A Branch and Price Algorithm for Coalition Structure Generation over Graphs

Emanuel Florentin Olariu[a], Cristian Frăsinaru[b] and Policiuc Abel Albert[c]
Department of Computer Science, "Alexandru Ioan Cuza" University, Bd. Carol, Iaşi, Romania

Keywords: Coalition Structure Generation, Linear/Quadratic Programming, Column Generation, Branch and Price.

Abstract: This paper presents an integer linear programming approach for the coalition structure generation (CSG) problem over graphs. Forming such structures is a major problem in areas like artificial intelligence an multi-agent systems. The problem asks to partition a given set of agents into coalitions in order to maximize their social well-fare - the agents being vertices in a given graph and their communication links being the edges. We give a truncated branch and price algorithm using valuation functions for which this problem is proven to be computationally hard. We consider three cases: first when the value of a coalition is the sum of the weights of its edges, second when the value takes account of both inter- and intra-coalitional disagreements and agreements, respectively, and another one when the value takes account of the pairs of adjacent agents which have common neighbors outside. The experimental results cover sets of up to fifty agents. Our approach prove that an off the shelf optimization solver can be used to solve CSG problem over graphs for some of the most used valuation functions. We prove also that for the coordination valuation the corresponding decision problem is NP-complete when the number of coalitions must be two.

1 INTRODUCTION

Coalition formation is one of the main approaches in multi-agent systems when cooperation among agents is required, especially if the agents have different objectives and collaboration skills. Coalition structure generation (CSG) is one of the steps in the coalition formation process (Sandholm et al., 2009) which includes also optimizing the coalitions performance and rewarding the coalitions value among the members.

Coalition structure generation is a major problem in artificial intelligence (Voice et al., 2012a), multi-agent systems (Bistaffa et al., 2014), (Voice et al., 2012a) communication networks, cooperative game theory (Deng and Papadimitriou, 1994), (Flammini et al., 2018), (Ueda et al., 2018), scheduling (Hoffman and Padberg, 1993), combinatorial auctions etc. Given a set of agents $V = \{1, 2, \ldots, n\}$, and a *valuation function* $v : 2^V \to \mathbb{R}$ 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 comes from real-world applications: consider a set of agents who can cooperate by working in groups. 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. Cooperative game theory usually uses a super-additive valuation function that values better a merged coalition than the sum of the values of the component coalitions; this leads to the great coalition formation which can be costly to coordinate or manipulate. Besides this 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; hence implicit representations of the valuation function are not practical. The literature presents a very large number of approaches to CSG problem like (Rahwan et al., 2015): dynamic programming, meta heuristic methods, branch and bound algorithms based on dividing the searching space, anytime algorithms that maintain a monotoni-

[a] https://orcid.org/0000-0001-5218-0312
[b] https://orcid.org/0000-0002-5246-7396
[c] https://orcid.org/0000-0002-7111-0026

Olariu, E., Frăsinaru, C. and Albert, P.
A Branch and Price Algorithm for Coalition Structure Generation over Graphs.
DOI: 10.5220/0010316303900397
In *Proceedings of the 13th International Conference on Agents and Artificial Intelligence (ICAART 2021) - Volume 1*, pages 390-397
ISBN: 978-989-758-484-8
Copyright © 2021 by SCITEPRESS – Science and Technology Publications, Lda. All rights reserved

cally improving feasible solution etc.

The classic CSG framework assumes no particular structure of the set of agents, while in many real world applications the agents are the nodes of a network. This is the background of the following version of the CSG problem: to generate a coalition structure when the agents are placed in the vertices of a given graph and the edges represent the communication links between them; each such link having a capacity that indicates the probability that the link works, the amount of information that can be changed along it, or the level of trust between the two agents. In such a context the value of a group of nodes must depend on the capacities of the edges between them and the group members must share a certain level of connectedness in order to ensure the spread of the information.

