



Alexandru Ioan Cuza University of Iași
Faculty of Computer Science
Str. General Berthelot 16, Iași 700483, România
Tel. +40-32-201090, email: tr@info.uaic.ro

**T
E
C
H
N
I
C
A
L

R
E
P
O
R
T**



**A Branch and Bound Algorithm for
Coalition Structure Generation over
Graphs**

**Emanuel Florentin Olariu,
Cristian Frăsinaru,
Albert Abel Policiuc**

TR 20-03, April 2020

ISSN 2668-1765

ISSN-L 1224-9327

A Branch and Bound Algorithm for Coalition Structure Generation over Graphs

Emanuel Florentin OLARIU, Cristian FRĂȘINARU, Albert Abel
POLICIUC

Faculty of Computer Science, Alexandru Ioan Cuza University Iași,
General Berthelot 16, 700483 Iași, Romania,
Email: olariu@info.uaic.ro, acf@info.uaic.ro,
abel.policiu@info.uaic.ro

Abstract

We give a column generation based branch and bound algorithm for coalition structure generation over graphs problem using valuation functions for which this problem is proven to be NP-complete. For a given graph $G = (V, E)$ and a valuation function $w : 2^V \rightarrow \mathbb{R}$, the problem is to find the most valuable coalition structure (or partition) of V . We consider two cases: first when the value of a coalition is the sum of the weights of its edges which can be positive or negative, second when the value of a coalition takes account of both inter- and intra-coalitional disagreements and agreements, respectively. For both valuations we give experimental results which cover for the first time sets of more than forty agents.

For another valuation function (coordination) we give only the theoretical considerations in the appendix.

Keywords: *coalition structure generation, linear programming, quadratic programming, column generation, coalition valuation function.*

1 Introduction

Coalition structure generation (CSG) is a major problem in artificial intelligence ([13]), multi-agent systems ([2], [14]) communication networks, cooperative game theory ([3], [4], [12]), scheduling ([7]), economic theory (like combinatorial auctions) etc. Given a set of agents $V = \{1, 2, \dots, n\}$, and a *valuation function* $v : 2^V \rightarrow \mathbb{R}$ assigning a value to any coalition of agents, the problem is to partition the set of agents into pairwise disjoint *classes (coalitions)* such that the sum of their values is maximized. CSG is one of the steps in the coalition formation process [11] which may include also optimizing the coalitions performance and rewarding the coalitions value among the members.

CSG comes from real-world applications: consider a set of agents who can cooperate by working in coalitions. Some of them work better together while others find difficult to cooperate. The problem is to maximize the so called social well-fare or the total value of the designated coalitions. Classical cooperative game theory usually uses a super-additive valuation function that values better a merged coalition than the sum of of the values of the component coalitions. This leads to the great coalition formation which can be costly to coordinate and/or manipulate. Besides these considerations there are natural constraints on possible coalitions, hence the abstractions like super-additivity are not always appropriate for modeling the coalitions values and the agents should be divided into smaller coalitions.

The problem is computationally difficult: the input specification for all $2^{|V|}$ possible coalitions is intractable even for reasonable values of $|V|$ and the computational difficulty maintains even under quite restrictive assumptions. The literature presents a very large number of approaches to CSG problem like (see [10]): dynamic programming, meta heuristic methods, branch and bound algorithms based on dividing the searching space, anytime algorithms that maintain a monotonically improving feasible solution etc.

Our setting is that of (mixed) integer linear/quadratic programming and is based on the classical model of the set partitioning problem in a graph. Given a graph $G = (V, E)$ the vertices are agents and the edges represent connections between agents. We study two types of valuation functions: the *edge sum* and the *correlation* functions. The value of a coalition by the edge sum valuation function ([3]) is the sum of the weights of its edges (suppose we have a weight function defined on the set of edges); this function frequently occurs in communication networks and cooperative game theory. The correlation valuation function ([1]) is defined on the entire coalition

structure, although it can be defined on coalitions only, and occurs in the clustering framework. This function takes account of the agreements from inside and the disagreements from outside the structure - an edge being labeled with a plus or a minus depending on whether the involved agents are similar or different.

Our approach is based on solving an integer linear programming problem, which is equivalent with CSG, using a branch and bound algorithm in which the problems in nodes are built by means of column generation method. This approach works as long as the involved sub-problem can be solved. We proved that this can be done for edge sum and correlation valuation functions since the sub-problem becomes a quadratic knapsack problem with forbidden configurations. This variant of quadratic knapsack problem can be solved in practice using the corresponding integer quadratic programming model or the mixed integer linear programming equivalent model.

Our numerical results show that we can approach by this method sets of up to 45 agents depending on the magnitude of the weight function for edge sum valuation function and up to 40 agents for the correlation valuation function, using a regular home PC.

The paper is organized as follows: section 2 describes the LP model and the column generation framework, section 3 is dedicated to the two valuation functions outlining the corresponding sub-problems and their linearizations, section 4 describes the branch and bound algorithm, and the last section contains the numerical results and the conclusions. The appendix contains ..

