RAIRO Operations Research
RAIRO Oper. Res. **35** (2001) 211-228

SEMI-DEFINITE POSITIVE PROGRAMMING RELAXATIONS FOR GRAPH K_n-COLORING IN FREQUENCY ASSIGNMENT

Philippe Meurdesoif[1] and Benoît Rottembourg[2]

Abstract. In this paper we will describe a new class of coloring problems, arising from military frequency assignment, where we want to minimize the number of distinct n-uples of colors used to color a given set of n-complete-subgraphs of a graph. We will propose two relaxations based on Semi-Definite Programming models for graph and hypergraph coloring, to approximate those (generally) NP-hard problems, as well as a generalization of the works of Karger *et al.* for hypergraph coloring, to find good feasible solutions with a probabilistic approach.

Résumé. Dans cet article, nous décrivons une nouvelle classe de problèmes de coloration rencontrés en Allocation de Fréquences militaire : nous voulons minimiser le nombre de n-uplets distincts utilisés pour colorier un ensemble doné de n-cliques d'un graphe. Pour approcher ces problèmes généralement NP-difficiles, nous proposons deux relaxations basées sur les modélisations semi-définies de la coloration de graphes et d'hypergraphes, ainsi qu'une généralisation des travaux de Karger *et al.* à la coloration d'hypergraphes, pour trouver de bonnes solutions faisables par une approche probabiliste.

Keywords: Discrete optimization, semidefinite programming, frequency assignment, graph coloring, hypergraph coloring.

Received May, 1999.

[1] Projet NUMOPT, INRIA Rhône-Alpes, 655 avenue de l'Europe, 38330 Montbonnot-Saint-Martin, France; e-mail: `Philippe.Meurdesoif@inrialpes.fr`

[2] Direction des Technologies Nouvelles, Bouygues, 1 avenue Eugène Freyssinet, 78061 Saint-Quentin-en-Yvelines, France; e-mail: `brottemb@challenger.bouygues.fr`

1. Introduction

With the recent explosion of mobile communication systems and the growing needs in transmitting data, available frequencies are becoming scarser, so one of the most acute issues is to densify frequency occupation with respect to communication quality criteria. In a military context, a recent technology is used: hopping frequency, where two radio sets do not communicate on *one* frequency, but periodically and synchronically change frequencies (typically 1000 times per second). We consider the case of a deployment composed of N networks (sets of radio sets communicating with each other). Every network must then be assigned a set of frequencies (called *frequency hopset*), which is composed of several frequency intervals in the VHF band. In an electronical war context, studies by the CELAR[3] show that partial overlapping of *frequency hopsets* can be allowed with acceptable deterioration of the communication quality.

Vehicles spread on an operation field are provided one or more radio sets, each one belonging to a given network. So, vehicles with two or three radio sets (called multipost vehicles), can belong to up to three different networks. Frequency hopsets assigned to such networks must be disjoint enough to avoid interferences, so we say that there is a "covehicle link" between them. There can also exist a "cosite link" between two networks when two vehicles are geographically close enough. We aim at assigning frequency hopsets to all networks so as to avoid interferencies.

Unfortunately, it is very hard to model, and furthermore to compute, interferencies in this context. Practically, it is necessary to simulate interferencies once we have a global assignment of frequency hopsets to know if an assignment is acceptable or not. Furthermore, a large amount of different criteria are used to evaluate the quality of such assignments. The obvious complexity of the simulation/optimization procedure leads to tremendous calculation time. So we propose in a first pretreatment step to reduce the problem by grouping networks which are not linked together into sets of networks to which the same frequency hopset can be assigned, so as to minimize the parameters which are responsible for the time-cost: the number of remaining vehicles with three radio sets (such vehicles, called "trisets", belong to three networks; this is the strongest constraint between networks), and of course the number of networks sets.

Assigning frequencies or frequency hopsets to radio sets with respect to exclusion constraints can obviously be regarded as graph-coloring problems: assigning a color to each vertex of a graph (each vertex corresponding to a network) so that no edge (corresponding to one or more links between two networks) has both its endpoints colored the same way.

The most common coloring problem is to find the smallest possible number of colors, called the "chromatic number" of the graph, which is known to be NP-Hard in the general case (see [9] for example). Our problem is a little different: it is to find the lowest number of triangles (corresponding to trisets physically) in the

[3] Centre d'ÉLectronique de l'ARmement.

final, reduced graph. A triangle is composed of three vertices mutually linked; alternatively, it can be called a 3-clique or K_3. Obviously, any n-clique of a graph will need exactly n colors in any coloration. In the following, we will call "flag" the n-uple of colors used to color a given n-clique of a graph. We can reformulate our problem this way: "being given a graph $G(V,E)$ and a set T_3 of 3-cliques of G, find a coloration of the graph minimizing the number of distinct flags used to color the K_3s in T_3".

In this paper, we will study the general form of this problem, which we will call "K_n-Coloring": minimizing the number of distinct flags to colors the n-cliques in T_n. Defaix [2] studied the problem with $n = 3$ through a Simulated Annealing approach, and the case $n = 1$ appears to be equivalent to the chromatic number of a graph, which is one of the most studied combinatorial problems. A recent and fruitful breakthroughs in this field was the application of a Semi-Definite Programming (SDP) relaxation by Karger *et al.* [6], based on the work of Goemans *et al.* [5], which we will recall later and adapt to our problem.

In the first part, we will explore the NP-Hardness of K_n-Coloring problems, and show a negative result on approximability of these problems. In the second part, we will show that the integer SDP formulation from Karger *et al.* is equivalent to Graph Coloring constraints. A peculiar cost function for our case will be discussed in the third part. In the fourth part we will extend Karger's approach for Graph Coloring to the problem of Hypergraph Coloring and show how it can be used to derive a bound on K_n-Coloring. Finally, practical results will be shown in the fifth part.