In this paper we provide a truncated branch and price algorithm which proved to be effective for some of the most used coalition valuations to form coalitions among agents embedded in a graph. Our algorithm is independent of the structure of the underlying graph and works for up to 50 agents. Our setting suggests that we can deal with CSG over graphs as long as the value of a coalition can be expressed as a polynomial of the characteristic vector of the coalition.

The remaining of the paper is organized as follows: Section 2 describes our setting and the related work, Section 3 describes the LP model and the column generation method, Section 4 introduces the valuation functions, the corresponding sub-problems and their equivalent formulations, Section 5 describes the branch and price algorithm, and the last Sections contain the numerical results and the conclusions.

2 BACKGROUND

Various approaches were employed for this problem ranging from constraint optimization to integer linear programming (ILP). In (Voice et al., 2012a) a dynamic programming (DP) algorithm is proposed for valuations having the independent disconnected members property, other DP approaches can be found in (Rahwan et al., 2009a) and (Michalak et al., 2016).

Mixed integer linear programming (MILP) technique is used in (Ueda et al., 2018) and (Tran–Thanh et al., 2013). In (Tran–Thanh et al., 2013) a coalitional skill vector model is introduced and the value of a coalition is measured as a function of the distance between the corresponding coalition's skill vector and a set of goals, while (Ueda et al., 2018) introduced a mixed integer programming formulation based on a set of rules. Other approaches are based on concise representations of the searching space and include branch and bound algorithms (Bistaffa et al., 2014), (Rahwan et al., 2009b), and constraint optimization methods (Ueda et al., 2018). Sometimes in order to reduce the execution time of the algorithm a truncating techniques are employed like in the case of anytime algorithms that can offer at any moment a suboptimal solution to the problem (Bistaffa et al., 2014), (Rahwan et al., 2009b), (Sandholm et al., 2009).

The CSG problem over graphs has been also studied for graphs with a particular structure, e. g. synergistic graphs (Voice et al., 2012b) and (Bistaffa et al., 2014), graphs with bounded tree-width in (Voice et al., 2012a)). To a limited extent the only related algorithms to ours are the MIlP formulations (Ueda et al., 2018) which require that the valuation function to have a rules-based representation or (Tran–Thanh et al., 2013) which uses a vector-based representation that assigns a skill vector to each agent, and a coalition's skill level will be expressed as an aggregation of the composing skill vectors. Hence we cannot compare our approach against these two approaches.

Our setting is that of a (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, therefore we require that the coalitions induce connected sub-graphs.

We study three types of valuations: the *edge sum*, the *correlation*, and the *coordination* functions. The *edge sum* valuation of a coalition (Deng and Papadimitriou, 1994) is the sum of the weights of its edges, this function frequently occurs in communication networks and cooperative game theory. The *correlation* valuation function (Bansal et al., 2004) occurs in the clustering framework and it 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. The *coordination* valuation accounts for all 3-cliques of three agents, two inside the coalition and one outside; this valuation aims to privileges coalitions having agents with common outside neighbors.

Our approach is based on solving an ILP 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 subproblem can be solved. We proved that this can be done for edge sum, correlation, and coordination valuation functions since the sub-problem becomes a quadratic knapsack problem or a mixed integer quadratically constrained programming problem with forbidden configurations. These problems can be solved in practice using the corresponding mixed integer quadratic programming model or the MILP 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, up to 50 agents for the correlation valuation function, and up to 35 agents for the coordination valuation function using just a regular PC.

3 LP MODEL AND COLUMN GENERATION

Consider $V = \{1,2,\ldots,n\}$ to be a set of n agents and $v : 2^V \to \mathbb{R}$ a valuation function on the power set of V. A *coalition structure* is a collection of pairwise disjoint exhaustive subsets (coalitions) $V_1, V_2, \ldots, V_p$, i.e. $V_i \cap V_j = \varnothing$ for all $1 \le i < j \le p$ and $\bigcup_{i=1}^{p} V_i = V$. The problem of finding a coalition structure of maximum value is equivalent with the set partitioning problem (SPP) (Hoffman and Padberg, 1993):