2 LP model and column generation

Consider $V = \{1, 2, \dots, n\}$ to be a set of n agents and $v : 2^V \rightarrow \mathbb{R}$ a valuation function on the power set of V . A *coalition structure* is a collection of disjoint exhaustive subsets C_1, C_2, \dots, C_p , where $C_i \cap C_j = \emptyset$ for all $1 \leq i < j \leq p$ and $\bigcup_{i=1}^p C_i = V$.

The problem of finding a coalition structure of maximum value is

equivalent with the set partitioning problem (SPP) (see [7]):

$$\max \sum_{j=0}^{2^n-1} v(C_j)x_j \quad (1)$$

$$\sum_{j=0}^{2^n-1} a_{ij}x_j = 1, \forall i \in V, \quad (2)$$

$$x_j \in \{0, 1\}, \forall j \in \{0, 1, \dots, 2^n - 1\} \quad (3)$$

where $\{C_0, C_1, \dots, C_{2^n-1}\}$ is an enumeration of 2^V and $(a_{ij})_{i \in V}$ is the characteristic vector of C_j , for each j , that is

$$a_{ij} = \begin{cases} 1, & \text{if } i \in C_j \\ 0, & \text{otherwise} \end{cases}, \text{ for each } j \in \{0, 1, \dots, 2^n - 1\}. \quad (4)$$

We can relax the integrality constraints by replacing (3) with

$$x_j \geq 0, \forall j \in \{0, 1, \dots, 2^n - 1\} \quad (3')$$

Writing the original relaxed problem as a minimum one means to replace (1) by (1') (by ignoring the minus in front of min)

$$\min \left(\sum_{j=0}^{2^n-1} -v(C_j)x_j \right) \quad (1')$$

The dual of the problem (1'), (2), (3') is

$$\max \left(\sum_{j=1}^n \pi_j \right) \quad (5)$$

$$\sum_{i=1}^n a_{ij}\pi_i \leq -v(C_j), \forall j \in \{0, 1, \dots, 2^n - 1\} \quad (6)$$

$$\pi_i \in \mathbb{R}, \forall i \in V \quad (7)$$

Let $C_0 = \emptyset$, and $C_j = \{j\}$, for any $j \in \{1, 2, \dots, m\}$. An initial feasible basic solution to the problem (1'), (2), (3') could be that corresponding to the coalition structure $(\{1\}, \{2\}, \dots, \{m\})$, that is, x_1, x_2, \dots, x_m . Now we

restrict the problem to a small number of variables including x_1, x_2, \dots, x_m and we get the restricted master problem (RMP):

$$\min \left(\sum_{j \in J} -v(C_j)x_j \right) \quad (8)$$

$$\sum_{j \in J} a_{ij}x_j = 1, \forall i \in V, \quad (9)$$

$$x_j \geq 0, \forall j \in J \quad (10)$$

where $J = \{1, 2, \dots, m\} \cup J' \subseteq \{0, 1, \dots, 2^m - 1\}$.

The dual of the RMP is

$$\max \left(\sum_{i=1}^n \pi_i \right) \quad (11)$$

$$\sum_{i=1}^n a_{ij}\pi_i \leq -v(C_j), \forall j \in J \quad (12)$$

$$\pi_i \in \mathbb{R}, \forall 1 \leq i \leq n \quad (13)$$

Let $(\mathbf{x}, \boldsymbol{\pi})$ be an optimum primal-dual solution for this pair of problems; we look for a non-basic variable (column) with the minimum negative reduced cost (Dantzig rule) - by solving a corresponding sub-problem - that would be added to the current restricted master problem. When such a variable doesn't exist we can stop: we have an optimum solution to the primal problem. The sub-problem is to find

$$j_0 = \arg \max_{j \in \{0, 1, \dots, 2^n - 1\} \setminus J} \left(v(C_j) + \sum_{i=1}^n a_{ij}\pi_i \right) \quad (14)$$

The arising question is: how can we solve problem (14)? The answer depends on the form of the valuation function $v(\cdot)$. In the following sections we will analyze this sub-problem for two of the most frequent used valuation functions defined for coalition structure over graphs: the *edge-sum* and the *correlation* valuation functions.

3 Coalition Structure Generation over Graphs

3.1 The edge-sum valuation

Let $G = (V, E)$ be graph and $w : E \rightarrow \mathbb{R}$ a weight function on its edges. The corresponding *edge-sum coalition valuation function* is

$$v : \mathcal{P}(V) \rightarrow \mathbb{R}, v(C) = \sum_{ij \in E: i, j \in C} w_{ij}, \forall C \subseteq V.$$

This function was extensively studied in the context of cooperative game theory ([3]) and the corresponding CSG problem is NP-hard being as hard as the MAX-CUT problem.

Now, if \mathbf{v} is the characteristic vector of the generic coalition C , then the sub-problem becomes:

$$\max_{\mathbf{v} \in \{0,1\}^m, \mathbf{v} \neq \chi_{C_j}, \forall j \in J} \left(\sum_{ij \in E} w_{ij} v_i v_j + \sum_{i \in V} \pi_i v_i \right) > 0 \quad (15)$$

That is, the sub-problem is a quadratic knapsack problem with forbidden configurations (QKPF). We can find a new column to add to the restricted master problem if and only if this quadratic knapsack problem has a strictly positive optimal objective value.

