

# On the existence and determination of satisfactory partitions in a graph

Cristina Bazgan<sup>1</sup>, Zsolt Tuza<sup>2</sup>, and Daniel Vanderpooten<sup>1</sup>

<sup>1</sup> LAMSADE, Université Paris-Dauphine, France  
{bazgan,vdp}@lamsade.dauphine.fr

<sup>2</sup> Computer and Automation Institute, Hungarian Academy of Sciences, Budapest  
and Department of Computer Science, University of Veszprém, Hungary

**Abstract.** The SATISFACTORY PARTITION problem consists in deciding if a given graph has a partition of its vertex set into two nonempty sets  $V_1, V_2$  such that for each vertex  $v$ , if  $v \in V_i$  then  $d_{V_i}(v) \geq s(v)$ , where  $s(v) \leq d(v)$  is a given integer-valued function. This problem was introduced by Gerber and Kobler [*EJOR* 125 (2000), 283–291] for  $s = \lfloor \frac{d}{2} \rfloor$ . In this paper we study the complexity of this problem for different values of  $s$ .

**Keywords:** Satisfactory partition, graph, complexity, polynomial algorithm, *NP*-complete, degree constraints.

## 1 Introduction

Gerber and Kobler introduced in [3] the problem of deciding if a given graph has a vertex partition into two nonempty sets such that each vertex has at least as many neighbors in its set as in the other. The complexity of this problem remains open in their paper. They showed the strong *NP*-hardness of a generalization of this problem where there are weights on the vertices and we ask for a vertex partition into two nonempty sets such that for each vertex the sum of weights of the neighbors in the same set is at least as large as the sum of weights of the neighbors in the other set. The case where edges are weighted was also proved to be strong *NP*-hard.

For a graph  $G = (V, E)$ , vertex  $v \in V$ , and subset  $Y \subseteq V$  we denote by  $d_Y(v)$  the number of vertices in  $Y$  that are adjacent to  $v$ ; and, as usual, we write  $d(v)$  for the degree  $d_V(v)$  of  $v$  in  $V$ . Throughout, the subgraph of graph  $G = (V, E)$  induced by  $Y \subseteq V$  will be denoted by  $G[Y]$ . The general problem we are interested in is as follows:

SATISFACTORY PARTITION

**Input:** A graph  $G = (V, E)$ , and a function  $s : V \rightarrow \mathbb{N}$  such that  $s(v) \leq d(v)$ , for all  $v \in V$ .

**Question:** Is there a nontrivial partition  $(V_1, V_2)$  of  $V$  such that, for every  $v \in V$ , if  $v \in V_i$  then  $d_{V_i}(v) \geq s(v)$  ?

Considering  $A \subset V$ , a vertex  $v \in A$  is *satisfied* in  $A$  if  $d_A(v) \geq s(v)$ . Moreover  $A$  is a *satisfactory subset* if all of its vertices are satisfied in  $A$ . If  $A, B \subseteq V$  are two disjoint, nonempty vertex subsets such that  $A$  and  $B$  are satisfactory subsets, we say that  $(A, B)$  is a *satisfactory pair*. If, in addition,  $(A, B)$  is a vertex partition then it will be called a *satisfactory partition* and a graph admitting such a partition is said to be *partitionable*.

In our statement of the SATISFACTORY PARTITION problem, the function  $s$  indicates the level of satisfaction required for the vertices to be satisfied. The problem studied by Gerber and Kobler corresponds to  $s = \lceil \frac{d}{2} \rceil$ . As remarked in [3], this problem may have no solution. In particular, the following graphs are not partitionable: complete graphs, stars, and complete bipartite graphs with at least one of the two vertex classes having odd size. Some other graphs are easily partitionable: cycles of size at least 4, trees which are not stars, and disconnected graphs.

After stating some preliminary results (Section 2), we study in Sections 3-5 SATISFACTORY PARTITION for different values of  $s$ . For  $s \leq \lceil \frac{d}{2} \rceil - 1$ , a result of Stiebitz [5] indicates that the graph always has such a partition. The original proof is not constructive, and we give here a polynomial-time algorithm that finds such a partition (Section 3). For  $s = \lceil \frac{d}{2} \rceil$ , we prove that for graphs with maximum degree at most 4, both the decision and search problems are polynomially solvable. This problem on general graphs remains open (Section 4). For  $\lceil \frac{d}{2} \rceil + 1 \leq s \leq d - 1$ , SATISFACTORY PARTITION is proved to be *NP*-complete (Section 5). For  $s = d$ , the problem is trivial since it consists in deciding whether the graph is disconnected.