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

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

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

where $\{C_1, \ldots, C_{2^n-1}\}$ is an enumeration of $2^V \setminus \{\varnothing\}$ and $(a_{ij})_{i \in V}$ is the characteristic vector of C_j, that is

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

We relax (3) by replacing them with

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

We write the relaxed problem as a minimum one by replacing (1) by

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

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

$$max \sum_{j=1}^{n} \pi_i \quad (4)$$

$$\sum_{i=1}^{n} a_{ij}\pi_i \le -v(C_j), \forall j = \overline{1, 2^n - 1} \quad (5)$$

$$\pi_i \in \mathbb{R}, \forall i \in V \quad (6)$$

We will use the column generation method (Lubbecke and Desrosiers, 2005) for dealing with the huge number of variables in the primal problem. Let $C_j = \{j\}$, for any $j \in \{1,2,\ldots,n\}$. An initial feasible basic solution to the problem (1'), (2), (3') could be that corresponding to the coalition structure $(\{1\},\{2\},\ldots\{n\})$, i. e., $x_1, x_2, \ldots, x_n$. We restrict the problem to a smaller number of variables including $x_1, x_2, \ldots, x_m$ and we get the restricted master problem (RMP):

$$min \sum_{j \in J} -v(C_j)x_j \quad (7)$$

$$\sum_{j \in J} a_{ij}x_j = 1, \forall i \in V \quad (8)$$

$$x_j \ge 0, \forall j \in J \quad (9)$$

where $J = \{1,2,\ldots,m\} \cup J' \subseteq \{1,2,\ldots,2^m - 1\}$.
The dual of the RMP is

$$max \sum_{i=1}^{n} \pi_i \quad (10)$$

$$\sum_{i=1}^{n} a_{ij}\pi_i \le -v(C_j), \forall j \in J \quad (11)$$

$$\pi_i \in \mathbb{R}, \forall 1 \le i \le n \quad (12)$$

Let $(\mathbf{x}, \pi)$ 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 does not exist we can stop: we have an optimum solution to the primal problem. The sub-problem is to find

$$j_0 = \underset{j \in \{0,1,\ldots,2^n-1\} \setminus J}{\arg\max} \left(v(C_j) + \sum_{i=1}^{n} a_{ij}\pi_i \right) \quad (13)$$

The arising question is: how can we solve problem (13)? 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*, and for another one the *coordination* valuation.

4 COALITION STRUCTURE GENERATION OVER GRAPHS

4.1 The Edge-Sum Valuation

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

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

This function was extensively studied in the context of cooperative game theory (Deng and Papadimitriou, 1994) and the corresponding CSG problem is NP-complete even for planar graphs (Voice et al., 2012a).

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

$$\max_{\mathbf{v}\in\{0,1\}^m, v\neq\chi_{C_j},\forall j\in J} \left(\sum_{ij\in E} w_{ij}v_iv_j + \sum_{i\in V}\pi_i v_i\right) > 0 \quad (14)$$

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:

$$max \left(\sum_{ij\in E} w_{ij}v_iv_j + \sum_{i\in V}\pi_i v_i\right) \quad (15)$$

$$\sum_{i\notin C_j} v_i + \sum_{i\in C_j}(1-v_i) \geq 1, \forall j\in J \quad (16)$$

$$v_i \in \{0,1\}, \forall i\in V \quad (17)$$

We cast the constraints (16) in order to avoid the already added coalitions/variables to the original problem and the already revealed unconnected coalitions (J is the set of indexes of these variables).

The problem (15) - (17) has a quadratic objective but only linear constraints: it is 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 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 non-negative integer or positive coefficients for the quadratic (some times non-diagonal) terms (Caprara et al., 1999), (Gallo et al., 1980), (Pisinger, 2007). Hence we had to settle to solve it as a general IQP problem using a mathematical optimization solver.

