



On Split-Coloring Problems

T. EKIM

tinaz.ekim@epfl.ch

D. de WERRA

dewerra.ima@epfl.ch

Institute of Mathematics—ROSE, Ecole Polytechnique Fédérale de Lausanne, CH-1015 Lausanne-Ecublens, Switzerland

Received November 19, 2003; Revised June 1, 2005; Accepted June 2, 2005

Abstract. We study a new coloring concept which generalizes the classical vertex coloring problem in a graph by extending the notion of stable sets to split graphs. First of all, we propose the packing problem of finding the split graph of maximum size where a split graph is a graph $G = (V, E)$ in which the vertex set V can be partitioned into a clique K and a stable set S . No condition is imposed on the edges linking vertices in S to the vertices in K . This maximum split graph problem gives rise to an associated partitioning problem that we call the split-coloring problem. Given a graph, the objective is to cover all his vertices by a least number of split graphs. Definitions related to this new problem are introduced. We mention some polynomially solvable cases and describe open questions on this area.

Keywords: split-coloring, vertex covering by split graphs, partitioning, packing

1. Introduction

Packing and partitioning problems in graphs have been widely studied by many authors for years. Maximum stable set and maximum clique problems are the basic packing problems and their associated partitioning problems are called respectively graph coloring and vertex covering by cliques. These problems, especially the graph coloring have many applications in scheduling (de Werra et al., 2005), timetabling, etc. but on the other hand, they are known to be among the most difficult NP-hard problems. In this paper, we define a new problem which includes the graph coloring problem and extends its field of application.

First of all, we consider the problem of finding in a graph G an induced split graph of maximum size. A split graph is a graph $G = (V, E)$ in which the vertex set V can be partitioned into a clique K and a stable set S . No condition is imposed on the edges linking vertices in S to the vertices in K . This maximum split graph problem gives rise to an associated partitioning problem that we call the split-coloring problem. It is a vertex coloring problem of a graph where we replace stable sets by split graphs. Given a graph, the objective is to cover all vertices by the least number of induced split graphs.

This problem is motivated both by applications and by theoretical properties as auto-complementarity and heredity. As a related application of the concept of split graph, we can mention a telecommunication problem which consists of assigning terminal nodes to concentrators and install concentrators and links in order to ensure an optimal traffic routing. We observe that split graphs appear in this problem since concentrators are represented by

cliques and terminal nodes assigned to a same concentrator constitute a stable set (Gourdin et al., 2002). The paper is organized as follows : in the second section some preliminary results are presented. We give an integer programming model of the split-coloring problem in section three. The fourth section is the resolution of the split-coloring problem in some special classes of graphs as for instance cacti. In section five, we give lower and upper bounds for χ_S in arbitrary graphs. Finally, the last section mentions some open questions for further research.

In what follows, K_l denotes a clique on l vertices, C_l a cycle on l vertices and P_l a path on l vertices.

2. Preliminary results

Split graphs are characterized as follows:

Theorem 1 (Földes and Hammer, 1976). *For every graph $G = (V, E)$, the following conditions are equivalent:*

1. G is a split graph,
2. G and \bar{G} are triangulated,
3. G does not contain $2K_2$, C_4 or C_5 .

One can remark that as the complement of a split graph is again a split graph, solving the split-coloring problem in a graph is equivalent to solve it in the complementary graph. More generally, if we can solve it in a class of graphs \mathcal{C} then we can also solve it in the class $\bar{\mathcal{C}}$ containing the complements of all graphs in \mathcal{C} and vice versa, thereby the problem of split-coloring the vertices of a graph is an *auto-complementary* problem.

On the other hand, we know that if G is a split graph then all subgraphs of G are also split graphs. This *hereditary* property with respect to vertices can be effectively exploited in the conception of algorithms.

The split-coloring problem is a very intriguing one because it generalizes the classical graph coloring problem and furthermore, it has never been studied systematically to our knowledge. A corresponding edge covering problem has been studied though (see Mahadev and Peled, 1995).

Let us call a subset S of vertices *split-independent* (*s.i.*) if the subgraph induced by S is a split graph. Then, our basic packing problem called *maximum split graph*, consists in finding the *split-independence number* $\alpha_S(G)$ of G , which is the maximum cardinality $|S|$ of a s.i. set S . In other words, we search for a maximum size induced split graph in G . It should be trivially noted that $\alpha_S(G) \geq \max(\alpha(G), \omega(G)) + 1$ where $\alpha(G)$ is the stability number and $\omega(G)$ is the maximum clique size. Moreover, one can observe that for any graph G , we have $\alpha(G) + \omega(G) - 1 \leq \alpha_S(G) \leq \alpha(G) + \omega(G)$ since a maximum clique and a maximum stable set have at most one vertex in common. The problem of finding $\alpha_S(G)$ is obviously NP-hard:

Theorem 2. *For a fixed k and a given graph G , it is NP-complete to determine whether $\alpha_S(G) \geq k$.*