## 2 Preliminary results

Firstly we establish a necessary and sufficient condition to obtain a satisfactory partition.

**Proposition 1.** *When  $s \leq \lceil \frac{d}{2} \rceil$ , a graph  $G = (V, E)$  is partitionable if and only if it contains a satisfactory pair  $(A, B)$ . Moreover, if a satisfactory pair  $(A, B)$  is given, then a satisfactory partition of  $G$  can be determined in polynomial time.*

*Proof.* The necessary part is obvious. The sufficient part is proved as follows. Let  $V_1 = A$  and  $V_2 = B$ . While there is a vertex  $v$  in  $V \setminus (V_1 \cup V_2)$  such that  $d_{V_1}(v) \geq s(v)$ , insert  $v$  into  $V_1$ . While there is a vertex  $v$  in  $V \setminus (V_1 \cup V_2)$  such that  $d_{V_2}(v) \geq s(v)$ , insert  $v$  into  $V_2$ . At the end, if  $C = V \setminus (V_1 \cup V_2) \neq \emptyset$ , then  $d_{V_1}(v) < s(v)$  and  $d_{V_2}(v) < s(v)$  for any  $v \in C$ . Since  $s(v) \leq \lceil \frac{d(v)}{2} \rceil$ , we have, for any  $v \in C$ ,  $d_{V_1 \cup C}(v) \geq s(v)$  and  $d_{V_2 \cup C}(v) \geq s(v)$ . Thus we can insert all vertices of  $C$  either into  $V_1$  or into  $V_2$ , forming a satisfactory partition.  $\square$

We show now how to determine satisfactory subsets efficiently.

**Proposition 2.** *For any  $s \leq d$ , it is decidable in polynomial time, if a subgraph  $G[A]$  of a graph  $G$  contains a satisfactory subset.*

*Proof.* The algorithm iteratively removes from  $G[A]$  the vertices  $v$  of degree less than  $s(v)$ , while it is possible.

If at the end of the algorithm we obtain a non-empty subgraph  $G[A']$ , then  $G[A]$  contains a satisfactory subset  $A'$  since  $d_{A'}(v) \geq s(v)$ , for all  $v \in A'$ .

Conversely, suppose that the algorithm removes all vertices from  $A$  and that  $G[A]$  would contain a satisfactory subset  $A'$ . The first vertex  $v$  of  $A'$  considered by the algorithm cannot be removed since its current degree is greater than or equal to  $d_{A'}(v) \geq s(v)$ . In this way no vertex of  $A'$  could be removed. Thus, if the entire set  $A$  gets deleted, then  $G[A]$  does not contain a satisfactory subset.  $\square$

A *minimal satisfactory subset*  $A$  is a satisfactory subset such that, for every  $A' \subset A$ , there exists a vertex  $u \in A'$  with  $d_{A'}(u) \leq s(u) - 1$ .

**Proposition 3.** *For any  $s \leq d$ , if a graph contains a satisfactory subset, then a minimal satisfactory subset  $A$  can be found in polynomial time.*

*Proof.* Let  $A_1$  be a satisfactory subset of  $G$ . We construct a sequence of subgraphs of  $G$ ,  $G[A_1], \dots, G[A_t]$  such that  $A_{i+1} \subset A_i$  and each  $A_i$  is a satisfactory subset:

In step  $i$  ( $i \geq 1$ ), we select a vertex  $v \in A_i$ , tentatively remove from  $A_i$  vertex  $v$  and we iteratively remove from  $A_i \setminus \{v\}$  the vertices  $u$  of degree less than  $s(u)$  until we obtain a set  $X$  which is either empty or a satisfactory subset. If  $X = \emptyset$ , then we iterate the previous procedure for vertices  $v \in A_i$  until a set  $X \neq \emptyset$  is obtained, and in this case we continue the construction with  $A_{i+1} = X$ . In the other case, i.e. where all the sets  $X$  obtained are empty, we stop the algorithm with  $t = i$ . The set  $A$  with the required properties is the set  $A_t$ .

