periods have all the same length (e.g. in case of talks within some scientific conference or lectures), we deal with Sel-Col in *unit interval graphs* (UIG), i.e., interval graphs admitting an interval representation where all intervals have unit length. On the other hand, if a schedule should be periodically repeated (as for the traffic lights on the crossroads or for continuous production systems), then instead of intervals on a time axis, we rather have arcs on a circle and the related intersection graph is a circular arc graph. Thus we obtain Sel-Col in circular arc graphs the complexity of which has been mentioned in Section 2.1, even for very restrictive cases. Another natural assumption can be made about the available time periods of a given speaker. It would be reasonable to assume that a speaker gives available periods around a given time t. Then the related scheduling problem becomes Sel-Col in (unit) interval graphs with compact clustering since available time intervals of a same speaker intersect at time t.

At this point, let us present some complexity results related to the above described special cases of Sel-Col. We first state the following NP-completeness result from (Demange et al., 2013).

Theorem 2.2 (*Demange et al., 2013*). *The decision version of Sel-Col is NP-complete in the strong sense in interval graphs with compact clustering, even if the vertices of the graph can be partitioned into three cliques.*

However, for a fixed k it becomes polynomial to decide whether a k-colorable selection does exist.

Theorem 2.3 (*Demange et al., 2014*). *k-Dsel-Col can be solved in polynomial time for interval graphs G with compact clustering $\mathcal{V}$. Moreover, if $\chi_{SEL}(G, \mathcal{V}) \leqslant k$, then one can compute a selective k-coloring.*

In (Demange et al., 2014), it is also shown as a corollary of the previous result that if G is an interval graph with compact clustering $\mathcal{V}$ and if $\omega(G_{\mathcal{V}}) \leqslant k$ for a constant k, where $G_{\mathcal{V}}$ is the intersection graph of the sets $C_i = \cup_{I \in V_i} I, i = 1, \ldots, p$ (which is an interval graph since $\mathcal{V}$ is compact), then $\chi_{SEL}(G, \mathcal{V})$ can be computed in polynomial time.

2.4.2. Quality test scheduling

Consider the following scheduling problem. In a factory, a product is manufactured on a production line composed of several tasks $t_1, \ldots, t_r$ some of which may be executed simultaneously. The production line is used only during the day and for security reasons and in order to ensure a correct process of the tasks, a quick test must be executed within each time period $T_1, \ldots, T_p$ of ℓ hours. For organizational reasons, the possible times when the tests can be executed are fixed in advance. Since the execution of a test involves a checking of all the machines (on which different tasks are being processed), the tasks that are executed during the time when the test is taking place, can be disturbed in the sense that some small errors may occur.

So in order to guarantee a high quality end-product for the clients, we want to minimize the impact of the tests on the quality. It is natural to consider that for each task the impact directly depends on the number of tests executed during this task. In some cases the global impact on the final product quality directly depends on the maximum (worst) impact on one task. The problem then consists in selecting tests slots in order to minimize the maximum number of tests during a same task. Another possibility is a system where there is a maximum tolerance of ℓ tests during a single task and one wants to know whether one can schedule all tests such that this tolerance level is respected for all tasks.

These two problems can be respectively represented as Sel-Col and ℓ-Dsel-Col in a special class of graphs, called *linear interval graphs* (LIG), that has been introduced in (Chudnovsky & Seymour, 2012). Take a line L and a finite set of points V of L; take a set of intervals from L (an interval means a proper subset of L homeomorphic to $[0, 1]$). The related linear interval graph $G = (V, E)$ has vertex set V and $u, v \in V$ are adjacent in G if $\{u, v\}$ is a subset of one of the intervals (see Fig. 2 for an example). It follows from this definition that if v_i is adjacent to v_j, with v_i being to the left of v_j in L, then v_i, v_j and all vertices lying between these two vertices in L form a clique. We say that $v_i < v_j$ if v_i lies on the left of v_j in L. When dealing with selective coloring in such graphs, a clustering $\mathcal{V} = \{V_1, \ldots, V_p\}$ of the vertex set is called *consecutive* if for all $v_i, v_j \in V_l$ with $v_i < v_j$, we have $v_r \in V_l$ for all v_r such that $v_i \leqslant v_r \leqslant v_j$.

