Advances on the List Stubborn Problem

Simone Dantas[1] **Luerbio Faria**[2] **Celina M. H. de Figueiredo**[3]
Sulamita Klein[4] **Loana T. Nogueira**[5] **Fábio Protti**[5]

[1] Instituto de Matemática
Universidade Federal Fluminense (UFF), Brazil
E-mail:sdantas@im.uff.br

[2] Faculdade de Formação de Professores
Universidade do Estado do Rio de Janeiro (UERJ), Brazil
E-mail:luerbio@cos.ufrj.br

[3] Programa de Engenharia de Sistemas e Computação - COPPE
Universidade Federal do Rio de Janeiro (UFRJ), Brazil
E-mail:celina@cos.ufrj.br

[4] Instituto de Matemática and Programa de Engenharia de Sistemas e Computação - COPPE
Universidade Federal do Rio de Janeiro (UFRJ), Brazil
E-mail:sula@cos.ufrj.br

[5] Instituto de Computação
Universidade Federal Fluminense (UFF), Brazil
E-mail:{loana, fabio}@ic.uff.br

Abstract

The 4-PARTITION problem is defined as partitioning the vertex set of a graph G into at most 4 parts A, B, C, D, where each part is not required to be nonempty, and is a stable set, a clique, or has no restriction; and pairs of distinct parts are completely nonadjacent, completely adjacent, or arbitrarily adjacent. The LIST 4-PARTITION problem generalizes the 4-partition problem by specifying for each vertex x, a list $L(x)$ of parts in which x is allowed to be placed. The only LIST 4-PARTITION problem not classified as either polynomial time solvable or NP-complete is the LIST STUBBORN problem (up to complementarity): A and B are stable sets, D is a clique, each vertex of A is nonadjacent to each vertex of C. We polynomially reduce the general LIST STUBBORN instance to a particular instance with a structured graph and only two types of lists. Additionally, we show that this particular LIST 4-PARTITION problem is polynomially equivalent to a nonlist problem, named TWOFOLD STUBBORN problem.

Keywords: Complexity, graph theory, matrix partitions, list partitions.

1 Introduction

Consider a partition of the vertices of a given graph into subsets satisfying constraints internally (a set may be required to be stable or a clique) and externally (two sets may be required to be completely nonadjacent or completely adjacent). For a given symmetric $k \times k$ matrix M over $\{0, 1, *\}$, an *M-partition* is a partition of the vertex set into at most k parts $A_1, A_2, \ldots A_k$, corresponding to the rows and columns of M, such that: for $i \neq j$, if $M_{ij} = 0$ (resp., 1, $*$), then 'no edges' (resp., 'all edges', 'no restriction') are required between vertices in part i and vertices in part j; if $M_{ii} = 0$ (resp., 1, $*$), then part i is required to be a stable set (resp., clique, arbitrary subgraph). Feder, Hell, Klein and Motwani (2003) introduced the M-PARTITION problem and generalized it to the LIST M-PARTITION problem, where additionally to being given a graph G and a symmetric $k \times k$ matrix M over $\{0, 1, *\}$, for each vertex v of G, we are given a list $L(v)$ that is a nonempty subset of $\{A_1, A_2, \ldots, A_k\}$, but not necessarily the lists satisfy $\bigcup_{v \in V} L(v) = \{A_1, \ldots, A_k\}$. The LIST M-PARTITION problem asks: "Does G admit an M-partition in which each vertex v is assigned to a part in $L(v)$?".

Every LIST M-PARTITION problem with M of dimension $k = 4$ was classified by Feder, Hell, Klein and Motwani (2003) as either solvable in quasi-polynomial time or NP-complete. Here, quasi-polynomial time is an $O(n^{c \log^t n})$ time complexity, where t and c are positive constants and n is the number of vertices in the input graph. There is no NP-complete problem that is known to have a quasi-polynomial-time solution, and it is generally believed that problems solvable in quasi-polynomial time are unlikely to be NP-complete. Cameron, Eschen, Hoàng and Sritharan (2007) showed that all LIST 4-PARTITION problems solved in quasi-polynomial time by Feder, Hell, Klein and Motwani (2003) are actually polynomial-time solvable, with the sole exception of the LIST STUBBORN problem (and its complement), for which the best known complexity remains quasi-polynomial time. Denote by A, B, C, D the parts of a four dimensional M-partition problem. We remark that in a general 4-partition problem, any of the four sets A, B, C, D may be empty. The LIST STUBBORN problem is the list M-partition problem where $M_{A,A} = 0$, $M_{B,B} = 0$, $M_{D,D} = 1$, $M_{A,C} = M_{C,A} = 0$, and all other entries are asterisks ($*$). Equivalently, parts A and B are stable sets, part D is a clique, and each vertex of A is nonadjacent to each vertex of C. According to (Cameron, Eschen, Hoàng and Sritharan 2007), a polynomial-time solution for the stubborn problem, if one exists, appears to be difficult and to require methods different from those presented in (Feder, Hell, Klein and Motwani 2003).