It is clear that  $A$  is a satisfactory subset. We have to prove that  $A$  is minimal. Suppose on the contrary, that there exists a subset  $A' \subset A$  such that  $d_{A'}(v) \geq s(v)$  for all  $v \in A'$ . Choose any  $u \in A \setminus A'$ . Continuing the procedure from  $A_t = A$  by removing  $u$ , a nonempty set  $A_{t+1} \supseteq A'$  would be generated, which contradicts the previous assumption that  $A_{t+1} = \emptyset$  holds for all  $u \in A$ .  $\square$

### 3 The case $s \leq \lceil \frac{d}{2} \rceil - 1$

Stiebitz [5] proved the following result:

**Theorem 1 ([5]).** *Let  $G = (V, E)$  be a graph, and let  $a, b : V \rightarrow \mathbb{N}$  be two functions such that  $d(x) \geq a(x) + b(x) + 1$  for every  $x \in V$ . Then, there is a nontrivial vertex partition  $(A, B)$  of  $V$  such that  $d_A(x) \geq a(x)$  for every  $x \in A$ , and  $d_B(x) \geq b(x)$  for every  $x \in B$ .*

Obviously, if we take  $a = b = s$ , Stiebitz's result shows the existence of a satisfactory partition for  $s \leq \lceil \frac{d}{2} \rceil - 1$ . The proof in [5] is not constructive. We show here how to determine such a partition in polynomial time.

**Proposition 4.** *Algorithm 1 finds in polynomial time a satisfactory pair in any graph for  $s \leq \lceil \frac{d}{2} \rceil - 1$ .*

---

**Algorithm 1** Determination of a satisfactory pair ( $s \leq \lceil \frac{d}{2} \rceil - 1$ )

---

Find a minimal satisfactory subset  $A \subseteq V$   
 $B \leftarrow V \setminus A$   
**if**  $G[B]$  contains a satisfactory subset **then**  
     $(A, \tilde{B})$  is a satisfactory pair,  $\tilde{B}$  being a satisfactory subset of  $G[B]$   
**else**  
    **while** ( $G[A]$  or  $G[B]$  has no satisfactory subset) **do**  
        **if**  $G[B]$  has no satisfactory subset **then**  
            Let  $v \in A$  such that  $d_A(v) \leq s(v)$   
             $A \leftarrow A \setminus \{v\}; B \leftarrow B \cup \{v\}$   
        **end if**  
        **if**  $G[A]$  has no satisfactory subset **then**  
            Let  $v \in B$  such that  $d_B(v) \leq s(v)$   
             $A \leftarrow A \cup \{v\}; B \leftarrow B \setminus \{v\}$   
        **end if**  
    **end while**  
     $(\tilde{A}, \tilde{B})$  is a satisfactory pair where  $\tilde{A}$  and  $\tilde{B}$  are satisfactory subsets of  $G[A]$  and  $G[B]$ , respectively  
**end if**

---

*Proof.* Using Proposition 3, the first step is computable in polynomial time. Using Proposition 2, the **if** and **while** conditions are polynomial-time decidable. We justify that in the **while** loop the selection of a vertex  $v$  is always possible. For this, we show that at the beginning of each iteration in the **while** loop,  $G[A]$  and  $G[B]$  are such that all their subgraphs contain a vertex  $u$  of degree at most  $s(u)$ ; and in addition at least one of  $G[A]$  and  $G[B]$  is such that all their subgraphs contain a vertex  $u$  of degree at most  $s(u) - 1$ . In fact, before entering the **while** loop, each subgraph of  $G[A]$  contains a vertex  $u$  of degree at most  $s(u)$ , since  $A$  is a minimal satisfactory subset. Also, before entering the **while** loop,  $G[B]$  has no satisfactory subset, which means that each of its subgraphs contains a vertex  $u$  of degree strictly smaller than  $s(u)$ . At the end of an iteration of the **while** loop after moving  $v$  from  $A$  to  $B$  for example, the degree of vertices in  $G[B \cup \{v\}]$  increases with at most one, so in each subgraph of  $G[B \cup \{v\}]$ , there is a vertex  $u$  of degree at most  $s(u)$ . Since we are inside the **while** loop only if one of the graphs  $G[A]$  and  $G[B]$  is such that none of the subgraphs has a satisfactory subset, the corresponding graph has in each subgraph a vertex  $u$  of degree at most  $s(u) - 1$  that can increase to at most  $s(u)$  after moving vertex  $v$  into the other vertex class. Hence, the operations inside the loop can always be performed.