2. Complexity and approximability results

In this part, we will express some complexity results, based on polynomial transformations from the Vertex-Coloring problems and their decision versions: p-Coloring (color a graph using at most p colors), which are known to be NP-Hard for $p \geq 3$ (the other cases being polynomial). We will denote $p - K_n$-Coloring the decision problem: "can I color a graph $G(V,E)$ using at most p distinct n-uples of colors to color the n-cliques in an initial set T_n?" There exist obvious transformations between $p - K_n$-Coloring problems, leading to the following propositions:

Proposition 1.

$$\forall p \geq 1,\, n \geq 1$$

$$p - K_n\text{−Coloring} \propto p - K_{n+1}\text{−Coloring}.$$

Proof. Let $G(V,E)$ and T_n be any instance of $p - K_n$-Coloring. We will form $G'(V',E')$ and T_{n+1}, an instance of $p - K_{n+1}$-Coloring, by adding a new vertex w linked to all vertices of V, and the cliques in T_{n+1} by connecting w to all cliques of T_n. Any coloring of G' will consist in a coloring of G and a different color assigned to w. Clearly, solving $p - K_n$-Coloring on G is equivalent to solve $p - K_{n+1}$-Coloring on G', hence the proposition. □

Proposition 2.

$$\forall p \geq 1,\, n \geq 1$$
$$p - K_n\text{−Coloring} \propto (p+1) - K_n\text{−Coloring}.$$

Proof. It suffices to create n new vertices forming a clique added to T_n. Linking at least one of these vertices to all vertices in V will force the new clique to be assigned a flag distinct from all others, thus the equivalence of both problems. □

More particularly, we can notice that K_1-Coloring amounts to solving the problem of (vertex)-coloring the subgraph of G induced by the vertices in T_1. Thus, we have the following:

Proposition 3. *$p - K_1$-Coloring is polynomial for $p = 1$ and $p = 2$. $p - K_1$-Coloring is NP-Complete $\forall p \geq 3$.*

Proposition 4. *$1 - K_2$-Coloring is polynomial. $p - K_2$-Coloring is NP-Complete $\forall p \geq 2$.*

Proof. $1 - K_2$-Coloring is equivalent to coloring the subgraph induced by the vertices involved in T_2 with 2 colors, which is polynomial. Moreover, we show that 4-Coloring $\propto 2 - K_2$-Coloring. Given a graph $G(V, E)$ we form a graph $G'(V', E')$ from G by adding $|V|$ new vertices w_i bijectively linked to the vertices v_i of V and we set $T_2 = \{(v_i, w_i)\}$. Clearly, G' is $2 - K_2$-colorable if and only if G is 4-colorable. Finally, using previous propositions, we can prove the NP-Completeness for $p \geq 3$. □

Proposition 5. *$\forall n \geq 3$, $p \geq 1$, $p - K_n$-Coloring is NP-Complete.*

Proof. It suffices to prove that $1 - K_3$-Coloring is NP-Complete, which we do through a transformation from 3-Coloring: we create $|E|$ new vertices and replace all edges in E by triangles using those vertices. Those triangles form T_3. Equivalence between those problems is obvious. □

There exists another polynomial transformation from a Coloring problem to a K_n-Coloring problem which preserves the optimal values.

Proposition 6.

$$p\text{−Coloring} \propto p - K_n'\text{−Coloring}.$$

Proof. Given a graph $G(V, E)$ we expand it to a graph $G'(V', E')$ by adding a $K_{(n-1)}$ clique of new vertices $w_1, \ldots, w_{n-1}$, completely linked to the vertices of V. The set of cliques T_n consists in all K_n of the form $(w_1, \ldots, w_{n-1}, v)$ with $v \in V$. In every coloration of G' the colors assigned to $w_1, \ldots, w_{n-1}$ are distinct to the ones assigned to the vertices of V. It is therefore clear that there exists a $p - K_n$-coloration of G' iff there exists a p-coloration of G. □

We can obtain non-approximability results for K_n-Coloring by combining this transformation with hardness of approximation results for Coloring, such as the result of Lund *et al.* [9]: "there is a constant δ such that no polynomial approximation algorithm can achieve a ratio of $|V|^\delta$ for graph coloring, unless $P = NP$", or the better result of Feige *et al.* [3]: "for all positive ε, it is intractable to approximate $\chi(G)$ to within $N^{1-\varepsilon}$ unless $NP \subseteq ZPP$". We get the following:

Theorem 1. *For all $\varepsilon > 0$ it is intractable to approximate K_n-Coloring within $(|V| - n + 1)^{1-\varepsilon}$ unless $NP \subseteq ZPP$.*

3. Semi-definite programming modelization of graph coloring

Our problem of K_n-Coloring is to find a coloration of a graph which optimizes a particular cost function. Constraints are in fact the same as for a classical coloring problem, thus we will try to find a good model of such constraints.

Among all the (polynomial in size) linear models, one of the simplest is to use the boolean variable x_{ik}: "vertex i is assigned the color k". But this model is made unefficient because of symmetries in variables k. Nonetheless, we will use it to derive a second model in boolean variables m_{ij}: "vertices i and j are assigned the same color". The link between both models is:

$$m_{ij} = \sum_{k=1}^{n} x_{ik} x_{jk} = u_i^T u_j \text{ with } u_i = \begin{pmatrix} x_{i1} \\ \vdots \\ x_{in} \end{pmatrix}.$$

Hence such matrix $M = (m_{ij})$ is Positive Semi-Definite (we recall that a matrix M is Positive Semi-Definite if, and only if, there exists a family of vectors $u_1, \cdots u_n$ such that $M_{ij} = u_i^T u_j \ \forall i, j$, and we denote it by $M \succeq 0$; we will also denote by $\mathcal{S}^+$ the space of Positive Semi-Definite matrices).