QKPF could be a computationally difficult problem since the quadratic knapsack problem (QKP) is known to be NP-hard (being a generalization of *Clique* problem). By rephrasing it we get:

$$\max \left(\sum_{ij \in E} w_{ij} v_i v_j + \sum_{i \in V} \pi_i v_i \right) \quad (16)$$

$$\sum_{i \in C_j} v_i + \sum_{i \notin C_j} (1 - v_i) \geq 1, \forall j \in J \quad (17)$$

$$v_i \in \{0, 1\}, \forall i \in V \quad (18)$$

This problem has a quadratic objective but only linear constraints: an Integer Quadratic Programming (IQP) problem. Such a problem can be solved by using a mathematical optimization solver as such or by linearizing it first in order to transform it into a Mixed Integer Linear Programming (MILP) problem. We note here that the known methods used for solving such a problem are not applicable here since the usual requirements are to have

natural or positive coefficients for the quadratic (some times non-diagonal) terms ([5], [8]). Hence we had to settle to solve it as a general IQP problem using a mathematical optimization solver.

3.2 The correlation valuation

Suppose we have a function $e : E \rightarrow \{+, -\}$ that assigns to each edge $ij \in E$ the labels $+$ or $-$ depending on whether agents i and j are similar or different (this function arises in clustering frameworks). A valuation function that takes account of both inter- and intra-coalitional similarities can be defined in the following way ([1]): we want to minimize the number of mistakes: a positive mistake occurs when $e(ij) = -$, with i and j belonging to the same coalition, a negative one occurs when $e(ij) = +$, but i and j belong to different coalitions. Or, equivalently, we maximize the number of agreements; first define, for a coalition $C \subseteq V$, the intra- and inter-coalitional connections

$$\text{Intra}^+(C) = |\{ij \in E : e(ij) = +, i, j \in C\}|,$$

$$\text{Inter}^-(C) = |\{ij \in E : e(ij) = -, i \in C, j \notin C\}|.$$

The *correlation valuation function* is

$$v : \mathcal{P}(V) \rightarrow \mathbb{R}, v(C) = \text{Intra}^+(C) + \text{Inter}^-(C)/2, \forall C \subseteq V.$$

It was proven in ([1]) that the corresponding sub-problem is *NP*-complete. We define two weight functions on the set of edges of G , $w^+, w^- : E \rightarrow \{0, 1\}$, by

$$w^+(ij) = w_{ij}^+ = \begin{cases} 1, & \text{if } e(ij) = + \\ 0, & \text{if } e(ij) = - \end{cases},$$

$$w^-(ij) = w_{ij}^- = \begin{cases} 1, & \text{if } e(ij) = - \\ 0, & \text{if } e(ij) = + \end{cases}, \forall ij \in E.$$

Let \mathbf{v} be the characteristic vector of the generic coalition C , then

$$v(C) = \frac{1}{2} \sum_{i \in V} \sum_{j \in V} w_{ij}^+ v_i v_j + \frac{1}{2} \sum_{i \in V} \sum_{j \in V} w_{ij}^- v_i (1 - v_j).$$

The sub-problem (14) becomes the following QKPf problem:

$$\max \left(\frac{1}{2} \sum_{i \in V} \sum_{j \in V} w_{ij}^+ v_i v_j + \frac{1}{2} \sum_{i \in V} \sum_{j \in V} w_{ij}^- v_i (1 - v_j) + \sum_{i \in V} \pi_i v_i \right) \quad (19)$$

$$\sum_{i \in C_j} v_i + \sum_{i \notin C_j} (1 - v_i) \geq 1, \forall j \in J \quad (20)$$

$$v_i \in \{0, 1\}, \forall i \in V \quad (21)$$

3.3 The linearization of QKPf

We present here the linearization due to Glover ([6]) for solving QKP. First, we consider the following equivalent form of the objective function quadratic fragment $\sum_{i=1}^n \sum_{j=1}^n w'_{ij} v_i v_j$, where $2w'_{ij} = w_{ij}$, for each $i \neq j$, and $w'_{ii} = 0$, for

all $i \in V$. Second, $\sum_{j=1}^n w'_{ij} v_i v_j$ is replaced by a new real variable u_i , and we add four new constraints

$$\begin{aligned} W_i'^- v_i &\leq u_i \leq W_i'^+ v_i, \\ \sum_{j=1}^n w'_{ij} v_j - W_i'^+(1 - v_i) &\leq u_i \leq \sum_{j=1}^n w'_{ij} v_j - W_i'^-(1 - v_i), \end{aligned}$$

where $W_i'^-$ and $W_i'^+$ are a lower and, respectively, an upper bound for $\sum_{i=1}^n \sum_{j=1}^n w'_{ij} v_j$, e.g.:

$$W_i'^- = \sum_{j:w'_{ij}<0} w'_{ij} \quad \text{and} \quad W_i'^+ = \sum_{j:w'_{ij}>0} w'_{ij} \quad (22)$$