We show now that the number of iterations is polynomially bounded. Consider any iteration of the **while** loop. Assume, without loss of generality, that  $G[B]$  has no satisfactory subset. By the choice of  $v$ , since  $d_A(v) \leq s(v)$ , we have  $d_B(v) \geq s(v) + 1$ . Thus, the number of edges between  $A$  and  $B$  decreases by at least one and thus the algorithm finishes after at most  $|E|$  iterations.  $\square$

**Theorem 2.** SATISFACTORY PARTITION for  $s \leq \lceil \frac{d}{2} \rceil - 1$  is polynomial-time solvable.

*Proof.* From Propositions 1 and 4. □

By slightly modifying Algorithm 1 replacing  $s$  by  $a$  or  $b$  in appropriate places we obtain the following result:

**Theorem 3.** Let  $G = (V, E)$  be a graph, and let  $a, b : V \rightarrow \mathbb{N}$  be two functions such that  $d(x) \geq a(x) + b(x) + 1$  for every  $x \in V$ . Then, we can find in polynomial time a nontrivial vertex partition  $(A, B)$  of  $V$  such that  $d_A(x) \geq a(x)$  for every  $x \in A$ , and  $d_B(x) \geq b(x)$  for every  $x \in B$ .

#### 4 The case $s = \lceil \frac{d}{2} \rceil$

In this section we show that for graphs  $G$  with  $\Delta(G) \leq 4$  it is polynomial-time solvable to decide if the graph is (not) partitionable, and also to find a satisfactory partition if it exists. In particular, all cubic graphs except  $K_4$  and  $K_{3,3}$  are partitionable and all 4-regular graphs except  $K_5$  are partitionable.

Firstly we prove two propositions.

**Proposition 5.** Each cubic graph containing a triangle, except  $K_4$ , is partitionable.

*Proof.* Let  $G$  be a cubic graph,  $G \neq K_4$ , and let  $C$  be a triangle of  $G$  with vertices  $v_1, v_2, v_3$ . Remark that a vertex outside  $C$  cannot have all its neighbors on  $C$  since  $G \neq K_4$ .

If each vertex of  $V \setminus V(C)$  has at most one neighbor on  $C$  then  $V_1 = V(C)$  and  $V_2 = V \setminus V(C)$  form a satisfactory partition.

Suppose that there is a vertex  $v_4$  with two neighbors  $v_1, v_2$  on  $C$ . If  $v_3$  and  $v_4$  have a common neighbor  $v_5$ , then  $V_1 = \{v_1, v_2, v_3, v_4, v_5\}$  and  $V_2 = V \setminus V_1 \neq \emptyset$  form a satisfactory partition of  $G$ . Otherwise  $V_1 = \{v_1, v_2, v_3, v_4\}$  and  $V_2 = V \setminus V_1 \neq \emptyset$  form a satisfactory partition of  $G$ . □

**Proposition 6.** Each cubic graph containing a cycle of size 4, except  $K_4$  and  $K_{3,3}$ , is partitionable.

*Proof.* Let  $G$  be a cubic graph other than  $K_4$  and  $K_{3,3}$ . If  $G$  contains a triangle then  $G$  is partitionable by Proposition 5. Otherwise let  $C = v_1 v_2 v_3 v_4$  be a cycle of size 4. A vertex outside  $C$  cannot have more than two neighbors on  $C$  since otherwise  $G$  contains a cycle shorter than  $C$ .

If each vertex of  $V \setminus V(C)$  has at most one neighbor on  $C$ , then  $V_1 = V(C)$  and  $V_2 = V \setminus V(C)$  form a satisfactory partition.