More generally, we can define such a matrix M for any kind of partitioning of a set. The following holds:

Proposition 7. *Let S be a set and M a boolean matrix of dimension $|S| \times |S|$. Formulations (i) and (ii) are equivalent:*

(i) *S is partitioned into subsets and $\forall (i,j) \in S \times S$, $M_{ij} = 1 \Leftrightarrow i$ and j belong to the same subset;*
(ii) *$M \succeq 0$, $M_{ii} = 1 \, \forall i \in \{1, \cdots, |S|\}$.*

Proof. If S is partitioned into subsets, defining $M_{ij} \in \{0; 1\}$: "i and j belong to the same subset"; we can define variables x_{ik} as above. Using the above formula we have (ii). Reciprocally, if such a matrix M exists, let us define the relation R:

$$iRj \Leftrightarrow M_{ij} = 1.$$

R is reflexive because $\forall i \, M_{ii} = 1 \Rightarrow iRi$;
R is symmetric because M is symmetric;

if iRj and jRk, let us suppose that we do not have iRk. Then the submatrix of M corresponding to i, j, k is:

$$\begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix} \not\succeq 0,$$

this in contradiction to the fact that $M \succeq 0$, so iRk and R is transitive. So, R is a relation of equivalence, whose classes define a partition of S. □

Corollary 1. *In a Semi-Definite Programming model, constraints of a Coloring Problem can be expressed as follows:*

$$\begin{aligned} M_{ii} &= 1 \qquad \forall i \in V \\ M_{ij} &= 0 \quad \forall (ij) \in E \\ M \in \{0;1\}^{|V|\times|V|} &\cap \mathcal{S}^+. \end{aligned}$$

If we want to minimize the number of colors (chromatic number), we can notice that it is possible to reproduce the same proof in a model where we replace 0 by a real $\alpha \le 0$, but we must verify that, for a partitioning into κ subsets (especially a κ-coloring), there exists a set of κ unit vectors in $\Re^{|S|}$ whose mutual dot products are exactly α. For any given integer κ, this can be achieved iff $\alpha \ge -\frac{1}{\kappa-1}$ (see [4] or [6]). This means that the existence of a matrix M such that:

$$\begin{aligned} M_{ii} &= 1 \quad \forall i \in S \\ M_{ij} &\in \left\{-\tfrac{1}{\kappa-1}; 1\right\} \\ M &\succeq 0 \end{aligned}$$

amounts to the existence of a partitioning of S into at most κ subsets. Minimizing κ amounts to minimizing $-\frac{1}{\kappa-1} = \alpha$, so we get the following theorem:

Theorem 2. *Chromatic number* $\chi(G) = 1 - \frac{1}{\alpha^*}$ *where*

$$\begin{aligned} & \alpha^* = \min \alpha \\ s.t. & \\ & M_{ii} = 1 \qquad \forall i \in V \\ & M_{ij} = \alpha \qquad \forall (ij) \in E \\ & M_{ij} \in \{\alpha; 1\} \quad \forall (i,j) \in V^2 \\ & M \succeq 0. \end{aligned}$$

Relaxing the discretization constraint into: $M_{ij} \in [\alpha; 1]$, $\forall (ij) \in V^2$ yields a polynomially computationable lower bound of $\chi(G)$. Karger *et al.* [6] showed that this bound was in fact equal to the Lovász number of $\bar{G}$: $\theta(\bar{G})$, introduced in [8].

Relaxed solutions are still Positive Semi-Definite matrices M which we can factorize (using a Cholesky method) to obtain $N = |V|$ unit vectors of $\Re^N$ $u_1, \cdots, u_N$ such that $u_i^T.u_j = \alpha$, $\forall (ij) \in E$. The idea is to use those vectors to construct a stable set by randomly choosing a vector r (center) in $\Re^N$ which will "capture" all u_i such that $r^T.u_i \geq c$ where c is a certain threshold. Then, if this set of vertices is not stable, we will make it stable by removing from the set one endpoint of any entirely captured edge. Iterating this process on non-captured vertices until all vertices are captured will result in a coloration of G. It is shown in [6] that for $c = \sqrt{2(1-2/k)\ln\Delta}$ (with Δ maximum degree of the graph), the expected number of colors is in $O(\Delta^{1-2/k}\sqrt{\ln\Delta}\ln n)$.

For K_n-Coloring, we propose to keep the coloring constraints of the SDP modelization of chromatic number, but we will need an other cost function to optimize.

An expression of the cost function for K_n-Coloring can be, denoting by $K^{(1)}, \cdots, K^{(|T_n|)}$ the cliques in T_n:

$$|T_n| - \text{MaxCard}\left\{i \leq |T_n| \,:\, \exists j < i \;\; s.t.\; K^{(i)} \equiv K^{(j)}\right\},$$

with $\equiv$ meaning that two cliques are assigned the same flag. The expression means that we maximize the number of cliques in T_n colored the same way than an other one; ordering the cliques prevent us from multiple countings.

We can notice that, since $\forall u, v,\; M_{uv} \in \{\alpha; 1\}$:

$$K^{(i)} \equiv K^{(j)} \Leftrightarrow \sum_{u \in K^{(i)}} \sum_{v \in K^{(j)}} M_{uv} = n + \alpha(n^2 - n).$$

This way we express the cost function as follows:

$$|T_n| - \text{MaxCard}\left\{i \leq |T_n| \,:\, \exists j < i \;\; s.t. \sum_{u \in K^{(i)}} \sum_{v \in K^{(j)}} M_{uv} = n + \alpha(n^2 - n)\right\}.$$

Unfortunately, this exact function is non-linear, so we cannot handle it in polynomial time after relaxing the Semi-Definite Program.

What we will do is to combine the constraint of "relaxed coloration" we have just obtained to another cost function, linear, such that the resulting vectors u_i are close enough to vectors of a "good" K_n-Coloring to guide the primalization to a solution with few distinct flags to cover T_n.