Proof: The maximum clique problem is reduced to maximum split graph problem. Let the graph $G = (V, E)$ (which is not a stable set) be an instance of maximum clique problem. We will consider the graph $G' = G \oplus I_N$ obtained from G by adding a stable set I_N of size $N \geq |V|$ which is completely linked to G , i.e., any vertex of V is linked to every vertex of I_N . Assume there is an algorithm for the maximum split graph problem which gives $\alpha_S(G')$ when applied to G' . Now, note that I_N is the only maximum stable set in G' . On the other hand, any maximum clique of G' is in the form $K_{\max} \cup \{x\}$ where K_{\max} is a maximum clique of G and where $x \in I_N$. This implies that there are no disjoint maximum clique and maximum stable set in G' , hence the graph induced by $K_{\max} \cup I_N$ is a maximum split graph in G' of cardinality $\alpha_S(G') = \omega(G') + N - 1 = \omega(G) + N$. The size of a maximum clique in G could be then easily computed by the formula $\omega(G) = \alpha_S(G') - N$. But this is not possible since maximum clique is NP-complete, consequently, so it is the maximum split graph problem. \square

The *split-coloring problem* consists in minimizing the number of split graphs which cover all the vertices of G . This minimum number is called the *split-chromatic number* and we denote it by χ_S . A trivial upper bound for $\chi_S(G)$ is $\min(\chi(G), \theta(G))$ where $\chi(G)$ is the chromatic number and $\theta(G)$ is the minimum number of cliques covering the vertices of G . We say that a graph $G = (V, E)$ is *k-split-colorable* if there exists a partition of V into k split graphs. Although split graphs had been extensively studied by many authors (see Földes and Hammer, 1976, 1977; Hammer and Simeone, 1981; Benzaken et al., 1985; Chernyak and Chernyak, 1991), this problem does not seem to have been studied to our knowledge. The only coloring problem related to split graphs which appears extensively in the literature deals with the case where the edge set of a graph has to be covered (Chernyak and Chernyak, 1991; Mahadev and Peled, 1995). However, this problem of edge coloring by split graphs has neither the hereditary character nor the auto-complementarity property mentioned above. We know that split graphs can be recognized in $O(|V|^2)$ time (Mahadev and Peled, 1995). In other words, the problem of determining for a given graph $G = (V, E)$ whether $\chi_S(G) = 1$ can be solved in $O(|V|^2)$ time.

Now, let us define a new notion which will be useful in our study.

Definition 1. A graph G is called *k-split-critical* if and only if $\chi_S(G) = k$ and $\forall v$, $\chi_S(G - v) = k - 1$.

According to this definition, $2K_2$, C_4 and C_5 are obviously the unique 2-split-critical structures.

Let H be a graph, we denote by kH a graph having k connected components isomorphic to H . In what follows H will be replaced by an odd cycle or a clique. By abuse of language, we often call kH , the graph consisting of k induced H . In order to characterize some 3-split-critical graphs and as a preliminary for further developments, we mention the following three facts. In what follows, OC denotes an odd cycle, i.e., a cycle on an odd number of vertices.

Fact 1. For any odd cycle OC of length at least 5, we have $\chi_S(OC) = 2$; a 2-split-coloring is obtained by choosing one clique (of size one or two) and two stable sets.

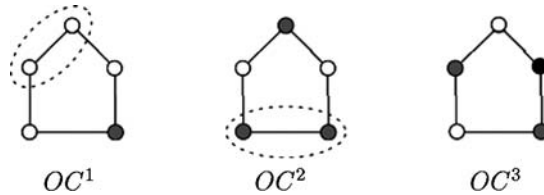


Figure 1. $3OC$ is 3-split-colorable.

Fact 2. For any two induced odd cycles and of length at least 5, we have $\chi_S(2OC) = 2$; a 2-split-coloring is obtained by choosing one clique on each cycle and two stable sets in the remaining graph.

Fact 3. Three induced odd cycles cannot be 2-split-colored, i.e., $\chi_S(3OC) \geq 3$.

Fact 3 is easily deduced from fact 2 since the third odd cycle can not be partitioned into two stable sets that we could color with the first two colors. Therefore, we have to assign a new color to at least one vertex of the third odd cycle as shown in figure 1. Cliques are encircled so that one may distinguish them from the stable sets of same color.

Proposition 1. A graph G consisting of three induced odd cycles $3OC$ or its complement $\bar{G} = \overline{3OC}$ is 3-split-critical.

Proof: It suffices to notice that we cannot do better than the coloring in figure 1 to split-color $3OC$ and that in this way, the inequality of fact 3 turns to be an equality. In addition, the auto-complementarity of the split-coloring problem gives the result. \square

Proposition 1 means that whenever a graph G contains a subgraph $3OC$ or its complement, the split-chromatic number of G is at least 3. However, the number of induced odd cycles do not play any key role in the determination of χ_S for $\chi_S \geq 3$ because $\chi_S(mOC) = 3$, $\forall m \geq 3$. We also know that the forbidden induced graphs of Proposition 1 do not cover all 3-split-colorable cases; there are examples of graphs which do not contain any of these structures but where $\chi_S(G) = 3$. It would be interesting to characterize 2-split-colorable graphs by forbidden induced configurations. On the other hand, the results of Brandstädt et al. in Brandstädt et al. (1998) implies that 2-split-colorability is polynomially solvable in all graphs. The following theorem can be derived from results in Brandstädt et al. (1998).

Theorem 3. For fixed $k \geq 3$ and a graph G , it is NP-complete to determine whether $\chi_S(G) \leq k$.