Otherwise, suppose that a vertex  $v_5$  has neighbors  $v_1$  and  $v_3$ . Since  $G \neq K_{3,3}$  there is no vertex of  $G$  with the three neighbors  $v_2, v_4, v_5$ . Thus, a vertex  $v_i$  with  $i \geq 6$  has at most two neighbors among  $\{v_2, v_4, v_5\}$ . If all vertices  $v_i$  with  $i \geq 6$  have at most one neighbor among  $\{v_2, v_4, v_5\}$  then  $V_1 = \{v_1, v_2, v_3, v_4, v_5\}$

and  $V_2 = V \setminus V_1 \neq \emptyset$  form a satisfactory partition of  $G$ . Otherwise, let  $v_6$  be a vertex that has  $v_2, v_4$  as neighbors. If all vertices  $v_i$  with  $i \geq 7$  have at most one neighbor among  $\{v_5, v_6\}$ , then  $V_1 = \{v_1, v_2, v_3, v_4, v_5, v_6\}$  and  $V_2 = V \setminus V_1 \neq \emptyset$  form a satisfactory partition of  $G$ . Otherwise, there is another vertex  $v_7$  with neighbors  $v_5, v_6$ . In this case  $V_1 = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$  and  $V_2 = V \setminus V_1 \neq \emptyset$  form a satisfactory partition of  $G$ .  $\square$

We show now how to determine a satisfactory partition.

---

**Algorithm 2** Determination of a satisfactory partition for cubic and 4-regular graphs

---

```

Let  $G$  be an  $(\ell + 1)$ -regular graph ( $\ell = 2$  or  $3$ ).
Search a shortest cycle  $C$ .
if  $|C| \geq 5$  then
     $V_1 \leftarrow V(C)$ ;  $V_2 \leftarrow V \setminus V(C)$ 
else
     $V_1 \leftarrow V(C)$ 
    while there exists a vertex  $v \in V \setminus V_1$  with at least  $\ell$  neighbors in  $V_1$  do
         $V_1 \leftarrow V_1 \cup \{v\}$ 
    end while
     $V_2 \leftarrow V \setminus V_1$ 
end if

```

---

**Theorem 4.** *All cubic graphs except  $K_4$  and  $K_{3,3}$  are partitionable in polynomial time.*

*Proof.* Let  $G$  be a cubic graph,  $G \neq K_4$  and  $K_{3,3}$ . Let us verify that Algorithm 2 with  $\ell = 2$  is correct.

If  $|C| = k \geq 5$ , then there are no two vertices on  $C$  with a common neighbor  $v$  outside  $C$ , since otherwise there exists in  $G$  a cycle of length at most  $\lfloor k/2 \rfloor + 2$ . For  $k \geq 5$  this would be a cycle shorter than  $C$ . So, each vertex outside  $C$  has at least two neighbors among  $V \setminus V(C)$  and thus  $V_1 = V(C)$  and  $V_2 = V \setminus V(C) \neq \emptyset$  form a partition where each vertex is satisfied.

If  $|C| \leq 4$ , then the proofs of the above propositions show that in the partition  $(V_1, V_2)$  each vertex is satisfied.  $\square$

**Theorem 5.** *All 4-regular graphs except  $K_5$  are partitionable, in polynomial time.*

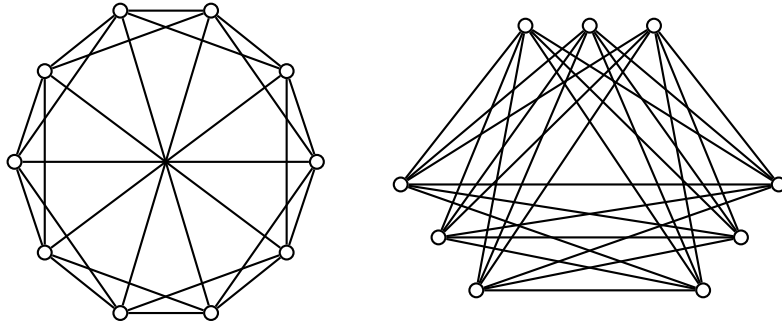
*Proof.* Let us see in the following that Algorithm 2 with  $\ell = 3$  is correct.

If  $|C| \geq 5$ , then as above,  $V_1 = V(C)$  and  $V_2 = V \setminus V(C) \neq \emptyset$  form a satisfactory partition.

If  $|C| = 4$ , then there is no vertex outside  $C$  with three neighbors on  $C$ , and thus  $V_1 = V(C)$  and  $V_2 = V \setminus V(C) \neq \emptyset$  form a satisfactory partition.