4.2 The Correlation Valuation

Suppose we have a function $e: E\to\{+,-\}$ that assigns to each edge $ij\in E$ a label $+$ 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 (Bansal et al., 2004): 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

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

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

The *correlation valuation function* is

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

It was known that the corresponding CSG problem is *NP*-complete (Bansal et al., 2004). We define two weight functions on the set of edges of G, $w^+, w^-: E\to\{0,1\}$, 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

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

The sub-problem (13) becomes the following QKPf problem:

$$\begin{aligned} max \quad & \left(\frac{1}{2}\sum_{i\in V}\sum_{j\in V} w^+_{ij}v_iv_j + \right. \\ & \left. +\frac{1}{2}\sum_{i\in V}\sum_{j\in V} w^-_{ij}v_i(1-v_j) + \sum_{i\in V}\pi_i v_i\right) \\ \sum_{i\notin C_j} v_i + \sum_{i\in C_j}(1-v_i) & \geq 1, \forall j\in J \\ v_i & \in \{0,1\}, \forall i\in V \end{aligned}$$

4.3 The Coordination Valuation

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

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

The *coordination coalition valuation function* (Voice et al., 2012a) is

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

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 neighbors from outside. We will generalize this definition by including the weight of the 3-clique as the product of the weights on its edges. Hence, we consider first $w : E \to \mathbb{R}_+$ to be a weight on the edges of G. Second, if $\mathbf{v} = (v_1, v_2, \ldots, v_n)$ is the characteristic vector for the generic coalition C, then its coordination value is

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

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

We note here that the generalization may have the following interpretation: consider a graph that have probabilities on its edges representing the reliability of the corresponding communication links - supposing that the edges works independently. By multiplying these probabilities for the edges of a 3-clique one gets the probability that the members of the group can communicate to each other.

The sub-problem (13) becomes:

$$\begin{aligned} & max \quad \left(\sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} w_{ij} w_{ik} w_{jk} v_i v_j (1 - v_k) + \sum_{i \in V} \pi_i v_i \right) \\ & \sum_{i \notin C_j} v_i + \sum_{i \in C_j} (1 - v_i) \quad \geq \quad 1, \forall j \in J \\ & v_i \quad \in \quad \{0,1\}, \forall i \in V \end{aligned}$$

Using the method of Glover and Wolsey (1974, 1975), we will transfom the cubic objective function into a linear one; by introducing the variables

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

the above problem becomes

$$\begin{aligned} & max \quad \left(\sum_{i=1}^{n} z_i + \sum_{i=1}^{n} \pi_i v_i \right) \\ & \sum_{i \notin C_j} v_i + \sum_{i \in C_j} (1 - v_i) \quad \geq \quad 1, \forall j \in J, \\ & \sum_{j=1}^{n} v_j \left[\sum_{k=1}^{n} w_{ij} w_{ik} w_{jk} (1 - v_k) \right] + \beta_i v_i - z_i \leq \beta_i, \forall i \in V \\ & \sum_{j=1}^{n} v_j \left[\sum_{k=1}^{n} w_{ij} w_{ik} w_{jk} (1 - v_k) \right] - z_i \geq 0, \forall i \in V \\ & z_i \geq 0, \forall i \in V \\ & \beta_i v_i - z_i \quad \geq \quad 0, \forall i \in V \\ & v_i \quad \in \quad \{0,1\}, \forall i \in V \\ & z_i \quad \in \quad \mathbb{R}, \forall i \in V \end{aligned}$$

where $\beta_i = \sum_{j=1}^{n} \left(\sum_{k: w_{ij} w_{ik} w_{jk} > 0} w_{ij} w_{ik} w_{jk} \right)$.