Proof: We prove this by reduction from the problem of vertex covering by cliques. Given an instance G of vertex covering by cliques, we transform this graph into G' by adding k

stable sets of size $k + 1$ each. They are then completely linked to each other and to G . It is clear that in a k -split-coloring of G' , all color classes will be a split graphs having one stable set of size $k + 1$ and a clique in G . Otherwise, assume that one stable set is taken in G and the other $k - 1$ stable sets are all added stable sets of size $k + 1$. But then, there remains one added stable set which has no vertex in any already fixed stable sets and hence, whose vertices have to be partitioned into k cliques which is impossible since it has $k + 1$ vertices. Therefore, it suffices to see that there is a k -split-coloring for G' if and only if there is a vertex covering by cliques of the vertices of G of size k . \square

Although split-coloring is NP-complete in general, there are some easier cases. For example, bipartite graphs and trees are 2-colorable in the classical sense so they are also 2 split-colorable (unless we have a split graph).

We have already seen that $3K_3$ and $\overline{3K_3}$ are 3-split-critical graphs. In fact, one may generalize this result to the case of k -split-coloring:

Proposition 2. kK_k , k induced cliques of size k , and $\overline{kK_k}$, k joined stable sets of size k with complete links, are k -split-critical.

Proof: One sees immediately that we cannot have a $(k - 1)$ -split-coloring. To split-color kK_k with k colors we assign repetitively a new color to a split graph consisting of an inclusion-wise maximal uncolored (sub-)clique of a K_k and a stable set having one vertex in each remaining maximal (sub-)clique. One may observe that removing any vertex makes our graph $(k - 1)$ -split-colorable. \square

On the other hand, we notice that mK_k with arbitrary large m is still k -split-colorable, i.e., we do not increase χ_S by adding new K_k 's, but it is no more k -split-critical. This fact can be observed for mK_4 in figure 2 where cliques of different split graphs are encircled so that one may distinguish them from the stable sets of same color.

We will see in Section 5 that the above proof gives a hint for computing a lower bound of $\chi_S(G)$ for arbitrary graphs.

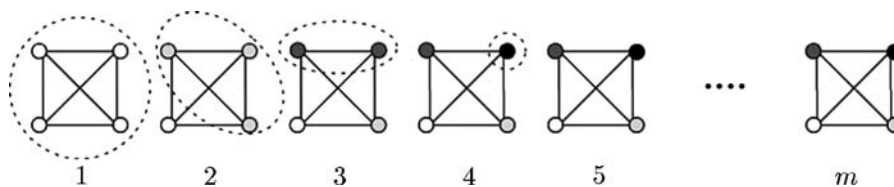


Figure 2. mK_4 is 4-split-colorable.

3. An integer programming model

One may suggest the following integer programming model for our split-coloring problem in a given graph $G = (V, E)$ with $|V| = n$.

$$\min \quad \sum_j z_j$$

$$x_{ij} = \begin{cases} 1 & \text{if vertex } i \text{ is in the clique of split graph } j \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

$$y_{ij} = \begin{cases} 1 & \text{if vertex } i \text{ is in the stable set of split graph } j \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

$$x_{ij} + x_{kj} \leq 1 \text{ if } ik \notin E, j = 1, \dots, l \quad (3)$$

$$y_{ij} + y_{kj} \leq 1 \text{ if } ik \in E, j = 1, \dots, l \quad (4)$$

$$\sum_i (x_{ij} + y_{ij}) \leq nz_j, j = 1, \dots, l \quad (5)$$

$$\sum_j (x_{ij} + y_{ij}) = 1, i = 1, \dots, n \quad (6)$$

$$z_j \in \{0, 1\}, j = 1, \dots, l$$

Here l is an upper bound of $\chi_S(G)$. We may choose $l = \lceil \frac{n}{3} \rceil$ since G can be trivially $\lceil \frac{n}{3} \rceil$ -split-colored by assigning one color to any triple of vertices.

Constraints of type 1 and 2 define the decision variables x_{ij} which give the vertices of G to be assigned to the clique of the split graph j and y_{ij} the vertices in G to be assigned to the stable set of the split graph j . Constraints (3) express that no two vertices not linked in G can be in a same clique and conversely, constraints (4) mean that no two vertices linked in G can be in a same stable set. Constraints (5) say that the number of vertices assigned to one split graph can not exceed n if this split graph exists and 0 otherwise. Finally, constraints (6) make sure that every vertex is colored by exactly one color. The number of colors used is then given by the variables z_j ; $z_j = 1$ means that color j is used in a solution and $z_j = 0$ says that the color j is not used.

One can notice that this model has $O(n^2)$ decision variables and $O(n^3)$ constraints. Advanced techniques have to be applied in order to solve this integer program for graphs having a large number of vertices. However its interest is theoretical and this formulation may be a basis for more efficient formulation involving additional constraints.

The above formulation suggests the following representation of our split coloring problem: let k be a positive integer and let $1, \dots, n$ be the nodes of G . We construct a graph $G(k)$ by first taking k copies G^1, \dots, G^k of G , let \hat{y}_{ij} ($i = 1, \dots, n$) be the nodes of G^j . Then we take k copies $\tilde{G}^1, \dots, \tilde{G}^k$ of \tilde{G} (the complement of G); let \hat{x}_{ij} ($i = 1, \dots, n$) be the nodes of \tilde{G}^j . Then for each i ($1 \leq i \leq n$) we form a clique on nodes $\hat{y}_{i1}, \dots, \hat{y}_{ik}, \hat{x}_{i1}, \dots, \hat{x}_{ik}$. The resulting graph is $G(k)$. We can then state:

Proposition 3. G has a k -split-coloring iff $G(k)$ has a stable set S with $|S| = n$.