If  $|C| = 3$ , then denote  $C = v_1v_2v_3$ . If each vertex of  $V \setminus V(C)$  has at most two neighbors on  $C$ , then  $G$  is partitionable, and  $V_1 = V(C)$  and  $V_2 = V \setminus V(C)$  form a satisfactory partition. Otherwise let  $v_4$  be a vertex with neighbors  $v_1, v_2, v_3$ . If each vertex  $v_i, i \geq 5$  has at most two neighbors among  $v_1, v_2, v_3, v_4$  then  $G$  is partitionable. Otherwise, since  $G \neq K_5$ , let  $v_5$  be a vertex with three neighbors among  $v_1, v_2, v_3, v_4$ . Then  $V_1 = \{v_1, v_2, v_3, v_4, v_5\}$  and  $V_2 = V \setminus V_1 \neq \emptyset$  form a satisfactory partition of  $G$ .  $\square$

Thus, all cubic graphs except  $K_4$  and  $K_{3,3}$  are partitionable and all 4-regular graphs except  $K_5$  are partitionable. These results cannot be extended for regular graphs with degree greater than 4 since there are 5-regular graphs, different from  $K_6$  and  $K_{5,5}$  that are not partitionable, and there are 6-regular graphs different from  $K_7$  that are not partitionable (see Figure 1).



**Fig. 1.** Non-partitionable 5-regular and 6-regular graphs

We consider now graphs with maximum degree at most 4. As usual, the minimum and maximum degree of  $G$  will be denoted by  $\delta(G)$  and  $\Delta(G)$ , respectively.

**Proposition 7.** *A graph  $G$  with  $\delta(G) = 3$  and  $\Delta(G) \leq 4$  is partitionable if and only if it contains two vertex-disjoint cycles.*

*Proof.* (If) Immediate from Proposition 1. (Only if) If  $G$  is partitionable then each vertex has at least two neighbors in its part, so each part contains a cycle.  $\square$

**Proposition 8.** *Let  $G$  be a graph with  $\Delta(G) \leq 4$ . Then  $G$  is partitionable if and only if we can add in  $G$  at most two disjoint edges between vertices of degrees 1 or 2, such that the resulting multigraph contains two vertex-disjoint cycles.*

*Proof.* (If) If  $G$  contains two disjoint cycles  $C_1, C_2$  then  $V(C_1)$  and  $V(C_2)$  can be completed to form a satisfactory partition, using Proposition 1.

If  $G$  has no two disjoint cycles but adding one edge  $(v_i, v_j)$  the graph  $G' = (V, E \cup \{(v_i, v_j)\})$  has two disjoint cycles  $C_1, C_2$  then  $(v_i, v_j)$  belongs to one of

these cycles. Then  $V(C_1)$  and  $V(C_2)$  form a satisfactory pair once we remove  $(v_i, v_j)$  since  $v_i$  and  $v_j$  have degree at most two.

Assume now that the addition of two non-adjacent edges  $(v_i, v_j), (v_k, v_\ell)$  is such that the new graph contains two disjoint cycles. Since these two edges are not adjacent, as above, the two disjoint cycles can be completed to a satisfactory partition.

(Only if) Let  $(V_1, V_2)$  be a satisfactory partition of  $G$ . If  $V_i$  ( $i = 1, 2$ ) contains no cycle, then we add one edge between two degree-1 vertices of a tree component inside  $V_i$ . If the tree in question is just an edge, then we add a parallel edge creating a multiple edge.  $\square$

**Theorem 6.** *Let  $G$  be a graph with  $\Delta(G) \leq 4$ . We can decide in polynomial time if  $G$  is (not) partitionable, and find a satisfactory partition of  $G$  if it exists.*

*Proof.* There is a polynomial number of choices to add at most two non-adjacent edges in  $G$ . For a fixed choice, we first verify if there are multiple edges. If there are two non-adjacent multiple edges, then we have found two disjoint cycles; if there is one multiple edge, then we search a cycle in the graph obtained by removing the two vertices incident to this edge. The graph is partitionable if and only if such a cycle exists. If the graph has no multiple edges, then we apply a polynomial algorithm that finds two disjoint cycles in a graph if they exist (Bodlaender [1]), to decide if the graph is partitionable.  $\square$