Let M, N be 3×3 matrices, F and H be graphs on the same vertex set V. The TWOFOLD 3-PARTITION problem asks: "does V admit a 3-partition such that it is an M-partition for graph F and it is an N-partition for graph H? In this work, we introduce the TWOFOLD STUBBORN problem, where we seek for a twofold 3-partition of V into parts X, Y and Z, according to matrix M, with $M_{X,X} = 0$, $M_{Y,Y} = *$, $M_{Z,Z} = 1$, and matrix N, with $N_{X,X} = N_{Y,Y} = 0$, $N_{Z,Z} = *$.

The COMPATIBLE 3-COLORING problem has been defined in (Feder & Hell 2006) and (Hell & Nesetril 2008) with a similar motivation to eliminate the need for lists. It is shown in (Feder & Hell 2006) that the LIST STUBBORN problem and the COMPATIBLE 3-COLORING problem are closely related and also that the COMPATIBLE 3-COLORING problem is at least as hard as the LIST STUBBORN problem. In (Feder, Hell, Kràl and Sgall 2005), it was shown that each of these two problems can be solved in $O(n^{O(\frac{\log n}{\log \log n})})$ time, thus improving the bound of $O(n^{O \log n})$ established in (Feder, Hell, Klein and Motwani 2003).

The COMPATIBLE k-COLORING problem is stated as follows: Given a complete graph G whose edges are colored by $0, 1, \ldots, k-1$, decide whether G admits a *compatible vertex coloring* by colors $0, 1, \ldots, k-1$ so that for every edge $e = uv$ of the graph, e, u, v cannot be simultaneously colored by the same color, that is, the situation $c(e) = c(u) = c(v)$ is not allowed, where $c(e), c(u), c(v)$ denote the colors of e, u, v, respectively.

We remark that if the TWOFOLD STUBBORN problem has a solution, then an associated COMPATIBLE 3-COLORING problem also has a solution. Given the graphs F and H on the same vertex set V, construct an instance G of the COMPATIBLE 3-COLORING problem in the following way: the vertex set of G is V, and an edge e is colored by 0 if it does not exist in F nor in H, by 1 if it exists only in F, and by 2 if it exists both in F and in H. An edge that exists only in H can receive any color. Let (X, Y, Z) be a solution of the TWOFOLD STUBBORN problem. Then $G[X]$ contains only edges colored by 0, $G[Y]$ contains edges colored by 0 or 1, and $G[Z]$ contains only edges colored by 1 or 2. This means that vertices in X can receive color 1, vertices in Y color 2, and vertices in Z color 0, yielding a compatible vertex coloring.

The advances on the LIST STUBBORN problem we report in this work are: we polynomially reduce a general LIST STUBBORN instance to a particular instance with a structured graph and only two types of lists; and show that this particular LIST 4-PARTITION problem is polynomially equivalent to the TWOFOLD STUBBORN problem.

The problem considered in the paper can be seen as a framework that embodies several important partition problems in graph theory. One application of such a framework is the classification of such partition problems according to their time complexity. In this context, the main, and relevant, contribution of the present paper is to show that the only LIST 4-PARTITION problem not classified as polynomial time solvable or NP-complete is polynomially equivalent to the nonlist TWOFOLD STUBBORN problem.

2 Reducing the LIST STUBBORN problem

The LIST STUBBORN problem reduces to solving a polynomial number of simplified LIST PARTITION problems, each defined on a graph with a particular structure and only two types of lists. The reduction consists of the steps:

1. *Cleaning up lists.* Repeatedly apply the following operation until no longer possible or the list of some vertex becomes empty: if there exists a vertex v with singleton list $L(v) = A$, B or C then for every neighbor w of v remove, respectively, A and C, B or A from $L(w)$; and if there exists a vertex v with $L(v) = D$ then for every non-neighbor w of v remove D from $L(w)$. At the end, if some list becomes empty then return NO, otherwise no conflicts exist. Remark a vertex with a singleton list must be placed in the corresponding part, so the current list problem has as input the set of vertices that currently have non singleton lists.