The problem (16) - (18) becomes, after replacing w'_{ij} , $W_i'^-$, and $W_i'^+$,

by correspondingly w_{ij} , W_i^- , and W_i^+ (and substituting $2u_i$ by u_i)

$$\begin{aligned}
& \max \left(\sum_{i \in V} \pi_i v_i + \sum_{i \in V} u_i \right) \\
& \sum_{i \in C_j} v_i + \sum_{i \notin C_j} (1 - v_i) \geq 1, \forall j \in J, \\
& \sum_{j:ij \in E} w_{ij} v_j + W_i^+ v_i - u_i \leq W_i^+, \forall i \in V, \\
& \sum_{j:ij \in E} w_{ij} v_j + W_i^- v_i - u_i \geq W_i^-, \forall i \in V, \\
& W_i^- v_i - u_i \leq 0, \forall i \in V, \\
& W_i^+ v_i - u_i \geq 0, \forall i \in V, \\
& v_i \in \{0, 1\}, \forall i \in V, \\
& u_i \in \mathbb{R}, \forall i \in V.
\end{aligned}$$

For the correlation valuation function the corresponding model can be linearized as in a similar way; replacing the coefficients of the linear fragment with $p_i = \frac{1}{2} \sum_{j \in V} w_{ij}^- + \pi_i$, $\forall i \in V$, we get the following linear formulation of the problem (19) - (21):

$$\begin{aligned}
& \max \left(\sum_{i \in V} p_i v_i + \sum_{i \in V} u_i \right) \\
& \sum_{i \in C_j} v_i + \sum_{i \notin C_j} (1 - v_i) \geq 1, \forall j \in J, \\
& \sum_{j:ij \in E} w'_{ij} v_j + W_i^+ v_i - u_i \leq W_i^+, \forall i \in V, \\
& \sum_{j:ij \in E} w'_{ij} v_j + W_i^- v_i - u_i \geq W_i^-, \forall i \in V, \\
& W_i^- v_i - u_i \leq 0, \forall i \in V, \\
& W_i^+ v_i - u_i \geq 0, \forall i \in V, \\
& v_i \in \{0, 1\}, \forall i \in V, \\
& u_i \in \mathbb{R}, \forall i \in V,
\end{aligned}$$

where $w'_{ij} = 0.5 \cdot (w_{ij}^+ - w_{ij}^-)$, for all $i, j \in V$, and the upper and lower bounds ($W_i'^+$ and $W_i'^-$) may have the same expressions like in (22).

These last problems are Mixed Integer Linear Programming (MILP) problems that have a reasonable size and can theoretically be solved by the means of a mathematical optimization solver. We presented these linear models for the sake of completeness, but the numerical results for them are, by now, - in terms of running time performance - not encouraging. The optimization solver we used performs better on the quadratic models.

4 Branch and Bound algorithm

Suppose that the current node in the branching tree has a subset of already covered agents, $U = \bigcup_{j \in J: x_j=1} C_j$, and $U' = V \setminus U$. For the edge-sum valuation the sub-problem becomes:

$$\max \left(\sum_{i \in U'} \sum_{j \in U'} w_{ij} v_i v_j + \sum_{i \in U'} \pi_i v_i \right) \quad (23)$$

$$\sum_{i \in C_j} v_i + \sum_{i \notin C_j} (1 - v_i) \geq 1, \forall j \in J, \text{ s. t. } C_j \subseteq U'. \quad (24)$$

$$v_i \in \{0, 1\}, \forall i \in U'. \quad (25)$$

For the correlation valuation function the sub-problem (14) becomes:

$$\max \left[\frac{1}{2} \sum_{i \in U'} \sum_{j \in U'} (w_{ij}^+ - w_{ij}^-) v_i v_j + \sum_{i \in U'} \left(\frac{1}{2} \sum_{j \in V} w_{ij}^- + \pi_i \right) v_i \right] \quad (26)$$

$$\sum_{i \in C_j} v_i + \sum_{i \notin C_j} (1 - v_i) \geq 1, \forall j \in J, \text{ s. t. } C_j \subseteq U'. \quad (27)$$

$$v_i \in \{0, 1\}, \forall i \in U'. \quad (28)$$

Both these problems have linearization variants.

At each node of the branching tree we first build the current LP relaxation in two steps: (1) we reduce the variables number by taking account of the branching variables along the path to the root, and (2) we add the necessary variables to the parent node LP relaxation by the means of column generation method.

Step (1) is implemented by effectively fixing to 0 the variables x_h such that $C_h \cap U \neq \emptyset$ (or, equivalently, $C_h \cap C_j \neq \emptyset$, for some branching variable x_j set to 1), and removing the branching variables x_i set to 0. The fixing procedure can be achieved by looking in each equation (2) that contains a branching variable set to 1. In this way any branching variable set to 1, that corresponds to a medium sized coalition, has the effect of drastically reducing the size of the corresponding mathematical programming model.

While step (1) is basically the same for both valuation functions, step (2) depends on the specific sub-problem. Step (2) consists in repeatedly solving the corresponding sub-problem while the optimum objective function value is strictly positive. The implementation, however, requires this value to be positive within some tolerance. If this step would not have been subjected to numerical restriction, then the algorithm would have been an exact method.

