

# A branch and bound method for the aggregation of symmetric relations

Irène Charon\* and Olivier Hudry

Telecom ParisTech,  
46, rue Barrault, 75634 Paris Cedex 13, France  
{Irene.Charon,Olivier.Hudry}@telecom-paristech.com  
<http://www.infres.enst.fr/~hudry>

**Abstract.** We consider the problem of the aggregation of symmetric relations into a median equivalence relation, with in particular two special cases: the one for which the symmetric relations are equivalence relations (Régnier’s problem), and the one of the approximation of one symmetric relation by an equivalence relation (Zahn’s problem). These problems arise for instance from the field of classification or clustering, when one wants to gather objects in such a way that the objects of any cluster can be considered as similar, while the objects of different clusters can be considered as dissimilar. We first state this problem as a clique partitioning problem in graph theory. As this problem is NP-hard, we then design a branch and bound algorithm to solve this problem, based on a Lagrangean relaxation method for the evaluation function and on noising methods for the initial bound.

**Keywords:** Branch and bound method, Lagrangean relaxation, noising methods, symmetric relations, equivalence relations, median relations, clustering.

## 1 Introduction

We consider here the problem of the approximation of  $m$  symmetric relations defined on a same finite set  $X$  into a so-called *median equivalence relation* (see below and [2]), with in particular two special cases: the one for which the  $m$  symmetric relations are equivalence relations (Régnier’s problem [7]), and the one of the approximation of only one symmetric relation ( $m = 1$ ) by an equivalence relation (Zahn’s problem [9]). These problems arise for instance from the field of classification or clustering: in this case,  $X$  is a set of entities (which can be objects, people, projects, propositions, alternatives, and so on) that we want to gather in subsets of  $X$  in such a way that the elements of any such subset can be considered as similar while the objects of different subsets can be considered as dissimilar. Each symmetric relation is associated with a criterion specifying, for any pair  $\{x, y\}$  of entities, whether  $x$  and  $y$  are similar or not. Then we try to

---

\* Research supported by the ANR project “Computational Social Choice” ANR-09-BLAN-0305-03.

find the best compromise between all these criteria. This leads us, in Section 2, to state this problem as a graph theoretical problem, that we call CPP for *clique partitioning problem*. As this problem is NP-hard, we design in Section 3 a branch and bound algorithm to solve this problem, based on a Lagrangean relaxation method for the evaluation function and on noising methods for the initial bound.

## 2 The clique partitioning problem

The problem that we consider here can be mathematically described as follows. We are given a collection  $\Pi = (S_1, S_2, \dots, S_m)$  of  $m$  symmetric binary relations  $S_k$ ,  $1 \leq k \leq m$ , all defined on a same finite set  $X$  of  $n$  elements (Régnier's problem [7] corresponds to the case for which all the relations  $S_k$  are equivalence relations; Zahn's problem [9] corresponds to the case for which  $m$  is equal to 1). We consider the number  $\delta(R, S)$  of disagreements between two binary relations  $R$  and  $S$ :

$$\delta(R, S) = |\{(i, j) \in X^2 \text{ with } [iRj \text{ and not } iSj] \text{ or } [iSj \text{ and not } iRj]\}|.$$

Then, for any equivalence relation  $E$ , we consider the *remoteness*  $\Delta(\Pi, E) = \sum_{k=1}^m \delta(S_k, E)$ , measuring the total number of disagreements between  $\Pi$  and  $E$ . Our problem thus consists in computing an equivalence relation  $E^*$ , called a *median equivalence relation* of  $\Pi$ , which minimizes  $\Delta$  over the set  $\mathcal{E}$  of all the equivalence relations defined on  $X$ :

$$\Delta(\Pi, E^*) = \min_{E \in \mathcal{E}} \Delta(\Pi, E).$$

The computation of  $E^*$  is NP-hard [8], and remains so even for Régnier's problem [1], [5] or for Zahn's problem [6].