2. *Eliminating A from lists containing AC.* Suppose $L(v)$ contains AC for some v. If there exists a solution where v is placed in A, then v could be moved to C to obtain another solution, because v is not adjacent to $A \setminus \{v\}$. Hence, we can assume that v is not placed in A in a final solution. Remove A from every list containing AC. If some singleton lists are generated, apply Step 1 again.

3. *Enumerating cliques.* At this point, the current non singleton lists are AB, AD, BC, BD, CD, ABD, BCD. Consider the subgraph G' induced by the vertices whose lists contain no C. Clearly, G' must be a $(2,1)$-graph, a graph that can be partitioned into at most two stable sets and one clique. The LIST $(2,1)$-PARTITION problem P' for G' (defined by the current lists AB, AD, BD, ABD of the vertices in $V(G')$) can be solved in polynomial time, see (Feder, Hell, Klein and Motwani 2003). If P' has no solution then return NO. Otherwise, we can obtain in polynomial time a solution (A_1, B_1, D_1) for P', where A_1, B_1 are stable sets and D_1 is a clique in G'. Observe that any other solution (A_2, B_2, D_2) for P' must satisfy $|D_2 \cap D_1| \geq |D_1| - 2$ and $|D_2 \setminus D_1| \leq 2$, since D_1, D_2 are cliques and $A_1 \cup B_1, A_2 \cup B_2$ induce bipartite graphs. This means that, considering the collection $\{(A_i, B_i, D_i)\}$ of all possible solutions for P', the number of distinct D_i's is polynomially bounded. Therefore, starting from D_1, we can enumerate the candidate cliques for a solution in time $O(n^4)$ by looking at the subsets $\mathscr{D} \subseteq V(G')$ such that $|\mathscr{D} \cap D_1| \geq |D_1| - 2$, $|\mathscr{D} \setminus D_1| \leq 2$, and $\mathscr{D}$ is a clique in G'. For each such $\mathscr{D}$ containing only vertices with lists AD, BD or ABD, vertex v is placed in part D if and only if $v \in \mathscr{D}$. Set $L(v) = D$ for every $v \in \mathscr{D}$ and apply Step 1 for cleaning up lists which means that are no longer vertices with lists AD, BD or ABD. So Step 3 takes a problem with lists AB, AD, BC, BD, CD, ABD, BCD and branches into $O(n^4)$ list problems with lists AB, BC, CD, BCD. Perform Steps 3.1 to 3.6 (these steps are nested into Step 3).

3.1. *Contracting vertices with lists AB.* Let G'' be the bipartite subgraph induced by the vertices with lists AB. Let $G''[X \cup Y]$ be a connected component of G'', and let $x \in X$. In every solution, if x is placed in A (resp. B) then every $x' \in X \setminus \{x\}$ must also be placed in A (resp. B). Thus, we can contract all the vertices in X into a single vertex x with $L(x) = AB$. (When contracting, set $N(x) = \cup_{v \in X} N(v)$.) Likewise, we can contract vertices in Y into a single vertex y with $L(y) = AB$. If X or Y is empty, say $X = \emptyset$, then $Y = \{y\}$; in this case, create a new vertex x adjacent to y and set $L(x) = AB$. At the end of this step, vertices with lists AB form an induced matching. This property will be preserved through the next steps.

3.2. *Eliminating lists CD.* We transform lists CD into lists BCD as follows. For each vertex v with $L(v) = CD$, create in current graph G new vertices x_v, y_v, z_v and new edges $vx_v, vy_v, x_vy_v, y_vz_v$. Set $L(v) = BCD, L(x_v) = BC, L(y_v) = L(z_v) = AB$. Vertex z_v is created to preserve the property that vertices with lists AB form an induced matching. After the transformation, v is not allowed to be placed in B, but can still be placed in C or D. Current lists are now AB, BC, BCD. See Figure 1.

3.3. *Checking triangles AB-AB-BCD.* For every triangle uvw with $L(u) = L(v) = AB$ and $L(w) = BCD$, set $L(w) = D$ (w cannot be placed in B or C). Next, apply Step 1 for cleaning up lists. At the end of this step, for every edge uv with $L(u) = L(v) = AB$ and every w with $L(w) = BCD$, w is adjacent to at most one of u and v. This property will be preserved through the next steps.