Our branching rule works in a classical way: we choose a variable, x_j , from the optimal solution in the current node of the branching tree such that its value is around 0.5 (e. g. $x_j \in (0.35, 0.65)$ - if possible). It is worth-noting that, for our specific problems, this enumerative method (branch and bound) doesn't need upper bounds because in almost all cases a feasible solution occurs very early - mostly after performing one of the first type (1) steps. The overall effect is that the branching tree has a medium size: we limited the number of branching tree size to 40, but this bound was hardly reached.

5 Numerical results

In this section we evaluate the performance of our algorithm. A major part of the running time of our algorithm is concentrated in the root node of the branching tree, where step (2) adds hundreds of new variables to the root model, hence hundreds of sub-problems to solve (fortunately Gurobi solver quickly finds optimal solutions to these QKPf problems), while for the other nodes of the tree the number of such sub-problems drastically reduces.

The algorithm has been written in Java and run on an Intel(R) Core (TM) i5-7500 CPU 3.40GHz computer with 8GB RAM, under Ubuntu 18.04.4 LTS. The linear and quadratic programming problems were solved using Gurobi 9.0 under an Academic License.

Since there are no benchmarks in the literature for edge-sum or correlation valuation functions, the algorithm has been tested on randomly generated problems. Our test problems were built using the Gilbert model, that is, each edge has a fixed probability of being present in the graph, inde-

pends of any other edges (probability 1 gives a complete graph). The weights on edges are independently generated using a Gaussian distribution $N(0, 0.2)$ (a certain similar valuation function occurs in [9]). The name of the benchmark file indicates the probability that an edge belongs to the graph ("p"), the number of agents ("n"), and the number of the instance ("s").

We slightly generalized the use of the correlation function by supposing that some edges don't have signs at all, that is, allowing an incomplete underlying graph. The random graphs were generated using the same model and an edge has a "+" or a "-" sign with a prescribed probability. The name of a benchmark file for this valuation indicates the probability that an edge belongs to the graph ("p"), the probability of the plus sign ("pSign"), the number of agents ("n"), and finally the number of the instance ("s").

Table 1 shows the results of our numerical tests; it contains the optimum value in the root (LP problem), the (most of the time) optimum value of the ILP problem found by branch and bound algorithm, the overall running time, the running time for finding and solving the LP problem in the root, the running time per node, the number of nodes of the branching tree, the total number of variables added during the execution, the number of nodes fathomed by integrality, and the gap between the LP and the ILP optimum values.

Our algorithm is one of the first to present competitive results for more than 20 agents for quite general valuation functions, using only a regular PC. For half of the tests the root LP problem has an integer solution and for all of them the gap is less than one percent. The branching trees (with few exceptions) are small and most of the running time is concentrated in the root and in the nodes corresponding to variables set to zero or variables set to one but associated with small sized coalitions. Conducting the branching rule to find medium or large sized coalitions variables seems to be of secondary interest since the resulting trees are small in size.

We listed only the results using the quadratic models for solving the sub-problem, however the linear models have a good potential which deserves future exploring work (Lagrangian relaxation, special treatment of the high degeneracy etc). Future work should also be directed towards getting rid of the main burden of our algorithm: find specific methods for solving QKPf. Another direction of future research could be a cut generation method (finding inequalities defining facets of a subjacent polyhedral relaxation) that tightens the LP relaxations towards a branch and cut algorithm.

Table 1: Solutions and computational times for different instances.