To state this problem as a 0-1 linear programming problem, let  $s^k = (s_{ij}^k)_{(i,j) \in X^2}$  ( $1 \leq k \leq m$ ) be the binary matrix defined by:  $s_{ij}^k = 1$  if  $iS_kj$  (i.e. if  $i$  and  $j$  are put together by  $S_k$ ), and  $s_{ij}^k = 0$  otherwise. Similarly, let  $(x_{ij})_{(i,j) \in X^2}$  denote the matrix associated with  $E$ :  $x_{ij} = 1$  if  $iEj$ ,  $x_{ij} = 0$  otherwise. It is easy to obtain the following:

$$\delta(S_k, E) = \sum_{(i,j) \in X^2} |s_{ij}^k - x_{ij}| = \sum_{(i,j) \in X^2} (s_{ij}^k - x_{ij})^2 = \sum_{(i,j) \in X^2} (s_{ij}^k + (1 - 2s_{ij}^k)x_{ij})$$

because of the binary property of the quantities  $s_{ij}^k$  and  $x_{ij}$ . Then we obtain, for the remoteness:

$$\Delta(\Pi, E) = \sum_{k=1}^m \sum_{(i,j) \in X^2} s_{ij}^k + \sum_{k=1}^m \sum_{(i,j) \in X^2} (1 - 2s_{ij}^k)x_{ij} = C + \sum_{(i,j) \in X^2} w_{ij}x_{ij}$$

where  $C = \sum_{k=1}^m \sum_{(i,j) \in X^2} s_{ij}^k$  is a constant and with, for  $(i, j) \in X^2$ :

$$w_{ij} = \sum_{k=1}^m (1 - 2s_{ij}^k) = m - 2|\{k \text{ with } 1 \leq k \leq m \text{ and } iS_kj\}|.$$

So, minimizing  $\Delta(\Pi, E)$  is the same as minimizing  $\sum_{(i,j) \in X^2} w_{ij}x_{ij}$ . Moreover, the constraints to state that  $E$  must belong to  $\mathcal{E}$  are the following:

- symmetry:  $\forall (i, j) \in X^2, x_{ij} = x_{ji}$ ;
- transitivity:  $\forall (i, j, h) \in X^3$  with  $i \neq j \neq h \neq i, x_{ij} + x_{jh} - x_{ih} \leq 1$ .

If we add the binary constraints:  $\forall (i, j) \in X^2, x_{ij} \in \{0, 1\}$ , we obtain the statement of the problem as a 0-1 linear programming problem.

We now may state this problem as a graph theoretic one. For this, we associate the complete graph  $K_n$  to  $\Pi$ , and we weight every edge  $\{i, j\}$  of  $K_n$  by  $w_{ij}$ . Then the variables  $x_{ij}$  equal to 1 define disjoint cliques (i.e. complete subgraphs) of  $K_n$ , and the value of  $\Delta(\Pi, E)$  is equal to the sum of the weights of the edges with both extremities inside a same clique. Hence our clique partitioning problem CPP. Note that the weights of the edges can be non-positive or non-negative integers. Moreover, the number of cliques into which we want to partition  $K_n$  is not given. Finally, CPP can be stated as follows: given a complete graph  $K_n = (X, A)$  whose edges  $\{i, j\}$  are weighted by non-positive or non-negative integers  $w_{ij}$ , partition  $X$  into  $p$  subsets  $X_1, X_2, \dots, X_p$ , where  $p$  is not given, so that  $\sum_{h=1}^p \sum_{(i,j) \in (X_h)^2} w_{ij}$  (i.e. the sum of the weights of the edges inside the cliques) is minimum.

### 3 The branch and bound method

To solve CPP, we design a branch and bound method BB. We briefly depict the main ingredients of BB.

The initial bound is provided by a metaheuristic, namely the *noising methods* [3], [4]. The noising methods usually compute very good solutions, quite often optimal, though we cannot know whether these solutions are indeed optimal.

The BB-tree is built as follows. The vertices  $v_i$  of  $K_n$  are integers belonging to  $\{1, 2, \dots, n\}$ . A partition with  $p$  subsets  $X_1, X_2, \dots, X_p$  is represented as:

$$\underbrace{v_1, v_2, \dots, v_{q_1}}_{X_1} \mid \underbrace{v_{q_1+1}, v_{q_1+2}, \dots, v_{q_2}}_{X_2} \mid \dots \mid \underbrace{v_{q_{p-1}+1}, v_{q_{p-1}+2}, \dots, v_{q_p}}_{X_p}$$

With such an encoding, a partition admits several representations. To avoid this, we suppose that the vertices are ordered by increasing value within a subset and subsets are ordered according to their smallest vertices; with the above notation, it means that we have:  $1 = v_1 < v_2 < \dots < v_{q_1}, v_{q_1+1} < v_{q_1+2} < \dots < v_{q_2}, \dots, v_{q_{p-1}+1} < v_{q_{p-1}+2} < \dots < v_{q_p}$ , and  $v_1 < v_{q_1+1} < \dots < v_{q_{p-1}+1}$ .