5 BRANCH AND PRICE ALGORITHM

In this section we give the details of our branch and price algorithm. 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 $C_h \cap U \neq \varnothing$ (or, equivalently, $C_h \cap C_j \neq \varnothing$, 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 chooses a variable, x_j, from the optimal solution in the current node of the branching tree having a value around 0.5 (e. g. $x_j \in (0.35, 0.65)$ - if possible). Our branch and bound does not need upper bounds because in almost all cases a feasible solution occurs very early (for the edge-sum and the correlation functions) - 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 for the first two valuations. 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 three valuation functions the sub-problems become:

$$
\begin{aligned}
max \quad & \left(\sum_{i \in U'} \sum_{j \in U'} w_{ij} v_i v_j + \sum_{i \in U'} \pi_i v_i \right) \\
\sum_{i \notin C_j} v_i + \sum_{i \in C_j} (1 - v_i) \quad & \geq \quad 1, \text{ if } C_j \subseteq U' \\
v_i \quad & \in \quad \{0,1\}, \forall i \in U',
\end{aligned}
$$

$$
\begin{aligned}
max \quad & \left[\frac{1}{2} \sum_{i \in U'} \sum_{j \in U'} (w_{ij}^{+} - w_{ij}^{-}) v_i v_j + \right. \\
& \left. + \sum_{i \in U'} \left(\frac{1}{2} \sum_{j \in V} w_{ij}^{-} + \pi_i \right) v_i \right] \\
\sum_{i \notin C_j} v_i + \sum_{i \in C_j} (1 - v_i) \quad & \geq \quad 1, \text{ if } C_j \subseteq U', \\
v_i \quad & \in \quad \{0,1\}, \forall i \in U',
\end{aligned}
$$

and

$$
\begin{aligned}
max \quad & \left(\sum_{i \in U'} z_i + \sum_{i \in U'} \pi_i v_i \right) \\
\sum_{i \notin C_j} v_i + \sum_{i \in C_j} (1 - v_i) \quad & \geq \quad 1, \text{ if } C_j \subseteq U', \\
\sum_{j \in U'} \left(\sum_{k=1}^{n} w_{ij} w_{ik} w_{jk} \right) v_j - & \sum_{j \in U'} \sum_{k \in U'} w_{ij} w_{ik} w_{jk} v_j v_k + \\
+ \beta_i' v_i - z_i \leq \beta_i', & \forall i \in U' \\
\sum_{j \in U'} \left(\sum_{k=1}^{n} w_{ij} w_{ik} w_{jk} \right) v_j - & \sum_{j \in U'} \sum_{k \in U'} w_{ij} w_{ik} w_{jk} v_j v_k - \\
& -z_i \geq 0, \forall i \in U' \\
& z_i \geq 0, \forall i \in U' \\
& \beta_i' v_i - z_i \quad \geq \quad 0, \forall i \in U' \\
& v_i \quad \in \quad \{0,1\}, \forall i \in U' \\
& z_i \quad \in \quad \mathbb{R}, \forall i \in U'.
\end{aligned}
$$

6 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 (but the solver quickly finds optimal solutions to these QKPf problems), while for the other nodes the number of 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, correlation or coordination valuation functions, the algorithm has been tested on randomly generated problems. Our test problems were built using the Gilbert model (each edge has a fixed probability of being present in the graph, independently of any other edges - probability 1 gives a complete graph). The weights are independently generated using a Gaussian distribution $N(0,0.2)$ (a certain similar valuation function occurs in (Rahwan et al., 2009a). The parameters of the benchmark file indicate the probability that an edge belongs to the graph ("p"), the number of agents ("n"), and the instance number ("s").

Table 1: Solutions and computational times for different instances (Edge-sum valuation).