4. Linearized cost function: Global resemblance

Given a matrix M corresponding to a coloration C (recall that $\forall i, j \;\; M_{ij} \in \{\alpha; 1\}$), let us define the notion of resemblance between two K_ns:

$$R(K^{(1)}, K^{(2)}) \triangleq \sum_{i \in K^{(1)}} \sum_{j \in K^{(2)}} M_{ij}.$$

Some colors are used to color both a vertex of $K^{(1)}$ and a vertex of $K^{(2)}$. Let us denote by $P_{1,2}$ the number of such colors. If a vertex u of $K^{(1)}$ and a vertex v of $K^{(2)}$ are colored with the same color, then $M_{uv} = 1$; else, $M_{uv} = \alpha$. Moreover, there are n^2 pairs $(u, v) \in K^{(1)} \times K^{(2)}$.

Hence we have

$$R(K^{(1)}, K^{(2)}) = P_{1,2} + \alpha[n^2 - P_{1,2}].$$

Our objective will be to maximize the global resemblance on all pairs of K_ns. This will consist in summing entries of M, leading to a sum of the form $A + B\alpha$, with A and B positive numbers depending on the coloration C. For every coloration matrix M with α, all matrices with the same $\alpha, 1$ structure but with $0 \geq \alpha' \geq \alpha$ will be feasible too, so maximizing the global resemblance will lead α to be zero at an optimal point; in the following we will consider $\alpha = 0$. Now, we have the following cost function:

$$\begin{aligned} R_G(C) &\triangleq \sum_{K^{(p)} \in T_n} \sum_{\left[\substack{K^{(q)} \in T_n, \\ p>q}\right]} R(K^{(p)}, K^{(q)}) \\ &= \sum_{K^{(p)} \in T_n} \sum_{\left[\substack{K^{(q)} \in T_n, \\ p>q}\right]} \sum_{i \in K^{(p)}} \sum_{j \in K^{(q)}} M_{ij} \\ &= \sum_{K^{(p)} \in T_n} \sum_{\left[\substack{K^{(q)} \in T_n, \\ p>q}\right]} P_{p,q}. \end{aligned}$$

Intuitively, a coloration maximizing R_G will have a limited diversity of distinct flags covering T_n.

We can notice that some pairs of K_ns cannot be assigned the same flag, if the subgraph of G induced by the vertices involved in such pairs is not n-colorable. For constant n, such a test is polynomial. Some pairs of incompatible K_ns may be forced to have up to $n - 1$ common colors, whereas they are still different in terms of K_n-Coloring. This phenomenon can lead to non-optimal solutions on some instances.

For example, if we want to K_3-color the graph G of Figure 1 with initial $T = \{\mathcal{T}_1; \mathcal{T}_2; \mathcal{T}_3\}$ (both plain and dashed lines indicate the edges, while the K_3s appear as plain-line triangles indicated by a small arc in one angle),we can find a coloring such that $\mathcal{T}_1$ and $\mathcal{T}_3$ receive the same flag, so G is 2-colorable. But if we optimize the global resemblance, we will obtain a coloration such that the resemblance between every pair of K_3s is 2, leading to a $3 - K_3$-coloring of G. Being given any positive integer p, by stacking p pairwise completely connected replicas of G, we will construct an instance for which maximizing the global resemblance will lead to a $3p - K_3$-coloration whereas it is $2p - K_3$-colorable. So, the global resemblance heuristic can lead to arbitrarily bad solutions for K_n-Coloring.

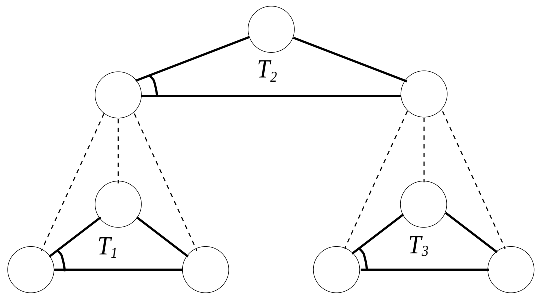

FIGURE 1. Graph with general resemblance leading to non-optimal solution.

We will avoid it by removing pairs of incompatible K_ns from the cost function. Then we have a new cost function ("structured resemblance"):

$$R_C(T_n) \triangleq \sum_{K^{(p)} \in T_n} \sum_{\left[\substack{K^{(q)} \in T_n, \\ p>q, \\ K^{(p)},\, K^{(q)} \text{compatible}}\right]} \sum_{i \in K^{(p)}} \sum_{j \in K^{(q)}} m_{ij}.$$

Application to the classic Coloring problem

To see what we can expect from this cost function, we will restrict ourselves to the case of Graph Coloring, which is actually equivalent to K_1-Coloring. We can express more precisely the corresponding cost function (structured resemblance):

$$R_C(T_1) = 1/2 \sum_{i \in T_1} \sum_{j \in T_1 \setminus [\{i\} \cup N(i)]} m_{ij},$$

with constant factor 1/2 because of double-countings.

In fact, if j is a neighbour of i, we have $m_{ij} = 0$ for any coloration (recall the hypothesis $\alpha = 0$), hence those terms can be added to the sum without modifying the value of R_C. We first show that both resemblances are in fact equal:

$$\begin{aligned} R_C(T_1) &= 1/2 \sum_{i \in T_1} \sum_{j \in T_1 \setminus \{i\}} m_{ij} \\ &= R_G(T_1) \quad \text{(Global Resemblance)}. \end{aligned}$$

Furthermore, considering a κ-coloration of color sets $C_1, \cdots, C_\kappa$, we can express this quantity more explicitly:

$$R_G(T_1) = 1/2 \sum_{i \in T_1} \sum_{\substack{j \in T_1 \setminus \{i\} \\ C(i)=C(j)}} m_{ij}(=1) + 1/2 \sum_{i \in T_1} \sum_{\substack{j \in T_1 \setminus \{i\} \\ C(i) \neq C(j)}} m_{ij}(=0)$$

$$\begin{aligned} R_G(T_1) &= 1/2 \sum_{\substack{(i,j)\in C_1\times C_1 \\ i\neq j}} 1 + \ldots + 1/2 \sum_{\substack{(i,j)\in C_\kappa\times C_\kappa \\ i\neq j}} 1 \\ &= \sum_{k=1}^{\kappa} \frac{|C_k|(|C_k|-1)}{2}. \end{aligned}$$

An exchange-based proof would show that such a cost function tends to favorise colors of great size; more precisely, any coloration optimal for $R_G(T_1)$ is a partition into iterative Max Stable sets.