3.4. *Checking triangles AB-AB-BC.* If there exists a triangle uvw with $L(u) = L(v) = AB$ and $L(w) = BC$, return NO. Otherwise, for every edge uv with $L(u) = L(v) = AB$ and every w with $L(w) = BC$, w is adjacent to at most one of u and v.

3.5. *Eliminating lists BC.* If G contains vertices with lists BC, include in G new vertices $a, b, c, d, e, f, u, v, w$ and new edges such that $abvu, cdwv, efwu$ are induced C_4's. Link u, v, w by edges to every original vertex of G with list BCD. Finally, set $L(u) = L(v) = L(w) = BCD$, set $L(a) = L(b) = L(c) = L(d) = L(e) = L(f) = AB$ and include D to the list of every vertex x with list BC. Say that x has an *augmented list.* Observe that at least two vertices of u, v, w, say u and v, cannot be placed in B. Since a or b must be placed in A, say a, we have that u must necessarily be placed in D. Since no vertex x with augmented list is adjacent to u, we have that x cannot be placed in D, as desired. This eliminates lists BC and does not affect the set of solutions for the problem. Lists become now AB, BCD. See Figure 2.

3.6. *Creating pendant vertices.* For every v with $L(v) = AB$, create two pendant vertices v', v'' adjacent to v such that $L(v') = L(v'') = BCD$. This operation will be useful in the proof of Theorem 1.

At the end of Step 3.6, the current graph G is equipped with only two types of lists, AB and BCD, such that: (i) vertices with lists AB form an induced matching; (ii) for every edge uv with $L(u) = L(v) = AB$ and every w with $L(w) = BCD$, w is adjacent to at most one of u and v; (iii) every vertex with list AB is adjacent to two pendant vertices with lists BCD. Call this particular LIST 4-PARTITION problem the REDUCED LIST STUBBORN problem.

3 Equivalence between the LIST STUBBORN problem and the TWOFOLD STUBBORN problem

In the TWOFOLD STUBBORN problem, we seek for a twofold 3-partition of vertex set V into parts X, Y and Z, according to two matrices M, with $M_{X,X} = 0$, $M_{Y,Y} = *$, $M_{Z,Z} = 1$, and N, with $N_{X,X} = N_{Y,Y} = 0$, $N_{Z,Z} = *$. Let G be an instance of the REDUCED LIST STUBBORN problem and define suitable graphs F and H, as follows. Let $F = G_{BCD}$ be the subgraph of G induced by the vertices with lists BCD. Define a graph H with $V(H) = V(G_{BCD})$ such that there exists an edge between $u, z \in V(H)$ if and only if there exists a path $uxyz$ in G with $L(x) = L(y) = AB$. Note that $uz \in E(H)$ implies that u and z cannot be placed in a same part of the set $\{B, C\}$. In other words, Graph H captures the behavior vertices with lists BCD must have in relation to vertices with lists AB.

Theorem 1 *If the* TWOFOLD STUBBORN *problem can be solved in polynomial time, then the* LIST STUBBORN *problem can be solved in polynomial time.*

Proof: We show that there is a solution of the REDUCED LIST STUBBORN problem for graph G if and only if there is a solution of the TWOFOLD STUBBORN problem for graphs G_{BCD}, H. Suppose there exists a solution (A, B, C, D) of the REDUCED LIST STUBBORN problem for graph G. Set $X = B \cap V(G_{BCD}), Y = C, Z = D$ to get a solution (X, Y, Z) of the TWOFOLD STUBBORN problem for graphs G_{BCD}, H. $G_{BCD}[X]$ is an edgeless graph and $G_{BCD}[Z]$ is a complete graph. Let $u, v \in X$. Since both u and v belong to B, by construction of H we have $uv \notin E(H)$. Thus $H[X]$ is an edgeless graph. If $x, y \in Y = C$, again by construction of H we have $xy \notin E(H)$, that is, $H[Y]$ is an edgeless graph.