	Instance	Solution		CPU time (seconds)			Number of		int. sol.	Gap
		LP	ILP	overall	root	per node	nodes	variables		
Edge-sum valuation	p0.8n40s0	17.877	17.866	775	547	258.3	3	316	1	0.06%
	p0.8n40s1	21.495	21.353	1,833	1,027	122.2	15	849	1	0.66%
	p0.8n40s2	20.108	20.108	720	720	720.0	1	253	1	0.00%
	p0.8n40s3	19.327	19.290	812	650	162.4	5	320	1	0.19%
	p0.8n40s4	18.166	18.099	897	584	179.4	5	318	1	0.36%
	p0.8n45s0	22.606	22.411	6,672	3,565	166.8	40	1,598	5	0.86%
	p0.8n45s1	22.580	22.480	4,451	2,902	404.63	11	622	2	0.44%
	p0.8n45s2	21.485	21.432	2,943	1,805	196.2	15	799	2	0.24%
	p0.8n45s3	23.884	23.884	2,815	2,815	2,815.0	1	312	1	0.00%
	p0.8n45s4	21.500	21.500	4,472	4,472	4,472.0	1	284	1	0.00%
	p1.0n35s0	17.154	17.154	346	346	346.0	1	203	1	0.00%
	p1.0n35s1	16.427	16.383	446	344	89.2	5	273	1	0.26%
	p1.0n35s2	19.406	19.241	1,651	590	47.1	35	1,231	2	0.85%
	p1.0n35s3	15.500	15.388	630	398	90.0	7	334	2	0.72%
	p1.0n35s4	19.064	18.987	979	414	108.7	9	515	1	0.40%
	p1.0n40s0	25.286	25.286	3,860	3,860	3,860.0	1	297	1	0.00%
	p1.0n40s1	23.994	23.789	3,019	2,313	335.4	9	534	3	0.85%
	p1.0n40s2	22.401	22.401	2,376	2,376	2,376.0	1	262	1	0.00%
	p1.0n40s3	21.942	21.942	3,300	3,300	3,300.0	1	274	1	0.00%
	p1.0n40s4	18.342	18.228	3,545	1,655	154.1	23	854	4	0.62%
Correlation valuation	p0.6pS0.6n35s0	224.0	224	256	256	256.0	1	433	1	0.00%
	p0.6pS0.6n35s1	243.0	243	458	350	50.8	9	688	2	0.00%
	p0.6pS0.6n35s2	239.0	239	451	451	451.0	1	671	1	0.00%
	p0.6pS0.6n35s3	241.8	241	711	222	28.4	25	1,684	1	0.33%
	p0.6pS0.6n35s4	248.5	248	465	392	155	3	705	1	0.20%
	p0.6pS0.6n40s0	292.0	292	811	811	811.0	1	538	1	0.00%
	p0.6pS0.6n40s1	298.0	298	1,778	1,778	1,778.0	1	1,130	1	0.00%
	p0.6pS0.6n40s2	289.0	289	1,735	1,735	1,735.0	1	596	1	0.00%
	p0.6pS0.6n40s3	311.0	311	938	938	938.0	1	500	1	0.00%
	p0.6pS0.6n40s4	313.5	313	4,542	3,062	1,514	3	1,565	1	0.15%
	p0.8pS0.6n35s0	304.0	304	1,714	1,277	342.8	5	829	3	0.00%
	p0.8pS0.6n35s1	294.5	294	1,746	834	116.4	15	1,614	2	0.16%
	p0.8pS0.6n35s2	299.0	299	1,670	1,670	1,670.0	1	704	1	0.00%
	p0.8pS0.6n35s3	307.0	307	1,070	1,070	1,070.0	1	567	1	0.00%
	p0.8pS0.6n35s4	298.0	298	1,137	1,137	1,137.0	1	422	1	0.00%
	p0.8pS0.6n40s0	395.0	395	4,271	4,271	4,271.0	1	839	1	0.00%
	p0.8pS0.6n40s1	380.5	380	6,643	4,522	2,214.0	3	1,500	1	0.13%
	p0.8pS0.6n40s2	385.0	385	6,402	6,402	6,402.0	1	850	1	0.00%
p0.8pS0.6n40s3	394.0	394	9,791	9,791	9,791.0	1	897	1	0.00%	
p0.8pS0.6n40s4	387.0	387	7,278	7,278	7,278	1	734	1	0.00%	

A Appendix: The coordination valuation

Let $G = (V, E)$ be graph and $C \subseteq V$ be a coalition, we define

$$n_i(C) = |\{jk \in E : j \in C, k \notin C, ij, ik \in E\}|.$$

The *coordination coalition valuation function* ([13]) is

$$v : 2^V \rightarrow \mathbb{R}, v(C) = \sum_{i \in C} n_i(C), \forall C \subseteq V.$$

This function accounts for all cliques of three agents, two of them being inside the coalition, while the third is outside. It is designed to offer coalitions that include agents that have common neighbours from outside. We will generalize this definition by including the weight of the 3-clique as the sum of the weights on its edges. Hence, we consider first $w : E \rightarrow \mathbb{R}$ to be an weight on the edges of G . Second, if \mathbf{v} is the characteristic vector for the generic coalition C , then

$$v(C) = \sum_{i=1}^n v_i \sum_{j=1}^n w_{ij} v_j \sum_{k=1}^n w_{ik} w_{jk} (1 - v_k).$$

We get the coordination valuation function by taking w to be the edge characteristic function, i. e., $w_{ij} = 1$, if $ij \in E$, and 0 otherwise. The sub-problem (14) becomes:

$$\max \left(\sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n w_{ij} w_{ik} w_{jk} v_i v_j (1 - v_k) + \sum_{i \in V} \pi_i v_i \right) \quad (29)$$

$$\sum_{i \in C_j} v_i + \sum_{i \notin C_j} (1 - v_i) \geq 1, \forall j \in J \quad (30)$$

$$v_i \in \{0, 1\}, \forall i \in V \quad (31)$$

A.1 Reducing the degree of the objective function

Using the same method of Glover from ([6]), we will transform the cubic objective function from (29) to a quadratic one by adding only n new unknowns and $4n$ new (quadratic) constraints. We introduce first the variables

$$z_i = v_i \sum_{j=1}^n v_j \left(\sum_{k=1}^n w_{ij} w_{ik} w_{jk} (1 - v_k) \right), \forall i \in V.$$