Modeling K_ns as vertices and finding incompatibility constraints between them will lead to a lower bound for K_n-Coloring obtained through a hypergraph coloring problem.

5. Hypergraph Coloring formulation

5.1. Relaxation through hypergraph coloring

A hypergraph consists in a set of vertices V and a set of hyperedges E, each hyperedge being a set of vertices. If all hyperedges are in fact of cardinality 2 (we say "dimension"), the hypergraph is actually a graph (or a multigraph). Let us partition E into subsets of hyperedges of same dimension: $E_2, \ldots, E_Q$ with $E_q = \{e \in E : |e| = q\}$. We define $m_q = |E_q|$.

The problem of hypergraph coloring consists in finding an assignment of colors on the vertices, such that no hyperedge is monochromatic.

As we saw in the section above, two K_ns are incompatible if the subgraph they induce is not n-colorable. This test can be extended to any subset S of T_n: if the subgraph induced by the vertices involved in S is not n-colorable, the cliques in S cannot be all colored with the same flag. Hence, we have the following proposition:

Proposition 8. *A coloring of the hypergraph $H(V_H, E_H)$ defined by:*

- *vertices in V_H represent the cliques of T_n;*
- *hyperedges in E_H correspond to subsets of T_n inducing non-n-colorable subgraphs of G;*

is a relaxation of K_n-coloring $G(V,E)$ on T_n.

We cannot compute a test for all subsets of T_n; in fact, if a subset S of T_n induces a non-n-colorable subgraph of G and, so, corresponds to a hyperedge e_S of H, any other subset S' of T_n containing S will have the same property, corresponding to a hyperedge of H containing e_S, so a weaker constraint of hypergraph coloring. Hence, we will limit ourselves to sets of cliques with no non-n-colorable subsets. To make it tractable, practically, we will enumerate all sets of a limited number of cliques.

Solving Hypergraph Coloring problems is NP-Hard. Some non-approximability results can be shown, and approximate coloring algorithms were studied for special cases of uniform hypergraphs (where all hyperdeges are of the same, given dimension) with known chromatic number (see [7] for example).

Finding a relaxed solution of Hypergraph Coloring on such hypergraphs will give us a lower bound for K_n-Coloring.

We now present an *SDP* relaxation for Hypergraph Coloring, extending the approach of Karger *et al.* [6].

5.2. A Karger-like approach for hypergraph coloring

Formulation

Let $H(V,H)$ be a hypergraph whose vertices are assigned colors, and $M \in \{\alpha; 1\}^{|V|\times|V|}$ $(\alpha \leq 0)$ the matrix corresponding to this coloration (defined as in 3). Let us consider a hyperedge e of dimension $|e| = q$. We have the following lemma:

Lemma 1. *e is non-monochromatic if and only if*

$$\sum_{i\in e}\sum_{j\in e\setminus\{i\}} M_{ij} \leq (q-1)(q-2) + 2\alpha(q-1).$$

Proof. If e is monochromatic, $\sum_{i\in e}\sum_{j\in e\setminus\{i\}} M_{ij} = q(q-1) > (q-1)(q-2) + 2\alpha(q-1)$. If e is not monochromatic, the vertices in e are grouped into $p \geq 2$ subsets (colors). If $p > 2$, we can notice that merging groups always increases the value of the double sum in the lemma; for $p = 2$, an exchange argument shows that this value increases as the sizes of the groups get more ill-balanced. So, the partition of the vertices of e that has the highest double sum value is the case of a group with $(q-1)$ vertices and a singleton, whose value is $(q-1)(q-2)+2\alpha(q-1)$. □

To obtain the following theorem, we only replace, in the expression of Graph Coloring from Theorem 2 the non-monochromaticity constraint on the edges by the non-monochromaticity constraint on the hyperedges we have exhibited in Lemma 1, to get an equivalent *SDP* formulation of Hypergraph Coloring.

Theorem 3. *The chromatic number of hypergraph H is:*

$$\chi(H) = 1 - \frac{1}{\alpha^*}$$

where

$$\alpha^* = \min \alpha$$

s.t.

$$\begin{aligned} M_{ii} &= 1 && \forall i \in V \\ \sum_{i\in e}\sum_{j\in e\setminus\{i\}} M_{ij} &\leq (q-1)(q-2) + 2\alpha(q-1) && \forall e \in H,\ \ q = |e| \\ M &\succeq 0 && \\ M_{ij} &\in \{\alpha; 1\} && \forall (i,j) \in V^2. \end{aligned}$$

Relaxing the discretization constraint gives us k, a lower bound on the chromatic number of the hypergraph, so on the optimum for K_n-Coloring; as well as for Graph Coloring SDP relaxation, we get a Semi-Definite Positive matrix that we factorize to obtain $N = |V|$ unit vectors u_i. In [6], a primalixation step for Graph Coloring was presented. We now see how to adapt it to the case of Hypergraph Coloring.

Probabilistic primalization procedure

For the hypergraph $H(V, E)$ we denote Q the maximal dimension of the hyperedges (dimension of the hypergraph). We denote d_i^q the degree of vertex i considering only hyperedges of dimension q. We can also define the notions of minimal, mean, maximal degrees at dimension q.