Suppose there exists a solution (X, Y, Z) of the TWOFOLD STUBBORN problem for graphs G_{BCD}, H. Analyze first the vertices in $V(G) \setminus V(G_{BCD})$, all of which have lists AB. Let $xy \in E(G)$ such that $L(x) = L(y) = AB$. Clearly, $N_G(x) \setminus \{y\} \subseteq V(G_{BCD})$. Let $S_x = (N_G(x) \setminus \{y\}) \cap (X \cup Y)$. By construction of H, all the vertices of S_x are adjacent in H to the pendant vertices y', y'' linked to y. In addition, since Z is a clique in G_{BCD} and $y'y'' \notin E(G_{BCD})$, one of y', y'' belongs to $X \cup Y$, which means that either $S_x \subseteq X$ or $S_x \subseteq Y$. In the former case include x in A and y in B, and in the latter include x in B and y in A. Proceed in this way for all the edges whose extreme vertices have lists AB. Since the vertices in $V(G) \setminus V(G_{BCD})$ form an induced matching, the subsets A and B constructed in this way are stable sets in G. This procedure ensures that no vertex in B is adjacent to a vertex in X, and no vertex in A is adjacent to a vertex in Y. Since X is a stable set in G_{BCD}, include all the vertices of X in B, and set $C = Y, D = Z$ to get (A, B, C, D) a solution of the REDUCED LIST STUBBORN problem for graph G. ■

Theorem 2 *If the* TWOFOLD STUBBORN *problem is NP-complete, then the* LIST STUBBORN *problem is NP-complete.*

Proof: Let F, H be an instance of the TWOFOLD STUBBORN problem. Construct G', instance of the LIST STUBBORN problem: $V(G') = V(H) \cup \{x_{uv}, y_{uv} \mid uv \in E(H)\}$, $E(G') = E(F) \cup \{ux_{uv}, vy_{uv}, x_{uv}y_{uv} \mid uv \in E(H)\}$. Set $L(v) = BCD$ for every $v \in V(H)$, and set $L(x_{uv}) = L(y_{uv}) = AB$ for every $uv \in E(H)$. The construction is terminated. It is not difficult to see that the pair F, H is a yes-instance of the TWOFOLD STUBBORN problem if and only if G' equipped with the lists above is a yes-instance of the LIST STUBBORN problem. ■

Corollary 3 *The* LIST STUBBORN PROBLEM *is polynomially equivalent to the* TWOFOLD STUBBORN *problem.* ■

We remark that the related TWOFOLD 3-PARTITION problem is NP-complete: matrix M, with $M_{X,X} = M_{Y,Y} = *$ and $M_{Z,Z} = 1$, and matrix N, with $N_{X,X} = N_{Y,Y} = 0$ and $N_{Z,Z} = *$, via a simple reduction from 3-COLORABILITY: given a graph G, define graphs F, H such that $F = \overline{G}$ and $H = G$; then $V = V(G)$ can be partitioned into 3 independent sets X, Y, Z if and only if V can be partitioned into subsets X, Y, Z such that Z is a clique in $F = \overline{G}$ and Y, Z are independent sets in $H = G$.

4 Final Remarks

Resolving the complexity of the LIST STUBBORN problem is an interesting open problem, as all other list 4-partition problems have been classified as polynomial or NP-complete (Cameron, Eschen, Hoàng and Sritharan 2007). The present paper does not resolve the question, but it shows that the LIST STUBBORN problem is polynomially equivalent to two problems. The first is the LIST STUBBORN problem with just types of lists, the lists AB and BCD and a very restricted structure. The second is the TWOFOLD STUBBORN problem: given two graphs F and H on the same vertex set, partition the vertices into sets X, Y and Z such that X and Z are an independent set and a clique, respectively, in F, and X and Y are independent in H.

The particular instance of the LIST STUBBORN problem where four vertices have the singleton lists, say: $L(x) = \{A\}$, $L(y) = \{B\}$, $L(z) = \{C\}$ and $L(w) = \{D\}$, and all remaining $n - 4$ vertices have the list $\{ABCD\}$ can be considered. Remark that by solving $O(n^4)$ such list problems, one may decide whether a given graph G admits a partition into four nonempty parts A, B, C, D.

This approach was used for the classification of the SKEW PARTITION problem as polynomial (de Figueiredo, Klein, Kohayakawa and Reed 2000). The SKEW PARTITION problem asks for a partition of the vertex set into four nonempty parts A, B, C, D such that each vertex of A is adjacent to each vertex of B, and each vertex of C is nonadjacent to each vertex of D. Remark that the SKEW PARTITION satisfies only external constraints.

In (Dantas, de Figueiredo, Gravier and Klein 2005), all 4-partition problems into four nonempty parts A, B, C, D satisfying only external constraints were classified into NP-complete or polynomial with the sole exception of the $2K_2$-partition, where each vertex of A is adjacent to each vertex of B, and each vertex of C is adjacent to each vertex of D. Remark that the LIST $2K_2$-PARTITION problem was classified as NP-complete (Feder, Hell, Klein and Motwani 2003).