Proof: There is a one-to-one correspondence between stable sets S with n nodes in $G(k)$ and k -split-colorings of G or equivalently integral solutions of the integer LP model (where k colors are used):

$$\begin{aligned}\hat{y}_{ij} \in S &\Leftrightarrow y_{ij} = 1 \\ \hat{x}_{ij} \in S &\Leftrightarrow x_{ij} = 1\end{aligned}$$

This means that \hat{y}_{ij} is in S if and only if node i is in the stable set of the split graph j and similarly, \hat{x}_{ij} is in S if and only if node i is in the clique of the split graph j . \square

Remark. If we simply consider $G(1)$, then one sees there is a one-to-one correspondence between (maximum) split graphs in G and (maximum) stable sets in $G(1)$. It is interesting to notice that as soon as G contains an induced P_3 , $G(1)$ is not perfect (it contains an induced C_5).

4. Split-coloring in some classes of graphs

In this section, we study the maximum split graph and split-coloring problems in some restricted classes of graphs where we may hope to find polynomial algorithms.

4.1. *Cacti*

A *cactus* is defined as a connected simple graph where no two elementary cycles share an edge. In other words, no two elementary cycles have more than one common vertex. Let us remark that a tree is a cactus. Conversely, when each cycle of a cactus is contracted to a vertex, then we get a tree.

Note that in cacti, the size of a clique is at most three. For the split-coloring of cacti, we will concentrate on elementary odd cycles since the only connected components remaining after coloring odd cycles are even cycles, paths and trees and once again, they are 2-colorable in the classical sense, i.e., 2-split-colorable by choosing only two stable sets. Moreover, we know that a cactus is bipartite if it contains no (elementary) odd cycles (de Werra et al., 2005) and hence it is 2-split-colorable. On the other hand, cacti are 3-split-colorable because they are 3-colorable in the classical sense. In conclusion, the split-coloring of cacti boils down to be a decision between 2 and 3-split-colorability. In the process of finding the split independence number of a cactus, the assignment of a vertex to a stable set and/or to a clique associated to a split graph is strongly related to the structure (even or odd cycle, bridge) to which this vertex belongs. We will see that we have to proceed differently for vertices located in an odd cycle than for others. First, we need a procedure listing all odd cycles in a cactus.

Although in arbitrary graphs the number of odd cycles can grow exponentially in the number of vertices, their number is bounded above by $\lceil \frac{n-1}{2} \rceil$ in cacti. This bound corresponds to the number of odd cycles in a cactus where a single central vertex is shared by all cycles of length three formed by the other $n - 1$ vertices. In order to include odd cycles of a cactus

$G = (V, E)$ with $|V| = n$, $|E| = m$ in a list \mathcal{L} , we first observe that any 2-connected component of a cactus is a cycle and we apply the algorithm of Tarjan (1972) to detect all separating vertices and all cycles. This is a depth first search algorithm which provides a list of blocks and a list of separating vertices (cut-nodes) for any arbitrary graph in time linear in $O(m)$. We recall that a block is a maximal non-separable subgraph, so it may be either a 2-connected component or an isthmus (Berge, 1983). Having a list of all blocks, it suffices to eliminate edges and cycles of even length to obtain \mathcal{L} .

4.1.1. Split independence number in cacti. In this section, we concentrate on the problem of finding an induced split graph of maximum size in a cactus $G = (V, E)$, $|V| = n$, $|E| = m$. In other words, we try to find a stable set S and a clique C in G such that $|S \cup C|$ is maximum. For this purpose, we will first describe a simple algorithm for a maximum stable set in a cactus, since, to our knowledge, such an algorithm has not been given elsewhere.

We adopt a dynamic programming approach. Given a cactus G and its lists of cycles and separating vertices, one can construct a corresponding arborescence A in the following way:

- Each cycle, edge not contained in a cycle and separating vertex in G is represented by a vertex in A ,
- Vertices corresponding to two cycles or to two separating vertices in G are not linked in A ,
- Two vertices are linked in A if and only if they correspond to a cycle and a separating vertex contained in it,
- All the edges in A are oriented towards a chosen separating vertex (preferably one closest to a leaf).

Having such an arborescence A , one can label separating vertices (v_j) in increasing order according to the exploration sense of A . Then, another label may be given to vertices representing cycles in such a way that labels of cycles linked to and oriented towards v_j would be smaller than the labels of cycles linked to and oriented towards v_i whenever $j < i$. An example of labeling can be seen in the figure 3 where the labels of cycles, edges and

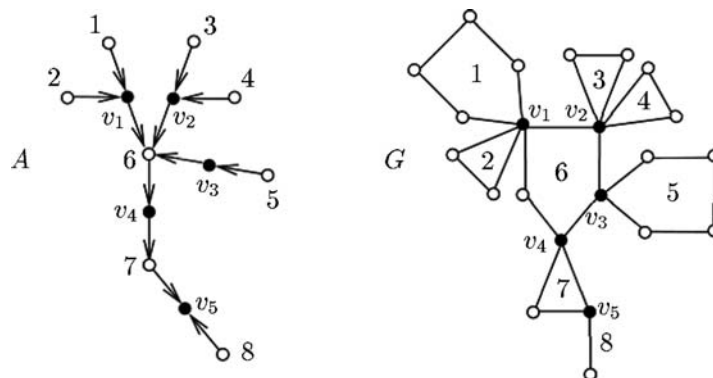


Figure 3. An arborescence A corresponding to graph G .

separating vertices (black ones) of G are determined by means of the arborescence A . This way of labeling of G is not unique but defines perfectly a direction for exploring the original graph G by dynamic programming.