Moreover, we will denote by $\mathcal{E}$ the expectation of a random variable and by $\mathcal{P}$ a probability, and define:

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-\frac{x^2}{2}}$$

and

$$\mathcal{N}(x) = \int_x^{+\infty} \Phi(t) \mathrm{d}t.$$

We can approximate $\mathcal{N}(x)$:

$$\forall x > 0 \quad \Phi(x) \left(\frac{1}{x} - \frac{1}{x^3} \right) < \mathcal{N}(x) < \Phi(x) \frac{1}{x} \cdot$$

Given N unit vectors u_i obtained as above; let c be a threshold number. We randomly choose a vector r with the normal distribution of $\Re^N$. We note $C = \{i | u_i^T.r \geq c\}$ (set of captured vectors), $N' = |C|$ (number of captured vectors) and $m' = |\{e \in E : \forall i \in e,\ i \in C\}|$ (number of hyperedges entirely captured). We will remove from C one vertex for every entirely captured hyperedge, so at most m'. So, we want the quantity $N' - m'$ to be large enough to obtain a "good" hypergraph coloring. The following lemma bounds the expectation of $N' - m'$.

Lemma 2. *Let us denote:*

$$\tilde{\Delta} = \sum_{q=2}^{Q} \frac{\Delta_q}{q a_q}, \ \textit{where} \ \begin{cases} \Delta_q \geq & \textit{mean degree at dim. } q, \\ a_q = & \sqrt{\frac{k-1}{[1-2(1/q-1/q^2)]k-1}}. \end{cases}$$

For any threshold c:

$$\mathcal{E}(N' - m') \geq N \left[\mathcal{N}(c) - \sum_{q=2}^{Q} \frac{\Delta_q}{q} \mathcal{N}(a_q c) \right].$$

Proof. First, we can notice that r has the normal distribution in $\Re^N$, which is equivalent to: $\forall (e_1, \ldots, e_N)$ orthonormal basis of $\Re^N$, the variables defined by $(e_i^T.r)$ are independent and normally distributed.

Thus, the probability for a vector to be captured is $\mathcal{P}(r^T.u_i \geq c) = \mathcal{N}(c)$ since $r^T.u_i$ is normally distributed. So, we get:

$$\mathcal{E}(N') = N\mathcal{N}(c).$$

Let us separate the captured hyperedges according to their dimensions: $m' = m'_2 + \ldots + m'_Q$. Let e be a hyperedge of dimension q; we can (w.l.o.g.) suppose that $e = \{v_1, \ldots, v_q\}$. The probability for e to be entirely captured is:

$$\begin{array}{rcl}\mathcal{P}(e \text{ captured}) & = & \mathcal{P}\left(u_1^T.r \geq c, \ldots, \text{ and } u_q^T.r \geq c\right) \\ & \leq & \mathcal{P}\left((u_1 + \ldots + u_q)^T.r \geq qc\right) \\ & = & \mathcal{P}\left(\left[\dfrac{u_1 + \ldots + u_q}{||u_1 + \ldots + u_q||}\right]^T .r \geq \dfrac{qc}{||u_1 + \ldots + u_q||}\right) \\ & = & \mathcal{N}\left(\dfrac{qc}{||u_1 + \ldots + u_q||}\right).\end{array}$$

$$\begin{array}{rcl}\frac{1}{q}||u_1 + \ldots + u_q|| & = & \frac{1}{q}\sqrt{u_1^2 + \ldots + u_q^2 + \sum_{i \neq j} u_i.u_j} \\ & \leq & \frac{1}{q}\sqrt{q + (q-1)(q-2) + 2(q-1)\alpha} \\ & = & \sqrt{\dfrac{[1 - 2(1/q - 1/q^2)]k - 1}{k-1}} \\ & = & 1/a_q.\end{array}$$

Hence $\mathcal{P}(e \text{ captured}) \leq \mathcal{N}(a_q c)$.

Let us notice that: $\forall q \quad a_2 \geq a_q \geq a_{q+1} \geq a_Q > 1$.

We denote $m_q = |E_q|$ and δ_q the mean degree at dimension q. So, $N\delta_q = qm_q$. Let Δ_q be such that $\Delta_q \geq \delta_q$. Then we have:

$$\mathcal{E}(m'_q) = m_q \mathcal{P}(e_q \text{ captured}) \leq N\frac{\Delta_q}{q}\mathcal{N}(a_q c).$$

Hence

$$\mathcal{E}(N' - m') \geq N\left[\mathcal{N}(c) - \sum_{q=2}^{Q}\frac{\Delta_q}{q}\mathcal{N}(a_q c)\right] = F(c). \qquad \square$$

For any instance of problems, we can calculate $\tilde{\Delta}$ and a_Q so as to express $F(c)$ explicitly and compute a c maximizing F. Alternatively, we will show how to choose c so as to have: $\mathcal{N}(c) > 2\sum_{q=2}^{Q}\frac{\Delta_q}{q}\mathcal{N}(a_q c)$, leading to a positive value for the expectation.

Lemma 3. *For:*

$$c = \sqrt{\frac{[1 - 2(1/Q - 1/Q^2)]k - 1}{(1/Q - 1/Q^2)k}\ln[(2+\varepsilon)\tilde{\Delta}]} \quad (\mathcal{H}1)$$

with: Δ_q *and* ε *chosen so that:*

$$\min\left(1, 2\left(1-\sqrt{\frac{1/Q-1/Q^2}{1-2(1/Q-1/Q^2)}}\right)\right) > \varepsilon \geq 4(1/Q-1/Q^2)\frac{2k}{k-2}\frac{1}{\ln(2\tilde{\Delta})}, \quad (\mathcal{H}2)$$

we have $\mathcal{N}(c) > 2\sum_{q=2}^{Q}\frac{\Delta_q}{q}\mathcal{N}(a_q c)$.