Instance	s0	s1	s2	s3	s4
Benchmark	$p=0.8, n=40$				
LP solution	17.87	21.49	20.10	19.32	18.16
ILP solution	17.86	21.35	20.10	19.29	18.13
Overall time (s)	744	1,684	673	756	850
Root time (s)	529	924	673	615	557
Per node (s)	248	122	673	151	179
# of nodes	3	15	1	5	5
# of integer sol.	1	1	1	1	1
# of variables	318	856	292	320	319
Gap	0.05%	0.65%	0.00%	0.19%	0.16%
Benchmark	$p=0.8, n=45$				
LP solution	22.60	22.58	21.48	23.88	21.50
ILP solution	22.41	22.49	21.46	23.88	21.50
Overall time (s)	6,372	4,381	2,898	2,726	4,345
Root time (s)	3,488	2,875	1,779	2,726	4,345
Per node (s)	159	398	193	2,726	4,345
# of nodes	40	11	15	1	1
# of integer sol.	5	7	2	1	1
# of variables	1,600	2,222	801	1,114	1,399
Gap	0.84%	0.39%	0.09%	0.00%	0.00%
Benchmark	$p=1.0, n=35$				
LP solution	17.15	16.42	19.40	15.50	19.06
ILP solution	17.15	16.38	19.25	15.38	19.01
Overall time (s)	323	425	1,603	594	937
Root time (s)	323	325	561	375	397
Per node (s)	323	85	45	84	104
# of nodes	1	5	35	7	9
# of integer sol.	1	1	2	2	1
# of variables	204	274	1,232	335	516
Gap	0.00%	0.24%	0.77%	0.77%	0.26%
Benchmark	$p=1.0, n=40$				
LP solution	25.28	23.99	22.40	21.94	18.34
ILP solution	25.28	23.78	22.40	21.94	18.25
Overall time (s)	3,874	2,988	2,444	3,264	3,395
Root time (s)	3,874	143	2,444	3,264	1,620
Per node (s)	3,874	332	2,444	3,264	147
# of nodes	1	9	1	1	23
# of integer sol.	1	3	1	1	4
# of variables	299	536	264	276	870
Gap	0.00%	0.88%	0.00%	0.00%	0.49%

Table 2: Solutions and computational times for different instances (Correlation valuation).

Instance	s0	s1	s2	s3	s4
Benchmark	$p = 0.4, pSign = 0.6, n = 50$				
LP solution	321.0	339.4	324.0	332.0	336.2
ILP solution	321	339	324	332	336
Overall time (s)	2,893	7,108	4,197	4,458	6,487
Root time (s)	2,893	2,947	4,197	4,458	3,284
Per node (s)	2,893	473.8	4,197	4,458	1,297
# of nodes	1	15	1	1	5
# of integer sol.	1	1	1	1	2
# of variables	1,072	3,071	1,168	1,139	4,399
# of disc. coal.	239	554	6	131	3,207
Gap	0.00%	0.11%	0.00%	0.00%	0.20%
Benchmark	$p = 0.6, pSign = 0.6, n = 40$				
LP solution	291.0	298.0	289.0	311.0	313.5
ILP solution	291	298	289	311	313
Overall time (s)	880	1,558	1,379	1,039	3,289
Root time (s)	880	1,558	1,379	1,039	2,352
Per node (s)	293	1,558	1,379	1,039	657
# of nodes	3	1	1	1	5
# of integer sol.	1	1	1	1	1
# of variables	843	1,031	527	531	1,241
# of disc. coal.	78	64	7	9	127
Gap	0.00%	0.00%	0.00%	0.00%	0.16%
Benchmark	$p = 0.6, pSign = 0.6, n = 45$				
LP solution	369.0	374.0	383.0	380.0	378.0
ILP solution	369	374	383	380	378
Overall time (s)	6,646	8,009	6,140	5,241	6,176
Root time (s)	5,105	8,009	6,140	5,241	6,176
Per node (s)	2,215	8,009	6,140	5,241	6,176
# of nodes	3	1	1	1	1
# of integer sol.	1	1	1	1	1
# of disc. coal.	19	13	3	20	3
# of variables	1,380	1,230	846	1,264	657
Gap	0.00%	0.00%	0.00%	0.00%	0.00%
Benchmark	$p = 0.8, pSign = 0.6, n = 35$				
LP solution	304.0	294.5	299.0	307.0	298.0
ILP solution	304	294	299	307	298
Overall time (s)	988	1,254	1,557	1,077	952
Root time (s)	988	808	1,557	1,077	952
Per node (s)	988	139	1,557	1,077	952
# of nodes	1	9	1	1	1
# of integer sol.	1	1	1	1	1
# of variables	634	1,072	705	583	389
# of disc. coal.	4	10	0	1	1
Gap	0.00%	0.16%	0.00%	0.00%	0.00%
Benchmark	$p = 0.8, pSign = 0.6, n = 40$				
LP solution	395.0	380.5	385.0	394.0	387.0
ILP solution	395	380	385	394	387
Overall time (s)	4,227	6,533	6,523	9,721	7,775
Root time (s)	4,227	3,782	6,523	9,721	7,775
Per node (s)	4,227	1,306	6,523	9,721	7,775
# of nodes	1	5	1	1	1
# of integer sol.	1	1	1	1	1
# of variables	840	1,728	851	886	735
# of disc. coal.	0	25	2	5	0
Gap	0.00%	0.13%	0.00%	0.00%	0.00%