What we have to do now is to describe one step of the procedure which consists in finding a maximum stable set in a caterpillar, i.e., a tree with the property that the removal of its leaves results in a path. One may start in the exploration order of separating vertices and compute the stability number in the subgraph G_1 lying in the lower part of v_1 (which consists of cycle(s) oriented towards v_1) according to two scenarios: v_1 is excluded in the maximum stable set of G_1 or it is authorized. Hence, for any separating vertex v_j , we will compute two weights; $v_j^e =$ maximum weight of a stable set in $G_j - v_j$, and $v_j^a =$ maximum weight of a stable set containing v_j in G_j . Having computed v_j^e and v_j^a for a separating vertex, we can replace $G_j - v_j$ by one vertex $v(G_j)$ of weight v_j^e since its best contribution to a maximum stable set of G , which in this case does not contain v_j , is equal to v_j^e . On the other hand, taking v_j in a stable set will imply that we add v_j^a vertices in a maximum stable set of G and that $v(G_j)$ cannot be taken in the same stable set. Consequently, the weight of the vertex v_j will be equal to v_j^a .

One can explore a whole cactus G repeating this procedure until we obtain a maximum stable set of G . Note that both for computing v_j^e and v_j^a , the problem with which we have to deal is exactly the maximum weighted stable set problem in a caterpillar, which is easily solved. Once an optimal solution is obtained, this maximum weight contains as information the set of vertices giving the corresponding maximum stable set. The number of calculations of this kind is equal to the number of separating vertices. The Algorithm StableCactus describes the application of the above idea in order to find a maximum stable set in a cactus. In figure 4, the method is represented for graph G of figure 3. At each step, only the weights that we need are shown in brackets. Note that the remaining graph after the application of Algorithm StableCactus is either a path or a cycle with an edge pending from one of its vertices, therefore the final computation of the algorithm is trivial.

Algorithm StableCactus

1. Set the weights of all vertices to 1;
2. Determine separating vertices and \mathcal{L} ; then construct an arborescence A ;
3. For any separating vertex v_j (considered in the exploration order of A) compute:

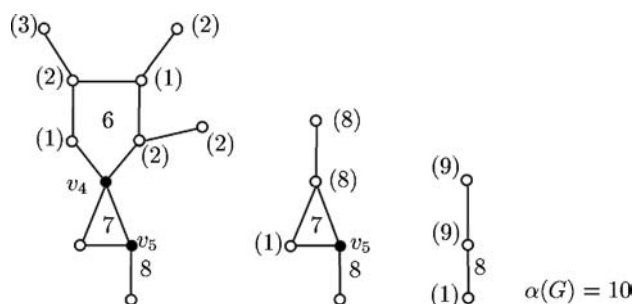


Figure 4. Computing the stability number of a cactus G .

- (a) v_j^e = maximum weighted stable set of G_j not containing v_j ;
- (b) v_j^a = maximum weighted stable set of G_j containing v_j ;
- (c) Replace G_j by a vertex x of weight v_j^e , assign the weight v_j^a to v_j , link x and v_j ;

4. Compute the weighted stability number of the remaining graph.

From the above discussion we have:

Lemma 1. *For a cactus G , $\alpha(G)$ can be found in time linear in $O(m+k)$, where k is the number of separating vertices in G .*

Theorem 4. *For a cactus G , an induced split graph of maximum size can be found in linear time.*

Proof: First of all, Algorithm StableCactus returns a maximum stable set S_{\max} giving $\alpha(G)$. Observe that we have $\alpha(G) + \omega(G) - 1 \leq \alpha_S(G) \leq \alpha(G) + \omega(G)$ for any graph. Having the list of all odd cycles \mathcal{L} , we search for a K_3^i in it which verifies $\alpha(G - K_3^i) = \alpha(G)$ for a maximum stable set S_{\max}^i of $G - K_3^i$. If such a pair (K_3^i, S_{\max}^i) exists, then it constitutes a split graph of maximum cardinality $\alpha_S(G) = \alpha(G) + 3$ in G . Otherwise any pair (K_3^i, S_{\max}^i) is a split graph of maximum cardinality $\alpha_S(G) = \alpha(G) + 2$. Whenever \mathcal{L} contains no K_3 , a clique of size 2 can be chosen in such a way that none of its vertices is contained in a maximum stable set S_{\max} of G . This claim is true since there would necessarily be two adjacent vertices of an odd cycle which do not figure in S_{\max} . Finally, if \mathcal{L} contains no odd cycles, then obviously $\alpha_S(G) = \alpha(G) + 1$. \square

4.1.2. Split-coloring in cacti. Having the list of all odd cycles of G , we will consider an auxiliary graph $G' = (V', E')$ where every vertex v'_i corresponds to an odd cycle OC^i of G , hence $|V'| = |\mathcal{L}| = L$. Two vertices v'_i and v'_j are linked if and only if the distance between OC^i and OC^j is at most 1.