Proof. Using the approximation inequalities for $\mathcal{N}$ and the fact that the a_q are decreasing down to $a_Q > 1$:

$$\begin{aligned}\frac{\mathcal{N}(c)}{\sum_{q=2}^{Q}\frac{\Delta_q}{q}\mathcal{N}(a_q c)} &\geq \frac{(1-1/c^2)e^{-c^2/2}}{\left(\sum_{q=2}^{Q}\frac{\Delta_q}{q a_q}\right)\mathrm{e}^{-a_Q^2 c^2/2}}\\ &\geq \frac{1-1/c^2}{\tilde{\Delta}}\mathrm{e}^{c^2(a_Q^2-1)/2}.\end{aligned}$$

To prove the lemma, we will show that, with the choice made in the hypotheses,

(i) $\frac{1}{c^2} \leq \frac{\varepsilon}{2} - \frac{\varepsilon^2}{4} \Rightarrow 1 - \frac{1}{c^2} \geq \frac{2}{2+\varepsilon}$, and

(ii) $\mathrm{e}^{c^2(a_Q^2-1)/2} = (2+\varepsilon)\tilde{\Delta}$.

We notice that $a_Q^2 - 1 = \frac{2(1/Q-1/Q^2)k}{[1-2(1/Q-1/Q^2)]k-1}$. Then, ($\mathcal{H}1$) is equivalent to ($ii$).

From ($\mathcal{H}1$) we can also derive:

$$\frac{1}{c^2} \leq (1/Q-1/Q^2)\frac{2k}{k-2}\frac{1}{\ln(2\tilde{\Delta})} \leq \varepsilon/4 \text{ from hypotheses.}$$

Choosing ε as in ($\mathcal{H}2$) we have: $\varepsilon < 1 \Rightarrow \frac{\varepsilon}{2} - \frac{\varepsilon^2}{4} > \frac{\varepsilon}{4}$. Then $\frac{1}{c^2} \leq \frac{\varepsilon}{4} < \frac{\varepsilon}{2} - \frac{\varepsilon^2}{4}$, hence ($i$) is verified.

Combining (i) and (ii), we obtain $\frac{1-1/c^2}{\tilde{\Delta}}\mathrm{e}^{c^2(a_Q^2-1)/2} \geq 2$, hence the lemma. □

Now, we can express the theorem:

Theorem 4. *For:*

$$c = \sqrt{\frac{[1-2(1/Q-1/Q^2)]k-1}{(1/Q-1/Q^2)k}\ln[(2+\varepsilon)\tilde{\Delta}]}$$

with:

- $$\tilde{\Delta} = \sum_{q=2}^{Q} \frac{\Delta_q}{q a_q}, \ \textit{where} \ \begin{cases} \Delta_q \geq & \textit{mean degree at dim. } q, \\ a_q = \sqrt{\dfrac{k-1}{[1-2(1/q-1/q^2)]k-1}} \end{cases}$$
- Δ_q *and* ε *chosen so that:*

$$\min(1, 2(1-\sqrt{\frac{1/Q-1/Q^2}{1-2(1/Q-1/Q^2)}})) > \varepsilon \geq 4(1/Q-1/Q^2)\frac{2k}{k-2}\frac{1}{\ln(2\tilde{\Delta})},$$

we have:

$$\mathcal{E}(N'-m') \geq ((2+\varepsilon)\tilde{\Delta})^{-\frac{[1-2(1/Q-1/Q^2)]k-1}{2(1/Q-1/Q^2)k}} \frac{N}{2\sqrt{2\pi}} \frac{1/Q-1/Q^2}{1-2(1/Q-1/Q^2)\sqrt{\ln((2+\varepsilon)\tilde{\Delta})}}.$$

Proof. Applying both lemmas 2 and 3, we have:

$$\mathcal{E}(N'-m') \geq N\frac{\mathcal{N}(c)}{2}.$$

We recall that

$$\mathcal{N}(c) \geq \Phi(c)\left(\frac{1}{c}-\frac{1}{c^3}\right)$$

with

$$\Phi(c) = \frac{1}{\sqrt{2\pi}}((2+\varepsilon)\tilde{\Delta})^{-\frac{[1-2(1/Q-1/Q^2)]k-1}{2(1/Q-1/Q^2)k}},$$

from hypotheses

$$\frac{1}{c}-\frac{1}{c^3} \geq \frac{1}{c}(1-\frac{\varepsilon}{2}) \geq \frac{1/Q-1/Q^2}{1-2(1/Q-1/Q^2)}\frac{1}{\sqrt{\ln(2+\varepsilon)\tilde{\Delta}}}.$$

Hence, with the hypotheses we finally get:

$$\mathcal{E}(N'-m') \geq \frac{N}{2}\frac{1/Q-1/Q^2}{1-2(1/Q-1/Q^2)}\frac{1}{\sqrt{2\pi\ln((2+\varepsilon)\tilde{\Delta})}}[(2+\varepsilon)\tilde{\Delta}]^{-\frac{[1-2(1/Q-1/Q^2)]k-1}{2(1/Q-1/Q^2)k}}.$$

□

This way, if we realize the expected values for each center, we can iterate the process as in [6], to finally obtain a

$$O\left(\frac{1-2(1/Q-1/Q^2)}{1/Q-1/Q^2}\ln((2+\varepsilon)\tilde{\Delta})((2+\varepsilon)\tilde{\Delta})^{\frac{[1-2(1/Q-1/Q^2)]k-1}{2(1/Q-1/Q^2)k}}\right)$$

coloring of the hypergraph.

6. Implementation and results

To implement these schemes, we used a Semidefinite Positive resolution code by Alizadeh *et al.*: SDPPack[4] [1]. Cost functions should be linear, constraints should be either equalities or Semi-Definite positiveness of a symmetric matrix, which is the form used in the sections above. This code is also especially designed for sparse matrices, hence it is particularly interesting since in our applications, the real graphs we are confronted to are generally not dense at all for topological reasons. We restricted ourselves to the "triset subgraph", the subgraph induced by the vertices of the triangles in T_3; clearly, optimizing the K_n-Coloring number on such a subgraph is equivalent to optimizing on the whole graph.