Second, add the constraints

$$\begin{aligned}
& Y_i^- v_i \leq z_i \leq Y_i^+ v_i, \forall i \in V \\
& \sum_{j=1}^n v_j \left(\sum_{k=1}^n w_{ij} w_{ik} w_{jk} (1 - v_k) \right) - Y_i^+ (1 - v_i) \leq z_i, \forall i \in V \\
& \sum_{j=1}^n v_j \left(\sum_{k=1}^n w_{ij} w_{ik} w_{jk} (1 - v_k) \right) - Y_i^- (1 - v_i) \geq z_i, \forall i \in V,
\end{aligned} \tag{32}$$

where Y_i^- and Y_i^+ are a lower and, respectively, an upper bound for the expression

$$\sum_{j=1}^n v_j \left(\sum_{k=1}^n w_{ij} w_{ik} w_{jk} (1 - v_k) \right).$$

Y_i^- and Y_i^+ could be

$$\begin{aligned}
Y_i^- &= \left(\sum_{j,k:w_{ij}w_{ik}w_{jk}<0} w_{ij}w_{ik}w_{jk} \right) / 2 \\
Y_i^+ &= \left(\sum_{j,k:w_{ij}w_{ik}w_{jk}>0} w_{ij}w_{ik}w_{jk} \right) / 2
\end{aligned} \tag{33}$$

The problem (29) - (31) becomes

$$\max \left(\sum_{i=1}^n \pi_i v_i + \sum_{i=1}^n z_i \right) \tag{34}$$

$$\sum_{i \in C_j} v_i + \sum_{i \notin C_j} (1 - v_i) \geq 1, \forall j \in J, \tag{35}$$

$$\sum_{j=1}^n v_j \left(\sum_{k=1}^n w_{ij} w_{ik} w_{jk} (1 - v_k) \right) + Y_i^+ v_i - z_i \leq Y_i^+, \forall i \in V, \tag{36}$$

$$\sum_{j=1}^n v_j \left(\sum_{k=1}^n w_{ij} w_{ik} w_{jk} (1 - v_k) \right) + Y_i^- v_i - z_i \geq Y_i^-, \forall i \in V, \tag{37}$$

$$Y_i^- v_i - z_i \leq 0, \forall i \in V, \tag{38}$$

$$Y_i^+ v_i - z_i \geq 0, \forall i \in V, \tag{39}$$

$$v_i \in \{0, 1\}, \forall i \in V, \tag{40}$$

$$z_i \in \mathbb{R}, \forall i \in V. \tag{41}$$

(For the coordination valuation function we can simply take $Y_i^- = 0$ and $Y_i^+ = n^2$, for all $i \in V$.)

This last problem is a Mixed Integer Quadratically Constrained Programming (MIQCP) problem that could also be solved using a mathematical optimization solver. Another way of addressing this problem is to apply again the method of Glover to the expressions

$$z_{ij} = v_j \left(\sum_{k=1}^n w_{ij} w_{ik} w_{jk} (1 - v_k) \right),$$

which will add n^2 new variables and $4n^2$ new constraints to the problem.

A.2 Branch and Bound version for the coordination valuation function

Suppose as above that the current node in the branching tree has a subset of already covered agents, $U = \bigcup_{j \in J: x_j=1} C_j$, and $U' = V \setminus U$. For this valuation function the sub-problem (14) becomes:

$$\begin{aligned} \max \quad & \left[\sum_{i \in U'} \sum_{j \in U'} \left(\sum_{k \in U'} w_{ij} w_{ik} w_{jk} v_i v_j (1 - v_k) + \sum_{k \in U} w_{ij} w_{ik} w_{jk} v_i v_j \right) + \right. \\ & \left. + \sum_{i \in U'} \pi_i v_i \right] \end{aligned} \quad (42)$$

$$\sum_{i \in C_j} v_i + \sum_{i \notin C_j} (1 - v_i) \geq 1, \forall j \in J, \quad (43)$$

$$v_i \in \{0, 1\}, \forall i \in U'. \quad (44)$$

Using the above degree reduction procedure for the objective function we introduce the variables

$$z_i = v_i \sum_{j \in U'} v_j \left(\sum_{k \in U'} w_{ij} w_{ik} w_{jk} (1 - v_k) + \sum_{k \in U} w_{ij} w_{ik} w_{jk} \right), \forall i \in U'.$$

Second, add the constraints

$$\begin{aligned}
& Y_i^- v_i \leq z_i \leq Y_i^+ v_i, \forall i \in V \\
& \sum_{j \in U'} v_j \left(\sum_{k \in U'} w_{ij} w_{ik} w_{jk} (1 - v_k) + \sum_{k \in U} w_{ij} w_{ik} w_{jk} \right) - Y_i^+ (1 - v_i) \leq z_i, \forall i \in U' \\
& \sum_{j \in U'} v_j \left(\sum_{k \in U'} w_{ij} w_{ik} w_{jk} (1 - v_k) + \sum_{k \in U} w_{ij} w_{ik} w_{jk} \right) - Y_i^- (1 - v_i) \geq z_i, \forall i \in U',
\end{aligned}$$

where Y_i^- and Y_i^+ are a lower and, respectively, an upper bound for the expression

$$\sum_{j \in U'} v_j \left(\sum_{k \in U'} w_{ij} w_{ik} w_{jk} (1 - v_k) + \sum_{k \in U} w_{ij} w_{ik} w_{jk} \right).$$