We slightly generalized the correlation function by supposing that some edges don't have signs at all, i. e., 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 parameters of a benchmark file for this valuation indicate the probability that an edge belongs to the graph ("p"), the probability of the plus sign ("pSign"), the number of agents ("n"), and the instance number ("s").

Table 3: Solutions and computational times for different instances (Coordination valuation).

Instance	s0	s1	s2	s3	s4
Benchmark	$p = 0.3, n = 35$				
LP solution	45.32	57.07	56.61	97.68	44.13
ILP solution	35.44	44.88	46.04	76.35	40.43
Total time (s)	1,852	3,234	8,458	6,450	5,119
Root time (s)	826	1,904	3,065	2,734	866
Per node (s)	45.2	78.89	206.31	157.4	124.9
# of nodes	41	41	41	41	41
# of int. sol.	4	3	2	3	2
# of vars.	4,165	2,275	15,566	3,298	2,152
# disc. coal.	3,462	1,710	14,995	2,555	254
Benchmark	$p = 0.4, n = 30$				
LP solution	69.08	90.62	79.88	51.72	88.14
ILP solution	54.92	90.34	79.52	51.43	87.92
Total time (s)	1,376	5,925	1,779	2,274	4,419
Root time (s)	727	3,014	1,143	1,202	2,312
Per node (s)	33.6	144.5	53.7	55.5	107.8
# of nodes	41	41	41	41	41
# of int. sol.	3	4	4	2	3
# of vars.	1,294	1,117	1,133	1,237	1,095
# disc. coal.	729	548	650	675	585

The parameters of a benchmark file for coordination valuation indicate the probability that an edge belongs to the graph ("p"), the number of agents ("n"), and finally the instance number ("s"). The probabilities (weights) on the edges are randomly generated as standard uniform variates.

Tables 1, 2, and 3 show the results of our numerical tests; they contain the optimum value in the root (LP problem), the (most of the time) optimum value of the ILP problem found by branch and price algorithm, the overall running time, the running time for finding and solving the root LP problem, the running time per node, the number of nodes of the tree, the total number of variables added during the execution, the number of nodes fathomed by integrality (where the best integral solution improves the objective function), the number of disconnected coalitions (which must be bypassed), and the gap between the LP and the ILP optimum values where is the case. For the edge-sum valuation the instances generated with $p = 0.8$ or 1.0 the algorithm found no variables corresponding to disconnected coalitions. The running time increases as the edge probability (hence the density) of the graph increases, but the number of disconnected coalitions decreases.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

7 CONCLUSIONS

We addressed the problem of coalition structure generation over graphs for three valuations functions where the set of feasible coalitions is restrained to those that induce connected sub-graphs. Our goal was to test the efficiency of the column generation method and the branch and bound algorithm (i. e. the branch and price algorithm) which needed to be truncated at least for one of the valuations (the coordination).

Our algorithm is one of the first to present competitive results for more than 40 agents for some of the most frequently used 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. For the first two valuations 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. For the edge-sum and the correlation valuations 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. for the coordination valuation the branching rule must privilege the variables set to 1 disregarding the size of the corresponding coalition. One of the conclusions of the numerical experiments is that the results depends on the magnitude of the weights. For the low density instances the number of disconnected coalitions could be high since the rejection procedure for such coalitions could not included in the mathematical programming model.