The subsets are progressively constructed. A node  $N$  of the BB-tree corresponds to the beginning of a partition encoding, something like:

$$\underbrace{v_1, v_2, \dots, v_{q_1}}_{X_1} \mid \underbrace{v_{q_1+1}, v_{q_1+2}, \dots, v_{q_2}}_{X_2} \mid \dots \mid \underbrace{v_{q_{h-1}+1}, v_{q_{h-1}+2}, \dots, v_{q_{h-1}+t}}_{X_h}$$

We extend  $N$  by at most  $n - q_{h-1} - t + 1$  new branches. The first branch is obtained by closing the current subset  $X_h$  and by creating a new subset  $X_{h+1}$

which will contain at least  $v_{q_{h-1}+t+1}$ . The other branches correspond with the possibilities to expand the current class  $X_h$  by adding an extra vertex (greater than  $v_{q_{h-1}+t}$ ) to it:  $v_{q_{h-1}+t+1}$ , or  $v_{q_{h-1}+t+2}$  but not  $v_{q_{h-1}+t+1}$ , or  $v_{q_{h-1}+t+3}$  but neither  $v_{q_{h-1}+t+1}$  nor  $v_{q_{h-1}+t+2}$ , and so on...

Three evaluation functions  $F_1, F_2, F_3$  are designed to evaluate the quality of every node  $N$  of the BB-tree. They can be split into two parts. The first part is the same for the three functions: it takes into account the contribution of the vertices already dispatched inside the subsets of the partition under construction associated with  $N$ ; for this, we only sum the weights of the edges with both extremities in a same subset. The second part depends on the function. For  $F_1$ , we add all the negative weights of the edges with at least one extremity greater than  $v_{q_{h-1}+t}$ . In  $F_2$ , we sharpen the design of  $F_1$  by considering some triples of vertices (triangles)  $\{a, b, c\}$  and by noting that if the weights of the edges between  $a, b$  and  $c$  have not the same sign, then the contribution of  $\{a, b, c\}$  cannot be the sum of the negative edges, as in  $F_1$ ; we design a greedy algorithm to choose these triangles in order to improve  $F_1$  as much as possible. The last function,  $F_3$ , is the most sophisticated. It is based on the Lagrangean relaxation of the transitivity constraints (see above).

Other ingredients, not described here, allow us also to cut branches of the BB-tree. During the talk, we will discuss the efficiency of the evaluation functions and of the other ingredients, based on experiments dealing with different kinds of graphs: instances of Régnier's problem or of Zahn's problem, instances coming from the literature, random instances, or instances with special combinatorial or algorithmic properties.

## References

1. J.-P. Barthélemy, B. Leclerc: The median procedure for partitions, in Partitioning data sets, I.J. Cox, P. Hansen, B. Julesz (eds), DIMACS Series in Disc. Math. and Theoretical Computer Science 19, Amer. Math. Soc., Providence, RI, 1995, 3-34.
2. J.-P. Barthélemy, B. Monjardet: The median procedure in cluster analysis and social choice theory, *Mathematical Social Sciences* 1, 1981, 235-267.
3. I. Charon, O. Hudry: Noising methods for a clique partitioning problem, *Discrete Applied Mathematics* 154 (5), 2006, 754-769.
4. I. Charon, O. Hudry: Self-tuning of the noising methods, *Optimization* 58 (7), 2009, 1-21.
5. O. Hudry: NP-hardness of Régnier's problem in classification, to appear in *Mathematics and Social Sciences*.
6. M. Krivanek, J. Moravek: NP-hard problems in hierarchical-tree clustering, *Acta Informatica* 23, 1986, 311-323.
7. S. Régnier: Sur quelques aspects mathématiques des problèmes de classification automatique, *I.C.C. Bulletin* 4, 175-191. Reprint: *Mathématiques et Sciences humaines* 82, 1983, 13-29.
8. Y. Wakabayashi: The Complexity of Computing Medians of Relations, *Resenhas*, 3 (3), 1998, 323-349.
9. C.T. Zahn: Approximating symmetric relations by equivalence relations, *SIAM Journal on Applied Mathematics*, 12, 1964, 840-847.