Y_i^- and Y_i^+ could be

$$\begin{aligned}
Y_i^- &= \left(\sum_{j,k \in U': w_{ij} w_{ik} w_{jk} < 0} w_{ij} w_{ik} w_{jk} \right) / 2 + \sum_{j \in U', k \in U: w_{ij} w_{ik} w_{jk} < 0} w_{ij} w_{ik} w_{jk} \\
Y_i^+ &= \left(\sum_{j,k \in U': w_{ij} w_{ik} w_{jk} > 0} w_{ij} w_{ik} w_{jk} \right) / 2 + \sum_{j \in U', k \in U: w_{ij} w_{ik} w_{jk} > 0} w_{ij} w_{ik} w_{jk}
\end{aligned}$$

The problem (29) - (31) becomes

$$\max \left(\sum_{i \in U'} z_i + \sum_{i \in W'} \pi_i v_i \right) \tag{45}$$

$$\sum_{i \in C_j} v_i + \sum_{i \notin C_j} (1 - v_i) \geq 1, \forall j \in J, \tag{46}$$

$$\sum_{j \in U'} v_j \left(\sum_{k \in U'} w_{ij} w_{ik} w_{jk} (1 - v_k) + \sum_{k \in U} w_{ij} w_{ik} w_{jk} \right) + Y_i^+ v_i - z_i \leq Y_i^+, \forall i \in U', \tag{47}$$

$$\sum_{j \in U'} v_j \left(\sum_{k \in U'} w_{ij} w_{ik} w_{jk} (1 - v_k) + \sum_{k \in U} w_{ij} w_{ik} w_{jk} \right) + Y_i^- v_i - z_i \geq Y_i^-, \forall i \in U', \tag{48}$$

$$Y_i^- v_i - z_i \leq 0, \forall i \in U', \tag{49}$$

$$Y_i^+ v_i - z_i \geq 0, \forall i \in U', \tag{50}$$

$$v_i \in \{0, 1\}, \forall i \in U', \quad (51)$$

$$z_i \in \mathbb{R}, \forall i \in U'. \quad (52)$$

(For the coordination valuation function we can simply take $Y_i^- = 0$ and $Y_i^+ = |U'|^2/2 + |U'| \cdot |U| \leq 3n^2/4$, for all $i \in U'$.)

References

- [1] Bansal, N., Blum, A., Chawla, S., *Correlation Clustering*, Machine Learning, 56, 89–113, 2004.
- [2] Bistaffa, F., Farinelli, A., Cerquides, J., Rodriguez-Aguilar, J., Ramchurn, S. D., *Anytime coalition structure generation on synergy graphs*, Proceedings of AAMAS'14, 13–20, 2014.
- [3] Deng, X., Papadimitriou, C., *On the complexity of cooperative solution concepts*, Mathematics of Operations Research, 19 (2), 257–266, 1994.
- [4] Flammini, M., Monaco, G., Moscardelli, L., Shalom, M., Zaks, S., *Online Coalition Structure Generation in Graph Games*, Proceedings of AAMAS'18, 1353–1361, 2018.
- [5] Gallo, G., Hammer, P.L., Simeone, B., *Quadratic Knapsack Problems*, Mathematical Programming, 12 (2), 132–149, 1980.
- [6] Glover, F, Wolsey, E., *Improved linear integer programming formulations of nonlinear integer problems*, Amanagement Science, 22 (4), 180–182, 1975.
- [7] Hoffman, K. L., Padberg, M., *Solving Airline Crew Scheduling Problems by Branch-and-Cut*, Management Science, 6 (39), 657–682, 1993.
- [8] Pisinger, D., *The Quadratic Knapsack Problem - a Survey*, Discrete Applied Mathematics, 155, 623–648, 2007.
- [9] Rahwan, T., Ramchurn, S. D., Dang, V. D., Giovanucci, A., Jennings, N. R., *Anytime optimal coalition structure generation*, Proceedings of AAAI'07, 1184–1190, 2007.
- [10] Rahwan, T., Michalak, T. P., Wooldridge, M., Jennings, N. R., *Coalition structure generation: A survey*, Artificial Intelligence, 229, 139–174, 2015.

- [11] Sandholm, T., Larson, K., Andersson, M., Shehory, O., Tohme, F., *Coalition structure generation with worst case guarantees*, Artificial Intelligence, 1–2, 209–238, 2009.
- [12] Ueda, S., Iwasaki, A., Conitzer, V., Ohta, N., Sakurai, Y., Yakoo, M., *Coalition structure generation in cooperative games with compact representations*, Autonomous Agents and Multi-Agent Systems, 32, 503–533, 2018.
- [13] Voice, T., Polukarov, M., Jennings, N. R., *Coalition Structure Generation over Graphs*, Journal of Artificial Intelligence Research, 45, 165–196, 2012.
- [14] Voice, T., Ramchurn, S., Jennings, N. R., *On coalition formation with sparse synergies*, Proceedings of AAMAS’12, 45, 223–230, 2012.