We now present our results on benchmark graphs (provided by the CELAR), which are typical of real situations encountered with communication networks of land forces, and we compare our results to those obtained with a more traditional simulated annealing approach. The problem considered here is, so, a K_3-coloring.

Both graphs presented here have 125 vertices (those networks correspond to 1049 vehicles) and the same triset initial structure, with 22 trisets. They differ in the definition of cosite links. Thus, the edge structures of the examples are slightly different:

	Benchmark 1	Benchmark 2
#vertices	125	125
#edges	299	376
#trisets of initial graph	22	22
#vertices of triset subgraph	28	28
#edges of triset subgraph	57	72

Defaix enabled us to compare the characteristics of the final frequency assignment obtained by proceeding the assignment procedure on the graph colored by different methods: "χ" means that we used a coloration optimal on the number of colors. "SDP" means that we used the SDP K_3-Coloring scheme presented here (we mention between parentheses the lower bound obtained through the hypergraph coloring scheme). "SA" a K_3-Coloration obtained by a Simulated Annealing approach. "$\emptyset$" means that we did not pre-color the graph at all. We obtained the following results, first on the pre-coloring phase, then on the final assignment:

[4]We thank the authors for the quasi on-line help they provided us during our experiments.

	Benchmark 1				Benchmark 2			
Col. method	χ	SDP	SA	$\emptyset$	χ	SDP	SA	$\emptyset$
#trisets [bound]	3	3[3]	3	22	6	3[3]	3	22
#colors	6	11	14	125	8	9	16	125
CPU time (min)	1	40	63	0	1	35	71	0
TEB saturation time (min)	4	98	136	[56]	61	53	131	[70]
min. bandwidth	122	54	34	(11)	59	110	48	(21)
mean bandwidth	337	338	379	(800)	321	371	366	(851)
mean overlap (%)	11.1	16.8	21.2	(47.5)	16.1	15.3	20.0	(47.7)

Without pre-coloring, we do not even get all T.E.B. (electromagnetic measures of interferences) bounded by a certain value, that we set as our constraints on assignment, within a 24 hour CPU time. SDP relaxation and Simulated Annealing approach give assignments of the same quality. In the case of Benchmark 1, the chromatic number approach leads us to an optimal K_n-coloration, thus to a good assignment, but with a low overlapping rate because of higher constraints between colors. Unfortunately though, a larger benchmark (250 vertices) was too large to implement the algorithm on it.

Moreover, bounds obtained by coloring the hypergraph defined in Proposition 8 or restrictions of the hypergraph to a fixed degree (in particular, a graph) are unefficient in a Branch-And-Bound procedure, even when they are close to the optimum, because of their slow calculations. In fact, preparing the data is a more costly step than solving the SDP problem ; finding an implemental procedure which modifies the matrix of constraints from node to node rather than computing it from scratch at each node, is likely to dramatically accelerate the whole process.

Concluding remarks

Our study aimed at pre-treating data for the assignment of frequency hopsets, which raises out a new family of coloring problems, generalizing the chromatic number problem. Though more theoretical than Simulated Annealing approaches, SDP relaxation and primalization seem to be particularly well suited to offer a good trade-off between the quality of feasible solutions found, the duality gap accuracy (between lower and upper bounds), and computational time.

Furthermore, we have presented here a generalization of Karger *et al.* probabilistic approach, to the case of Hypergraph Coloring. In [10] a derandomization of this approach is shown, but since the dimensions of the hypergraphs here are not bounded, their proof cannot be directly adapted. Derandomizing our algorithm still remains an open issue.

Some other criteria have not been introduced yet, such as the number of networks per color and the level of resulting constraints between colors (hopsets) improving the communication quality required by military needs. Moreover, an other priority will be to adapt our algorithm to larger sizes to anticipate the complexification of deployments.

References

[1] F. Alizadeh, J.-P. Haeberly, M.V. Nayakkankuppam and M.L. Overton, *SDPPack User's Guide, version 0.8 beta*. Technical report, NYU Computer Science Dept. (1997) 30 p. URL: http://www.cs.nyu.edu/phd_students/madhu/sdppack/sdppack.html

[2] T. Defaix, *Optimisation stochastique et allocation de plans de fréquences pour des réseaux à évasion de fréquences*. Thèse de doctorat, Université de Rennes 1 (1996) 175 p.

[3] U. Feige and J. Kilian, Zero Knowledge and the Chromatic Number, in *Proc. of the 11th Annual IEEE Conference in Computing Complexity*, Preliminary Version (1996) 278-287.

[4] A. Frieze and M. Jerrum, Improved approximation algorithms for MAX k-cut and MAX BISECTION, in *Proc. of the Fourth MPS Conference on Integer Programming and Combinatorial Optimization*. Springer-Verlag (1995) Paper Version: 21 p.

[5] M.X. Goemans and D.P. Williamson, Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming. *J. ACM* **42** (1995) 1115-1145.

[6] D. Karger, R. Motwani and M. Sudan, Approximate graph coloring by semidefinite programming. *J. ACM* **45** (1998) 246-265.

[7] M. Krivelevich and B. Sudakov, *Approximate coloring of uniform hypergraphs*. DIMACS Technical Report 98–31 (1998) 15 p.

[8] L. Lovász, On the Shannon capacity of a graph. *IEEE Trans. Inform. Theory* **IT-25** (1979) 1-7.

[9] C. Lund and M. Yannakakis, On the hardness of approximating minimization problems. *J. ACM* **41** (1994) 960-981.

[10] S. Mahajan and H. Ramesh, Derandomizing semidefinite programming based approximation algorithms, in *Proc. of the 36th Annual IEEE Symposium on Foundations of Computer Science* (1995) Paper Version: 19 p.