## 5 The case $\lceil \frac{d}{2} \rceil + 1 \leq s \leq d - 1$

Chvátal introduced in [2] the decomposition problem of bicoloring the vertices of a graph in such a way that each vertex has at most one neighbor with a different color. He gave a polynomial-time algorithm for this problem for graphs with maximum degree 3. For graphs with vertices of degree 2 and 3 this problem coincides with SATISFACTORY PARTITION when  $s = \lceil \frac{d}{2} \rceil$ . Theorem 6 contains this result. Chvátal also proved the *NP*-hardness of this problem for graphs with minimum degree  $\delta(G) = 3$  and maximum degree  $\Delta(G) = 4$ . This result implies the following result for our problem.

**Theorem 7.** *SATISFACTORY PARTITION is *NP*-complete when  $s = d - 1$ , even for graphs with  $\delta(G) = 3$  and  $\Delta(G) = 4$ .*

Observe that Chvátal's problem coincides with SATISFACTORY PARTITION when  $s = \lceil \frac{d}{2} \rceil + 1$  or  $s = d - 1$  for graphs  $G$  with  $\delta(G) = 4$  and  $\Delta(G) = 5$ . In order to prove the *NP*-hardness of SATISFACTORY PARTITION when  $\lceil \frac{d}{2} \rceil + 1 \leq s \leq d - 1$ , we just need to adapt Chvátal's construction using a graph with  $\delta(G) = 4$  and  $\Delta(G) = 5$ .

**Theorem 8.** *SATISFACTORY PARTITION for  $\lceil \frac{d}{2} \rceil + 1 \leq s \leq d - 1$  is *NP*-complete even for graphs  $G$  with  $\delta(G) = 4$  and  $\Delta(G) = 5$ .*



Before proving this theorem, we define a problem used in the reduction.

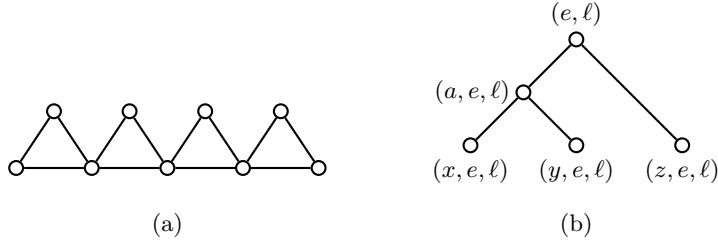
**BICOLORING HYPERGRAPHS**

**Input:** A hypergraph  $H = (V, E)$ .

**Question:** Is there a coloring with two colors Red and Blue of the vertices such that no hyperedge is monochromatic?

BICOLORING HYPERGRAPHS is *NP*-hard even if all hyperedges have size 3 ([4]).

*Proof.* The reduction is from BICOLORING HYPERGRAPHS with hyperedges of size 3. Given an input hypergraph  $H = (V, E)$  where  $V = \{v_1, \dots, v_n\}$  and  $E = \{e_1, \dots, e_m\}$ , we construct a graph  $G = (V', E')$  such that  $H$  is bicolorable if and only if  $G$  is partitionable. We first briefly describe Chvátal's construction, which uses the following graph  $Q_d$  that is a sequence of  $d$  triangles (see Figure 2 (a)).



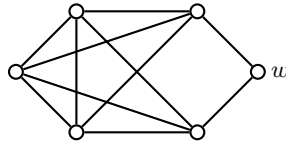
**Fig. 2.** (a) Graph  $Q_d$  for  $d = 4$ ; (b) The gadget  $T_{e,\ell}$