Algorithm Cactus

1. Construct G' ;
2. If there is a stable set of size three in G' then $\chi_S(G) = 3$, otherwise:
 - (a) Choose v'_1 and v'_2 such that $v'_1 v'_2 \notin E'$ and set $K^{/1} = \{v'_1\}$, $K^{/2} = \{v'_2\}$ and $R = \emptyset$;
 - (b) For every vertex $v'_i \in V'$, if $v'_i v'_1 \in E'$ and $v'_i v'_2 \notin E'$ then store v'_i in $K^{/1}$, if $v'_i v'_2 \in E'$ and $v'_i v'_1 \notin E'$ then store v'_i in $K^{/2}$ else store v'_i in R ;
 - (c) For every vertex $v'_i \in R$, if v'_i is adjacent to every vertex in $K^{/1}$ (resp. in $K^{/2}$) then remove v'_i from R to $K^{/1}$ (resp. to $K^{/2}$);
 - (d) If $R = \emptyset$ then $K^{/1}$ and $K^{/2}$ are both cliques and G is 2-split-colorable;
 - (e) Otherwise $\exists v \in V'$ which constitutes a clique neither with $K^{/1}$ nor with $K^{/2}$; G is a OC_5 -cactus which is 3-split-colorable.

Theorem 5. *Algorithm Cactus decides in $O(L^3)$ time whether a cactus is 2 or 3-split-colorable.*

Proof: We first remark that the stability number of G' is the largest number of induced odd cycles in G . To be more precise, we are not interested in the exact value of $\alpha(G')$ because we know that $\alpha(G') \geq 3$ yields a 3-split-colorable cactus due to fact 3. So, a simple procedure which tests for every triplet in G' , in $O(L^3)$ time, whether it forms a stable set or not does the job. Having such a triplet, we may claim that our cactus is 3-split-colorable and otherwise we fix an arbitrary pair of non-adjacent vertices v'_1, v'_2 corresponding to two induced odd cycles in G .

Having two cliques covering the vertex set V' of G' , we can trivially split-color G with two colors. More precisely, the existence of a clique in G' implies that there is a set of odd cycles in G which are pairwise at distance at most 1 and that there is a clique in G of size 1, 2 or 3, which touches at least one vertex of each odd cycle of this set. In other words, these two cliques K'^1 and K'^2 in G' tell us exactly how to choose two cliques K^1 and K^2 in G , each one associated to a split graph. Then using fact 1, we may conclude that a partition of the vertex set V' into two cliques corresponds to a 2-split-coloring of G . That is why Algorithm Cactus searches for a partition into K'^1 and K'^2 that we would like to be cliques. According to this aim, K'^1 contains only vertices which are adjacent to v'_1 and so does K'^2 . We can see in figure 5 how these two cliques imply a 2-split-coloring.

In general, the phase of vertex-assignment yields two cliques K'^1 and K'^2 because of the fact that we have at most two induced odd cycles. There is only one case which is easily detected by Algorithm Cactus, where we do not obtain this result. It is due to the possible

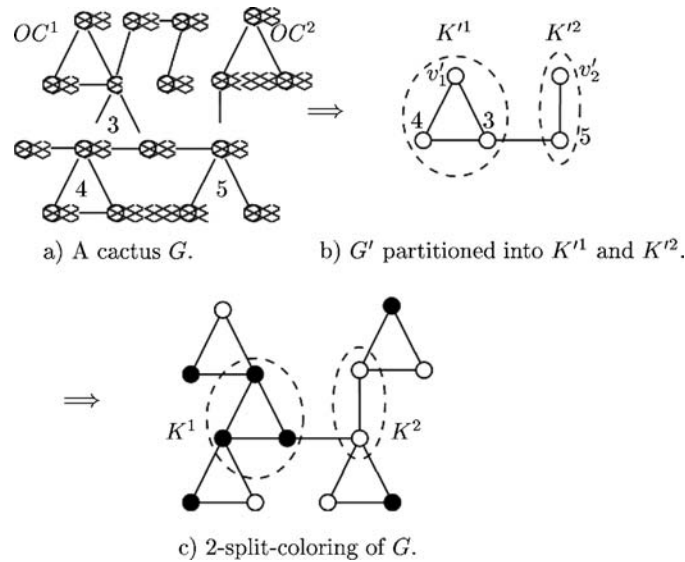


Figure 5. An example of split-coloring of a cactus.

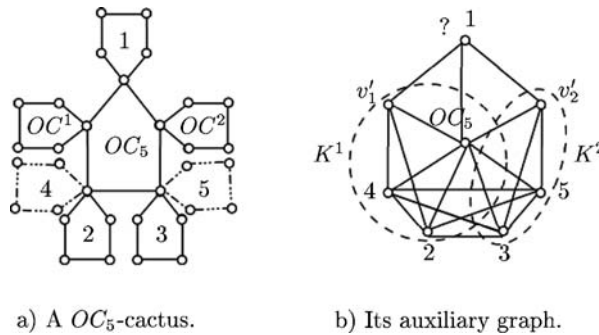


Figure 6. OC_5 -cacti are 3-split-colorable.

type of connection between two fixed induced odd cycles OC^1 and OC^2 : OC^1 and OC^2 linked by a OC_5 . This exception called a OC_5 -cactus, appears in figure 6 where odd cycles of length 5 around the central OC_5 can be indifferently replaced by odd cycles of any length. As shown in figure 6, odd cycles can be added in a way similar to odd cycles labeled 4 and 5, sharing any vertex of the central OC_5 . We observe that there is no way to partition the vertices of a OC_5 -cactus into two cliques. \square

It follows that a cactus G is 2-split-colorable if and only if there are at most two induced odd cycles and G is not a OC_5 -cactus; it will be 3-split-colorable otherwise.

Here we find another 3-split-critical structure in addition to the ones of Section 2: a OC_5 -cactus with unique odd cycles linked to each vertex of the central OC_5 is 3-split-critical.