In our scheduling problem, the real line L is representing time. For each possible time when a test can take place we set a point on L. Let $V_i, i = 1, \ldots, p$, be the set of such points which correspond to possible times when a test can take place during the time period T_i, and let $V = \cup_{i=1}^{p} V_i$ be the set of all such points on L. Thus the points belonging to a same set V_i are necessarily consecutive on L. We represent each task t_j, $j \in \{1, \ldots, r\}$, by an interval I_j with endpoints corresponding to the starting and finishing time of the task. We consider the related linear interval graph G associated with V and the intervals $I_j, j \in \{1, \ldots, r\}$. Clearly the sets $V_1, \ldots, V_p$ corresponding to possible test times within $T_1, \ldots, T_p$ form a partition of V which is a consecutive clustering. Now, our problem consists in choosing in each cluster V_i a vertex such that the size of a maximum clique in the graph induced by the chosen vertices is as small as possible. Since linear interval graphs are perfect (a graph G is perfect if for every induced subgraph H of G we have $\chi(H) = \omega(H)$), it follows that this is equivalent to choosing in each cluster V_i a vertex such that the chromatic number of the graph induced by the chosen vertices is minimized. Thus the first mentioned problem is equivalent to SEL-COL in linear interval graphs with consecutive clustering. The second one corresponds to ℓ–Dsel-Col in the same graph with the same clustering.

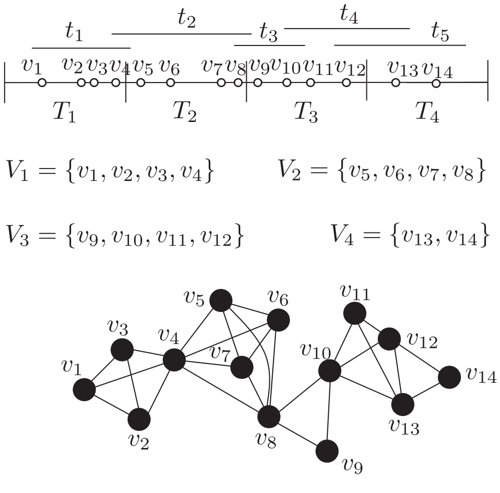
t_1
t_2
t_3
t_4
t_5
v_1 v_2 v_3 v_4 v_5 v_6 v_7 v_8 v_9 v_{10} v_{11} v_{12} v_{13} v_{14}
T_1
T_2
T_3
T_4
$V_1 = \{v_1, v_2, v_3, v_4\}$
$V_2 = \{v_5, v_6, v_7, v_8\}$
$V_3 = \{v_9, v_{10}, v_{11}, v_{12}\}$
$V_4 = \{v_{13}, v_{14}\}$
v_1
v_2
v_3
v_4
v_5
v_6
v_7
v_8
v_9
v_{10}
v_{11}
v_{12}
v_{13}
v_{14}

Fig. 2. An LIG representation and the corresponding graph.

Notice that it follows from (Chudnovsky & Seymour, 2005) that linear interval graphs are equivalent to *proper interval graphs* (PIG) (i.e., interval graphs admitting an interval representation in which no interval properly contains another), which are in turn equivalent to UIG as shown in (Roberts, 1969). In (Demange et al., 2014), it is shown that k-Dsel-Col is solvable in linear time with respect to the number of vertices and edges in LIG with consecutive clustering, leading to a polynomial algorithm for Sel-Col.

As already mentioned, if we deal with repetitive scheduling problems, then we get a graph model with CAG. Then, if each arc has unit length or no arc properly contains another, we obtain respectively *unit circular arc graphs* (UCAG) and *proper circular arc graphs* (PCAG). Also, if the points are located on a circle instead of a line, LIG become *linear circular arc graphs* (LCAG). Again it follows from (Chudnovsky & Seymour, 2005) that PCAG and LCAG are equivalent. However, the approach used in the proof of polynomial time solvability of k-Dsel-Col in LIG with consecutive clustering cannot be easily generalized to the circular case; this leaves k-Dsel-Col in LCAG with consecutive clustering as an open problem.