As remarked in (Dantas, de Figueiredo, Gravier and Klein 2005), a solution for the particular instance of the LIST STUBBORN problem where four vertices have the singleton lists, say: $L(x) = \{A\}$, $L(y) = \{B\}$, $L(z) = \{C\}$ and $L(w) = \{D\}$, and all remaining $n - 4$ vertices have the list $\{ABCD\}$ can be obtained in polynomial time. First, note that lists ABD, AB, AD may be discarded, since a vertex containing A in its list must contain also C. Now, given the remaining non singleton lists $\{ABCD, ABC, ACD, BCD, AC, BC, BD, CD\}$, a solution is obtained by placing all vertices containing C in their nontrivial lists into part C, and by verifying whether the remaining graph (the subgraph induced by the vertices with list BD) is a split graph which can be done by applying 2-SAT.

Up to now there are no 4-partition problems for which the list problem is classified as NP-complete whereas the nonlist nonempty part problem is classified as polynomial. Remark that the $2K_2$-partition and the stubborn partition are the remaining candidates to be complexity separating problems.

The generalization LIST STUBBORN SANDWICH problem, where two graphs $G^1 = (V, E^1), G^2 = (V, E^2)$ such that $E^1 \subseteq E^2$, and for each vertex a list of parts in which the vertex is allowed to be placed are given, and we look for a sandwich graph $G = (V, E)$ such that $E^1 \subseteq E \subseteq E^2$ with a stubborn partiton satisfying the lists requirements, was proved to be NP-complete (Dantas and Faria 2007).

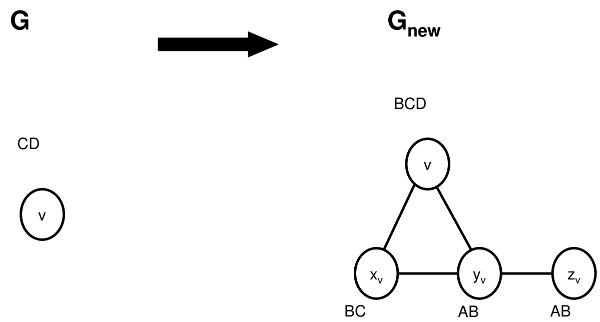

Figure 1: Reduction rule 3.2

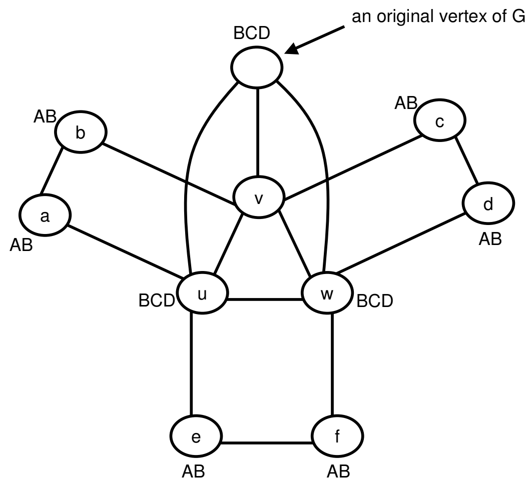

Figure 2: Reduction rule 3.5

References

Cameron, K., Eschen, E. M., Hoàng, C. T. & Sritharan, R. (2007), The complexity of the list partition problem for graphs, SIAM J. Discrete Math. **21**, 900–929.

Dantas, S., de Figueiredo, C. M. H., Gravier, S. & Klein, S., Finding H-partition problems, Theor. Inform. Appl. 39, 133–144.

Dantas, S. & Faria, L. (2007), On stubborn graph sandwich problems, In: Second International Workshop on Modeling, Optimization, and Complexity (MOC'2007), 46–52.

Feder, T., Hell, P., Klein, S. & Motwani, R. (2003), List partitions, SIAM J. Discrete Math. **16**, 449–478.

Feder, T., Hell, P., Král, D. & Sgall, J. (2005), Two algorithms for general list matrix partitions, Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 870–876.

Feder, T. & Hell, P. (2006), Full constraint satisfaction problems, SIAM J. Discrete Math. 36, 230–246.

Hell, P. & Nesetril, J. (2008), Colouring, constraint satisfaction, and complexity, Computer Science Review **2**, 143–163.

de Figueiredo, C. M. H., Klein, S., Kohayakawa, Y & Reed, B. (2000), Finding skew partitions eficiently, J. Algorithms 37, 505–521.