4.2. Triangulated graphs

4.2.1. Split independence number in triangulated graphs

Theorem 6. For a triangulated graph $G = (V, E)$ where $|V| = n$ and $|E| = m$, the split-independence number $\alpha_S(G)$ can be obtained in $O(D(n + m))$ time where D is the number of maximum cliques in G .

Proof: We can enumerate all maximal cliques of a triangulated graph $G = (V, E)$ in $O(n + m)$ time (Tarjan and Yannakakis, 1985) exploiting the fact that a triangulated graph has a perfect elimination order which can start with any simplicial vertex (Dirac, 1961; Fulkerson and Gross, 1965). On the other hand, for a triangulated graph G , a maximum stable set can be found by a similar approach, again in $O(n + m)$ time (Gavril, 1972). It is easily seen that a maximum stable set and a maximum clique cannot share more than one vertex. Therefore, in the best case, a maximum split graph would have $\alpha(G) + \omega(G)$ vertices and otherwise $\alpha(G) + \omega(G) - 1$ vertices. It suffices to find a maximum stable set S'_i in $G - K^i$ where K^i is a maximum clique. It should be trivially noted that $\alpha(G) - 1 \leq |S'_i| \leq \alpha(G)$. If there exists S'_i such that $|S'_i| = \alpha(G)$ for some i , then S'_i is also a maximum stable set in G and $K^i \cup S'_i$

is a maximum split graph of size $\alpha_S(G) = \alpha(G) + \omega(G)$. Otherwise, we can conclude that there is no pair of maximum clique and maximum stable set having no shared vertex. In this case, any pair (K^i, S'_i) forms a maximum split graph of size $\alpha_S(G) = \alpha(G) + \omega(G) - 1$. To decide, we only run once the algorithm finding all maximal cliques in G and D times the algorithm finding a maximum stable set in $G - K^i$ where D is the number of maximum cliques. \square

4.2.2. Split-coloring in triangulated graphs

Theorem 7. *For a triangulated graph G , the split-coloring problem can be solved in $O(n^2(m + n))$ time.*

Proof: This follows from Hell et al. (2004) by observing that their algorithm finds, for a fixed k , a collection of k stable sets so that the remaining vertices can be covered by a minimum number l of cliques. Clearly, this partitioning gives a split-coloring of value $\max(k, l)$. It is obtained in $O(n(m + n))$ time. By repeating this algorithm for at most n different values of k , one can find a minimum split-coloring. \square

We would like to mention a final observation on arbitrary graphs. We notice that an approach similar to Turán's theorem (see Berge, 1983 Ch. 13) has no meaning in the framework of split-coloring problem. Turán's theorem deals with the question of finding a lower bound on the number of edges of a graph with n vertices and stability number $\alpha(G)$. The analogous problem for split-coloring would be to find a lower bound on the number of edges of a given graph G with split-independence number $\alpha_S(G)$. We observe that decreasing the number of edges in a graph does not increase systematically its split-independence number; it may happen that the removal of one edge may increase the split-independence number or decrease it. Let us show this on a very simple example. We take a C_4 as a graph having $\alpha_S(G) = 3$. Deleting any edge gives a P_4 with a split-independence number of 4. When we remove the edge in the middle of this P_4 , we obtain a $2K_2$ with again $\alpha_S(G) = 3$.

5. Bounds of χ_S in arbitrary graphs

5.1. A lower bound for χ_S

We have already mentioned at the end of Section 2 that the proof of Proposition 2 would serve as a hint to obtain a lower bound of χ_S in arbitrary graphs. Let us determine a procedure which gives an ordered set of induced maximal cliques in an arbitrary graph G .

Procedure CliquesList

1. Choose a vertex v and find a maximal clique K^v containing v ;
2. Store K^v in \mathcal{K} , then remove K^v and its neighbours from G ;
3. Repeat 1. and 2. until $G = \emptyset$;
4. Reorder $\mathcal{K} = \{K^i\}$ in non-increasing order of clique sizes.

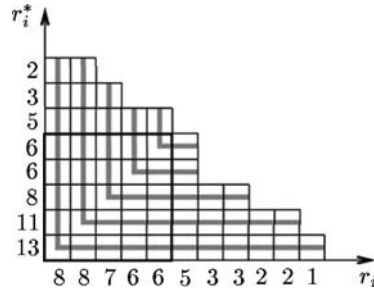


Figure 7. Optimal split-coloring of \mathcal{K} .

Procedure CliquesList returns a list \mathcal{K} of induced maximal cliques in G . In other words, there is no edge linking vertices of any pair of cliques in \mathcal{K} . Moreover, this list is maximal in the sense that adding any new maximal clique in \mathcal{K} introduces edges between cliques. An optimal split-coloring $\chi_S(\mathcal{K})$ of \mathcal{K} constitutes a lower bound for $\chi_S(G)$ since in the best case, G would be split-colored by the first $\chi_S(\mathcal{K})$ colors without any need of new colors; $\chi_S(\mathcal{K}) \leq \chi_S(G)$. Therefore, let us concentrate on the optimal split-coloring of a set of induced cliques of arbitrary sizes.