2.4.3. Berth allocation problem

Let us now describe another scheduling problem which is known as the *berth allocation problem* (various versions of this problem exist in the literature; see for instance (Türkoğulları, Taşkın, Aras, & Altınel, 2014) for more information about the version considered here). Let us first describe the problem and then give the graph model which amounts to 1-Dsel-Col in the so-called rectangle intersection graphs (which generalize interval graphs).

Assume we are given a set of p ships which need to be loaded/unloaded in a port. For each ship, we are given its length as well as the time it takes to load/unload the ship. The dock to which the ships have to be moored contains several berths (at predefined positions) where the ships can stay while being loaded/unloaded. The berth allocation problem then consists in deciding which ship will be loaded/unloaded at which berth and during which time interval. Here we consider the case where two ships cannot be simultaneously loaded/unloaded at a same berth and depending on their lengths, they may not be loaded/unloaded at two distinct berths that are too close to each other.

Now consider the plane where the x-axis represents the dock (i.e., its length) and the y-axis represents time (i.e., 24 h of a day). Each berth is then represented by a point on the x-axis corresponding to its real position on the dock. Now the possible loading/unloading of a ship of length ℓ at a berth located at position d which takes place between time t_1 and t_2 can be represented as a rectangle in the plane of size $\ell \times (t_2 - t_1)$ with corners

$(d-\frac{\ell}{2},t_1)$, $(d-\frac{\ell}{2},t_2)$, $(d+\frac{\ell}{2},t_1)$, $(d+\frac{\ell}{2},t_2)$ (see Fig. 3). Note that the security distance that should be held between two ships on neighboring berths (respectively the security time gap between two loading/unloading on the same berth) is taken into account in the length ℓ (respectively the loading/unloading time) of each ship.

For each ship, we are given the possible times when it may be loaded/unloaded as well as the possible berths at which it may stay (due to possible resource constraints a ship may not be loaded/unloaded at any time and at any berth). Hence, for each ship $i, i = 1, \ldots, p$, we may draw the set V_i of all rectangles corresponding to all loading/unloading possibilities. Now we construct a graph $G = (V, E)$ as follows. $V = \cup_{i=1}^{p} V_i$ is the set of vertices and we link two vertices if and only if the corresponding rectangles intersect. Thus we obtain a so-called *rectangle intersection graph* (see for instance Rim & Nakajima, 1995). Furthermore we define the clustering $\mathcal{V} = (V_1, \ldots, V_p)$ of V (a cluster corresponds to all loading/unloading locations and times for a given ship). Note that in Fig. 3, only one possible pair of position and time (hence only one possible rectangle) per ship is represented for reason of simplicity. Now clearly, the berth allocation problem is equivalent to choosing exactly one vertex in each cluster (i.e., one rectangle for each ship) such that the chosen vertices induce a stable set (i.e., the chosen rectangles pairwise do not intersect). Thus it is equivalent to 1-Dsel-Col in G with clustering $\mathcal{V}$. Since 1-Dsel-Col is NP-complete in interval graphs (Demange et al., 2013) and rectangle intersection graphs generalize interval graphs to two dimensions, this problem is also NP-complete. In the literature, there has been various ways to deal with the berth allocation problem such as heuristics, simulation, multi-agent and integer programming approaches (Stahlbock & Voß, 2008). It would be interesting to explore graph theoretical approaches as well in order to solve the berth allocation problem.

2.5. Multiple stacks TSP

In this part, we describe a model that motivates the selective graph coloring problem in permutation graphs. Let us first give some definitions. A *permutation graph* is a graph whose vertices represent the elements of a permutation and whose edges represent pairs of elements that are reversed by the permutation (Golumbic, 2004). Among various alternative ways of defining permutation graphs, the following one is especially relevant in order to illustrate multiple stacks TSP. Given a set of intervals on the real line containing a common point, we consider the graph G whose vertices are the intervals and where two vertices are adjacent if the corresponding intervals *overlap*, i.e., they have a non-empty intersection and none is included in the other. The relation with permutations can be illustrated as follows. Consider n intervals sharing a common point on the real line. Name the intervals $I_1, \ldots, I_n$ such that the indices are taken in non-increasing order of the left endpoints of the intervals. Now number the right endpoints of the intervals according to the intervals they belong to, i.e., the right endpoint of interval I_j is numbered j, for $j = 1, \ldots, n$. Then the order in which the numbers $1, \ldots, n$ appear on the real line corresponds to the permutation σ associated with the intervals and the graph G corresponds to the permutation graph[1] of σ (Gavril, 1973) (see Fig. 4 for an example).