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 (inequalities defining facets of a subjacent polyhedral relaxation) that tightens the LP relaxations towards a branch and cut algorithm.

ACKNOWLEDGEMENTS

The first author contribution in this this project is funded by the Ministry of Research and Innovation within Program 1 – Development of the national RD system, Subprogram 1.2 – Institutional Performance – RDI excellence funding projects, Contract no.34PFE/19.10.2018.

REFERENCES

Bansal, N., Blum, A., and Chawla, S. (2004). Correlation clustering. *Machine Learning*, (56):89–113.

Bistaffa, F., Farinelli, A., Cerquides, J., Rodriguez-Aguilar, J., and Ramchurn, S. D. (2014). Anytime coalition structure generation on synergy graphs. In *Proceedings of AAMAS'14*, pages 13–20.

Caprara, A., Pisinger, D., and Toth, P. (1999). Exact solution of the quadratic knapsack problem. *INFORMS Journal on Computing*, 11(2):125–137.

Deng, X. and Papadimitriou, C. (1994). On the complexity of cooperative solution concepts. *Mathematics of Operations Research*, 19(2):257–266.

Flammini, M., Monaco, G., Moscardelli, L., Shalom, M., and Zaks, S. (2018). Online coalition structure generation in graph games. In *Proceedings of AAMAS'18*, pages 1353–1361.

Gallo, G., Hammer, P., and Simeone, B. (1980). Quadratic knapsack problems. *Math. Progr. St.*, 12(2):132–149.

Hoffman, K. L. and Padberg, M. (1993). Solving airline crew scheduling problems by branch-and-cut. *Management Science*, 6(39):657–682.

Lubbecke, M. and Desrosiers, J. (2005). Selected topics in column generation. *Operations Research*, 53(6):1007–1023.

Michalak, P. T., Rahwan, T., Elkind, E., Wooldridge, M., and Jennings, N. R. (2016). A hybrid exact algorithm for complete set partitioning. *Artificial Intelligence*, (230):14–50.

Pisinger, D. (2007). The quadratic knapsack problem - a survey. *Discr. App. Mathematics*, 155(5):623–648.

Rahwan, T., Michalak, P. T., Elkind, E., Wooldridge, M., and Jennings, N. R. (2009a). An exact algorithm for coalition structure generation and complete set partitioning. *Res. Rep., Oxford*, (CS-RR-13-09).

Rahwan, T., Michalak, P. T., Wooldridge, M., and Jennings, N. R. (2015). Coalition structure generation: A survey. *Artificial Intelligence*, (229):139–174.

Rahwan, T., Ramchurn, S., Jennings, N. R., and Giovannucci, A. (2009b). An anytime algorithm for coalition structure generation. *J. of Art. Int. Res.*, (34):521–567.

Sandholm, T., Larson, K., Andersson, M., Shehory, O., and Tohme, F. (2009). Coalition structure generation with worst case guarantees. *Artificial Intelligence*, 1(2):209–238.

Tran–Thanh, L., Nguyen, T. D., Rahwan, T., Rogers, A., and Jennings, N. R. (2013). An efficient vector–based representation for coalitional games. In *Proceedings of the 23rd IJCAI*, page 383–389.

Ueda, S., Iwasaki, A., N., V. C., Sakurai, O. Y., and Yakoo, M. (2018). Coalition structure generation in cooperative games with compact representations. *Proceedings of AAMAS'18*, (32):503–533.

Voice, T., Polukarov, M., and Jennings, N. R. (2012a). Coalition structure generation over graphs. *J. of Art. Intel. Res.*, (45):165–196.

Voice, T., Ramchurn, S., and Jennings, N. R. (2012b). On coalition formation with sparse synergies. In *Proceedings of AAMAS'12*, pages 223–230.