Our approach can be visualized in a diagram where cliques K^i are represented on the x axis as columns of length proportional to their cardinalities r_i . Respecting the non-increasing order of clique sizes while doing this implies that we can read on the y axis the number r_i^* of cliques K^k such that $r_k \geq i$. In this formulation, our problem consists in finding the largest k such that $\min(r_k, r_k^*) \geq k$. This amounts to finding the largest square that can be inserted under the stairs. We may use the following strategy: choose repetitively the largest remaining (not entirely colored) clique as the clique of a new split graph G_i (or color) and one vertex (not colored) from every other clique as the stable set of the same split graph. Each G_i is represented in figure 7 by a grey broken line (with breakpoint at entry (i, i)). Repeating this until no vertex remains uncolored gives rise to a split-coloring which uses exactly k colors. In figure 7 we have $\mathcal{K} = \{K_8^1, K_8^2, K_7^3, K_6^4, K_6^5, K_5^6, K_3^7, K_3^8, K_2^9, K_2^{10}, K_1^{11}\}$ and we obtain a 5-split-coloring which is optimal.

5.2. An upper bound for χ_S

Assume that we have a k -coloring (not necessarily an optimal coloring) of G given by stable sets (S^1, \dots, S^k) . For any $p \leq k$, an optimal clique cover of $(S^1 \cup \dots \cup S^p)$ together with an optimal coloring of $(S^{p+1} \cup \dots \cup S^k)$ constitutes a split-coloring of G in $\max\{\theta(S^1 \cup \dots \cup S^p), \chi(S^{p+1} \cup \dots \cup S^k)\}$ colors. Therefore, the split-chromatic number of G , $\chi_S(G)$, would be less than or equal to the minimum on p of this quantity. Furthermore, one can determine the minimum on any possible p -tuple of stable sets. Hence, the upper bound is expressed in the following way: $\chi_S(G) \leq \min_p \{\min_{S^1 \cup \dots \cup S^p} \{\max\{\theta(S^1 \cup \dots \cup S^p); \chi(G - (S^1 \cup \dots \cup S^p))\}\}\}$.

Beyond the theoretical interest of this upper bound, one can identify cases where it can be efficiently used in practice. First of all, if we can compute $\chi(G)$ in polynomial time then we will be able to take into account a least number of stable sets in searching for the

minimum on p . For instance, this is the case for perfect graphs. Furthermore, in this context, $\theta(S^{i^1} \cup \dots \cup S^{i^p})$ and $\chi(G - (S^{i^1} \cup \dots \cup S^{i^p}))$ can also be computed in polynomial time, providing an overall complexity of polynomial time to obtain an upper bound. One may search for conditions (on G or on the optimal coloring of G) for this upper bound to be tight. Note that, once more, we are more likely to find such a condition on perfect graphs.

6. Conclusion

A new type of coloring problem is introduced and some basic results are discussed. Several open questions arise from this work. For instance, for which classes of graphs can the split-coloring and/or maximum split graph problems be solved in polynomial time? On the other hand, it would be interesting to study the approximability properties of split-coloring problems. Development of techniques to solve the integer programming model of Section 3 is also a natural direction for further research.

References

- C. Benzaken, P. Hammer, and D. de Werra, "Split graphs of Dilworth number 2," *Discrete Mathematics*, vol. 55, pp. 123–127, 1985.
- C. Berge, *Graphes*, Bordas, Paris, 1983.
- A. Brandstädt, V. Le, and T. Szymczak, "The complexity of some problems related to graph 3-colorability," *Discrete Applied Mathematics*, vol. 89, pp. 59–73, 1998.
- Z.A. Chernyak and A. Chernyak, "Split dimension of graphs," *Discrete Mathematics*, vol. 89, pp. 1–6, 1991.
- D. de Werra, M. Demange, J. Monnot, and V. Paschos, "A hypocoloring model for batch scheduling," *Discrete Applied Mathematics*, vol. 146, pp. 3–26, 2005.
- G. Dirac, "On rigid circuit graphs," *Abh. Math. Sem. Univ. Hamburg*, no. 25, pp. 71–76, 1961.
- S. Földes and P. Hammer, "On split graphs and some related questions," in *Problèmes Combinatoires et Théorie des Graphes*, Orsay, France, Colloques Internationaux C.N.R.S. 260, 1976, pp. 139–140.
- S. Földes and P. Hammer, "Split graphs," *Congressum Numerantium*, vol. 19, pp. 311–315, 1977.
- D. Fulkerson and O. Gross, "Incidence matrixes and interval graphs," *Pacific Journal of Math.*, vol. 15, pp. 835–855, 1965.
- F. Gavril, "Algorithms for minimum coloring, maximum clique, minimum covering by cliques, and maximum independent set of a chordal graph," *SIAM J. Comput.*, vol. 1, pp. 180–187, 1972.
- E. Gourdin, M. Labbé, and H. Yaman, "Telecommunication and location," in *Facility location*, Springer, Berlin, 2002, pp. 275–305.
- P. Hammer and B. Simeone, "The splittance of a graph," *Combinatorica*, vol. 1, pp. 275–284, 1981.
- P. Hell, S. Klein, L. Nogueira, and F. Protti, "Partitioning chordal graphs into independent sets and cliques," *Discrete Applied Mathematics*, vol. 141, pp. 185–194, 2004.
- N. Mahadev and U. Peled, *Threshold Graphs and Related Topics*, Ann. Disc. Mat., North-Holland, vol. 56, 1995.
- R. Tarjan, "Depth first search and linear graph algorithms," *SIAM J. Comput.*, vol. 1, pp. 146–160, 1972.
- R.E. Tarjan and M. Yannakakis, "Addendum: Simple linear time algorithms to test chordality of graphs, test acyclicity of hypergraphs and selectively reduce acyclic hypergraphs," *SIAM J. Comput.*, vol. 14, no. 1, pp. 254–255, 1985.