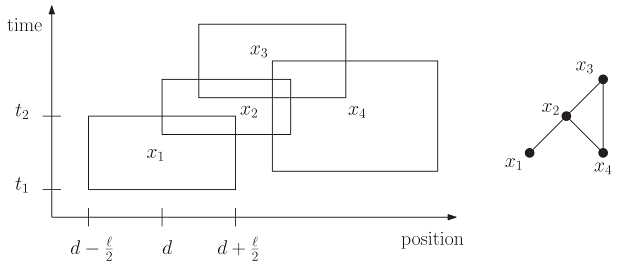

Fig. 3. Four ships represented by rectangles on the plane of position and time, and their rectangle intersection graph.

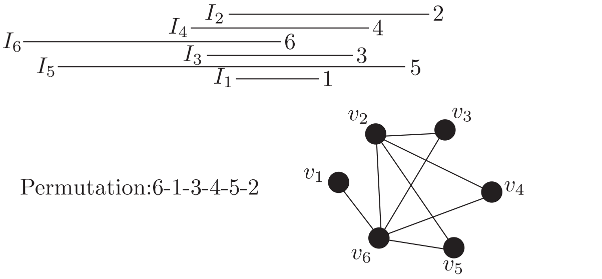

Fig. 4. The right endpoints of the intervals define a permutation σ. On the bottom right the corresponding permutation graph in which every vertex v_i represents an interval I_i, for $i = 1, \ldots, 6$.

Consider now the so-called double TSP problem with multiple stacks (Petersen & Madsen, 2009): we are given two networks; items are collected in the first network during a pick-up tour and delivered in the second network during a delivery tour. Between collection and delivery, items are packed in containers containing different rows called *tracks*. Then containers are transported (for example through shipping) and then delivered during the delivery tour. The problem consists in determining the two tours and the allocation of items to tracks such that the total cost is minimized with the constraint that no swap can be operated on tracks. Given that tracks have first-in last-out structures, items allocated to each track are delivered in reverse order of their collecting order.

An interesting particular case occurs when both tours are fixed for every day pick-up and delivery. In this case, the problem reduces to the allocation of items to tracks which can be represented as a coloring problem in a permutation graph. Consider indeed a linear axis with first all pick-up points on the left side arranged in their visit order and on the right side delivery points in their visit order. Each item is associated to an interval starting at one pick-up point and ending at one delivery point. We consider the permutation graph corresponding to this set of intervals. Two items corresponding to overlapping intervals cannot be allocated to the same track if one wants to avoid swaps, which leads to a coloring problem in a permutation graph.

Suppose now that for each item we have different possibilities of pick-up and/or delivery points. All these possibilities induce several intervals associated with one item. We then need to select exactly one interval per item, corresponding to one pick-up point and one delivery point in order to minimize the total number of tracks needed. Considering the permutation graph G associated with all intervals and defining a clustering $\mathcal{V}$ such that all the vertices corresponding to possible intervals associated with one item are in a same cluster, the problem reduces to Sel-Col in $(G, \mathcal{V})$. We note that as shown in (Demange et al., 2013), 1-dsel-col is NP-complete in permutation graphs.