Graph  $G$  contains two subgraphs  $B_0, B_1$  and subgraphs  $A_i$  for each  $v_i \in V$ .  $B_0$  and  $B_1$  are graphs  $Q_{n+m-2}$  and  $A_i$  is  $Q_{2d_i-1}$ , where  $d_i$  is the number of hyperedges of  $H$  containing  $v_i$ . In each  $B_\ell, \ell = 0, 1$  the first  $n$  consecutive vertices of degree 2 are labeled by  $(v_1, \ell), \dots, (v_n, \ell)$ ; the remaining  $m$  consecutive vertices of degree 2 are labeled by  $(e_1, \ell), \dots, (e_m, \ell)$ . The  $2d_i + 1$  vertices of degree 2 in  $A_i$  are labeled by  $v_i^*$  and  $(v_i, e_j, 0), (v_i, e_j, 1)$  if  $v_i$  belongs to hyperedge  $e_j$ . Another gadget used by Chvátal is the graph of Figure 2 (b). For each  $\ell = 0, 1$  and for each hyperedge  $e$  of  $H$  containing vertices  $x, y, z$ , we add the graph  $T_{e,\ell}$ . This graph has the property that if the three vertices of degree 1 are in the same part of a satisfactory partition then the other two vertices of  $T_{e,\ell}$  are in the same part. Also, if two of the three vertices of degree 1 are in different parts of a satisfactory partition then the vertex of degree 2 could be in either of the two parts of the partition.

We specify in the following the edges in  $G$  that link the subgraphs  $A_1, \dots, A_n, B_0, B_1$ . Each  $v_i^*$  is joint with  $(v_i, 0)$  and  $(v_i, 1)$  for  $i = 1, \dots, n$ . Subgraphs  $B_0$  and  $B_1$  are joined to subgraphs  $A_j, j = 1, \dots, n$  using gadgets  $T_{e,\ell}, e \in E$  and  $\ell = 0, 1$ , by identifying vertices with the same labels. With this construction Chvátal proves the *NP*-hardness of this decomposition problem.

We introduce the gadget  $C_{5,w}$  (see Figure 3) in order to transform any vertex of the graph of Chvátal's construction of degree 2 or 3 to a vertex of degree 4 or

5 while preserving the reduction. The property of this gadget is that its vertices are necessarily in the same part of a satisfactory partition.



**Fig. 3.** The gadget  $C_{5,w}$

We identify vertex  $w$  of such a gadget  $C_{5,w}$  with vertices  $(v_i, 0)$  and  $(v_i, 1)$  for  $i = 1, \dots, n$  and with vertices  $(v_i, e_j, 0)$  and  $(v_i, e_j, 1)$  of the graphs  $A_i$ . Also, we identify vertex  $w$  of such a gadget  $C_{5,w}$  with vertices  $(a, e, \ell)$  of gadgets  $T_{e,\ell}$ .

We justify in the following that  $H$  is bicolorable if and only if  $G'$  is partitionable.

Suppose firstly that  $G'$  is partitionable. It is easy to see that all vertices in  $B_0$  must belong to the same part of the satisfactory partition. The same property holds for  $B_1$  and also for  $A_i$  for  $v_i \in V$ . Subgraphs  $B_0$  and  $B_1$  must be in the different parts of the satisfactory partition since otherwise all vertices of  $G$  would be in the same part. We construct a vertex bicoloring of  $H$  from this partition as follows: if  $A_i$  is in the same part as  $B_0$  then  $v_i$  is colored Red, and if it is in the same part as  $B_1$  then  $v_i$  is colored Blue. Given a hyperedge  $e$  of  $H$  containing vertices  $x, y, z$ , it is easy to see using the properties of  $T_{e,\ell}$  that  $e$  cannot be monochromatic, since otherwise  $(e, 0)$  and  $(e, 1)$  would be in the same part of the partition.

Suppose that  $H$  is bicolorable. All vertices of  $A_i$  in  $G$  corresponding to vertices  $v_i$  in  $H$  of the same color belong to the same part. This partition can be extended to a satisfactory partition.  $\square$

## References

1. H. L. Bodlaender, *On disjoint cycles*, International Journal of Foundations of Computer Science 5(1) (1994), 59–68.
2. V. Chvátal, *Recognizing decomposable graphs*, Journal of Graph Theory 8 (1984), 51–53.
3. M. Gerber and D. Kobler, *Algorithmic approach to the satisfactory graph partitioning problem*, European Journal of Operation Research, 125 (2000), 283–291.
4. L. Lovász, *Coverings and coloring of hypergraphs*, In Proceedings of 4th Southeastern Conference on Combinatorics, Graph Theory and Computing, Utilitas Mathematica, Winnipeg, 1973, 3–12.
5. M. Stiebitz, *Decomposing graphs under degree constraints*, Journal of Graph Theory 23 (1996), 321–324.