3. Conclusion

This paper presents a collection of applications, from various domains, of the selective graph coloring problem. To the best of our knowledge, only the application for wavelength assignment in optical networks was identified in the literature as directly

[1] Note that if the condition that all intervals have a common point does not hold, then the graph G is an *overlap graph*, a class of graphs that is larger than permutation graphs.

connected with this graph coloring problem. We pointed out some typical situations for which a usual coloring problem model turns into a selective graph coloring problem model and we can expect that following a similar thought process, many other usual graph coloring applications could be still relevant in a selective framework. For some of the mentioned models and in particular for some scheduling problems and frequency allocation problems, many generalizations of the usual graph coloring have been defined, like T-coloring, multi-coloring, split-coloring, weighted coloring, list-coloring, bounded coloring etc. Clearly, all these generalizations could be applied to the selective case with very similar motivating frameworks. Each domain of application motivates the study of the problem in some specific class of instance graphs. This also emphasizes the question of the computational complexity of Sel-Col and of k-Dsel-Col in these classes. Such a systematic study was initiated in (Demange et al., 2013) and continued in (Demange et al., 2014) for some cases discussed here. The main complexity results related to the applications considered in this paper are mentioned here.

For all hard cases, a next step will be to design exact methods together with efficient approximation algorithms or heuristics. In order to do so, one may use efficient methods that have been developed before for problems closely related to SEL-COL. More specifically, exact or heuristic methods for (unit) interval graphs and (unit) disk graphs would be of particular interest.

Acknowledgements

This work was partially supported by TUBITAK PIA BOSPHORUS Grant No. 111M303 and PHC BOSPHORE 2012 PROJECT No. 26284RB. M. Demange was also partially supported by the French Agency for Research under the DEFI program TODO, ANR-09-EMER-010. Both supports are greatly acknowledged.

References

Archetti, C., Bianchessi, N., Hertz, A., Colombet, A., & Gagnon, F. (2013). Directed weighted improper coloring for cellular channel allocation. *Discrete Applied Mathematics*. http://dx.doi.org/10.1016/j.dam.2013.11.018.

Beauquier, B., Bermond, J. -C., Gargano, L., Hell, P., Perennes, S., & Vaccaro, U. (1997). Graph problems arising from wavelength-routing in all-optical networks. In *Proceedings of the 2nd workshop on optics and computer science, 1997, Satellite of IPPS'97*.

Bonomo, F., Cornaz, D., Ekim, T., & Ries, B. (2013). Perfectness of clustered graphs. *Discrete Optimization, 10*, 296–303.

Breu, H., & Kirkpatrick, D. G. (1998). Unit disk graph recognition is NP-hard. *Computational Geometry: Theory and Applications, 9*(1-2), 3–24.

Chudnovsky, M., & Seymour, P. (2005). The structure of claw-free graphs. *Surveys in Combinatorics, London Mathematical Society Lecture Note Series, 327*, 153–171.

Chudnovsky, M., & Seymour, P. (2012). Claw-free graphs VII. Quasi-line graphs. *Journal of Combinatorial Theory Series B, 102*, 1267–1294.

Clark, B. N., Colbourn, C. J., & Johnson, D. S. (1990). Unit disk graphs. *Discrete Mathematics, 86*(1–3), 165–177.

Couderc, O. (1998). A new paradigm for dichotomy-based constrained encoding. In *Proceedings of design, automation and test in Europe (DATE), Paris*.

Dandashi, A., & Al-Mouhamed, M. (2010). Graph Coloring for class scheduling. In *Proceedings of IEEE/ACS international conference on computer systems and applications (AICCSA)*.

Demange, M., Ekim, T., & de Werra, D. (2009). A tutorial on the use of graph coloring for some problems in robotics. *European Journal of Operational Research, 192*(1), 41–55.

Demange M., Monnot J., & Ries B. (2013). Selective coloring in interval graphs. Manuscript.

Demange M., Ekim T., & Ries B. (2014). On minimum and maximum selective graph coloring in several graph classes. Manuscript.

Demange, M., Monnot, J., Petrica, P., & Ries, B. (2013). On the complexity of the selective graph coloring problem in some special classes of graphs. *Theoretical Computer Science*. in press, http://dx.doi.org/10.1016/j.tcs.2013.04.018.

Eisenblätter, A., Grötschel, M., & Koster, A. M. C. A. (2002). Frequency planning and ramifications of coloring. *Discussiones Mathematicae Graph Theory, 22*(1), 51–88.

Erlebach, T., & Jansen, K. (2001). The complexity of path coloring and call scheduling. *Theoretical Computer Science, 255*, 33–50.

Frota, Y., Maculan, N., Noronha, T. F., & Ribeiro, C. C. (2010). A branch-and-cut algorithm for partition coloring. *Networks, 55*(3), 194–204.

Gamache, M., Hertz, A., & Ouellet, J. (2007). A graph coloring model for a feasibility problem in crew scheduling. *Computers and Operations Research, 34*, 2384–2395.

Garey, M. R., Johnson, D. S., Miller, G. L., & Papadimitriou, C. H. (1980). The complexity of coloring circular arcs and chords. *SIAM Journal on Algebraic and Discrete Methods, 1*(2), 216–227.

Gavril, F. (1973). Algorithms for a maximum clique and a maximum independent set of a circle graph. *Networks, 3*, 261–273.

Giaro, K., Kubale, M., & Obszarski, P. (2009). A graph coloring approach to scheduling of multiprocessor tasks on dedicated machines with availability constraints. *Discrete Applied Mathematics, 157*(17), 3625–3630.

Golumbic, M. C. (2004). *Algorithmic graph theory and perfect graphs*. Computer Science and Applied Mathematics, Academic Press.

Gräf, A., Stumpf, M., & Weißenfels, G. (1998). On coloring unit disk graphs. *Algorithmica, 20*(3), 277–293.

Guangzhi, L., & Simha, R. (2000). The partition coloring problem and its application to wavelength routing and assignment. In *Proceedings of the first workshop on optical networks, Dallas*.

Hoshino, E. A., Frota, Y., & de Souza, C. C. (2011). A branch-and-price approach for the partition coloring problem. *Operations Research Letters, 39*(2), 132–137.

Noronha, T. F., & Ribeiro, C. C. (2006). Routing and wavelength assignment by partition colouring. *European Journal of Operational Research, 71*(3), 797–810.

Peeters, R. (1991). *On coloring j-unit sphere graphs*. Technical Report, Tilburg University.

Petersen, H. L., & Madsen, O. B. G. (2009). The double traveling salesman problem with multiple stacks – Formulation and heuristic solution approaches. *European Journal of Operational Research, 198*(1), 139–147.

Qu, R., Burke, E. K., & McCollum, B. (2009). Adaptive automated construction of hybrid heuristics for exam timetabling and graph colouring problems. *European Journal of Operational Research, 198*(2), 392–404.

Rim, C. S., & Nakajima, K. (1995). On rectangle intersection and overlap graphs. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, 42*(9), 549–553.

Roberts, F. S. (1969). Indifference graphs. In F. Harary (Ed.), *Proof techniques in graph theory* (pp. 139–146). NY: Academic Press.

Sotskov, Y. N., Dolgui, A., & Werner, F. (2001). Unit-time job-shop scheduling via mixed graph coloring. *International Journal of Mathematical Algorithms, 2*(4), 289–325.

Stahlbock, R., & Voß, S. (2008). Operations research at container terminals: A literature update. *OR Spectrum, 30*, 1–52.

Talaván, P. M., & Yáñez, J. (2008). The graph coloring problem: A neuronal network approach. *European Journal of Operational Research, 191*(1), 100–111.

Touhami, S., Bourjolly, J.-M., & Laporte, G. (2009). Partial integration of frequency allocation within antenna positioning in GSM mobile networks. *European Journal of Operational Research, 193*(2), 541–551.

Tracey, J. H. (1966). Internal state assignments for asynchronous sequential machines. *IEEE Transactions on Electronic Computers, 15*(4), 551–560.

Türkoğulları, Y., Taşkın, Z. C., Aras, N., & Altınel, K. (2014). Optimal berth allocation and time-invariant quay crane assignment in container terminals. *European Journal of Operational Research, 235*(1), 88–101.

Zang, H., Jue, J. P., & Mukherjee, B. (2000). A review of routing and wavelength assignment approaches for wavelength routed optical WDM networks. *Optical Networks Magazine, 1*(1), 47–60.

Zhou, Y., & Poo, G.-S. (2005). Optical multicast over wavelength-routed WDM networks: A survey. *Optical Switching and Networking, 2*(3), 176–197.