A DISTRIBUTED ALGORITHM FOR COALITION FORMATION IN LINEAR PRODUCTION DOMAIN

Chattrakul Sombattheera
Decision Systems Lab
School of Information Technology and Computer Science University of Wollongong, NSW, Australia

Aditya Ghose
Decision Systems Lab
School of Information Technology and Computer Science University of Wollongong, NSW, Australia

Keywords: Linear production, coalition formation, coalition structure

Abstract: Coalition formation is an important area of research in multi-agent systems. The large number of agents can make coalition formation become a complex process. The problem of dealing with large number of agents during coalition formation has received little attention in the literature. Previous studies assume that each coalition value is known a priori. This assumption is impractical in realworld settings. Furthermore, the problem of finding coalition values become intractable for even a relatively small number of agents. This work studies coalition formation among fully cooperative agents in linear production domain, where each coalition value is not known a priori. The common goal of the agents is to maximize the system's profit. We propose a distributed algorithm that allow agents to i) deliberate profitable coalitions, and ii) compute optimal coalition structure. We show that our algorithm outperforms exhaustive search when generating profitable coalitions, which can be used to achieve optimal coalition structure that yields the system's maximal profit.

1 INTRODUCTION

Coalition formation is an important area of research in multi-agent systems. It studies the process that leads to cooperation among agents. The process involves i) negotiation in order to exchange information among agents, and ii) deliberation in order to decide with which agents should they cooperate. Coalition formation research has its roots in cooperative game theory where the focus is on what coalitions would form and what the payoffs for agents are. On the other hand, researchers in multi-agent systems are concerned with the complexity of coalition formation and in particular, settings where the number of agents involved is very large. The large number of agents can make coalition formation become a complex process, i.e., there is a large number of messages sent across while negotiating and there is a large number of possible coalitions to be considered while deliberating.

The problem of dealing with large number of agents during coalition formation has received little attention in the literature. A small of number of studies have considered the complexity of deliberation (Sandholm et al., 1999). These studies, as in game theory, assume that a coalition value is associated with each coalition. Then they focus on how the optimal coalition structure can be achieved in a timely fashion. The complexity of finding coalition structure using thourough search is exponetial time (Sandholm et al., 1999).

On the other hand, this work considers coalition formation where coalition values are not known a priori. This kind of setting is common in real world environment. Examples include internal/external collaboration in third party logistics providers, cooperation among nodes in grid computing and cooperation among service providers in composite web service environment. These real world scenarios make coalition formation highly complex because agents have to i)compute coalition values, and ii) compute the optimal coalition structure. Given m agents in a coalition formation process, the number of possible coalitions is 2^m, which is also the number of coalition values to be computed. The process of computing coalition values is complex, as is the process of deliberation. The problem becomes intractable even for a relatively small values of m.

Our goal in this research is to deal with this complexity. We modify Owen's linear production game where agents have to agree to pool their resources together in order to produce goods. The original work assumes a superadditive environment, where agents

Sombattheera C. and Ghose A. (2006).
A DISTRIBUTED ALGORITHM FOR COALITION FORMATION IN LINEAR PRODUCTION DOMAIN.
In *Proceedings of the Eighth International Conference on Enterprise Information Systems - AIDSS*, pages 17-22
DOI: 10.5220/0002445000170022

can simply form the grand coalition. Such an assumption is impractical in the real world. Since the cost of cooperation has to be taken into account while negotiating to form coalitions.

The outline of this paper is as follows. We introduce our setting. We describe how our algorithm works and discuss both in deliberating and forming coalitions. Then we discuss about the experiment and show empirical results. We discuss related work which followed by conclusion and future work.

2 COALITION FRAMEWORK

2.1 Linear Production Domain

Linear production games (Owen, 1975) are those in which agents are given resources and try to pool resources to produce goods in order to maximize the system's profit. Owen (Owen, 1975) studied linear production games in superadditive environment. Here, we consider linear production games in non-superadditive environments. We are given a set of agents, $A = \{a_1, a_2, \ldots, a_m\}$, whose goals are to maximize the system's profit. We are also given a set of resources $R = \{r_1, r_2 \ldots, r_n\}$ and a set of goods $G = \{g_1, g_2, \ldots, g_o\}$. Resources themselves are not valuable but they can be used to produce goods, which are valuable to agents. Let $L = [\alpha_{ij}]_{n\times o}$, where $\alpha_{ij} \in \mathbb{Z}^+$, be the matrix that specifies the units of each resource $r_i \in R$ required to produce a unit of the good $g_j \in G$. Such a matrix is called a *linear technology matrix* (Owen, 1975). The price of each unit of goods produced is specified by the vector $P = [p_j]_{1\times o}$. Each agent $a_k \in A$ is given a resource bundle $b^k = [b^k_i]_{n\times 1}$. In this setting, some agents would have the incentive to cooperate, e.g., if they cannot produce a certain good using only the resources at their disposal. Hence agents have to cooperate, i.e. form coalitions, in order to create value from their resources. Let $S \subseteq A$ be a coalition. It will have a total of

$$b^S_i = \sum_{k\in S} b^k_i$$

of the i^{th} resource. The members of coalition S can use all these resources to produce any vector $x = \langle x_1, x_2, \ldots, x_o\rangle$ of goods that satisfies the following constraints:

$$\begin{array}{rcl} \alpha_{11}x_1 + \alpha_{12}x_2 + \ldots + \alpha_{1o}x_o & \leq & b^S_1, \\ \alpha_{21}x_1 + \alpha_{22}x_2 + \ldots + \alpha_{2o}x_o & \leq & b^S_2, \\ \vdots & \vdots & \vdots, \\ \alpha_{n1}x_1 + \alpha_{n2}x_2 + \ldots + \alpha_{no}x_o & \leq & b^S_n \end{array}$$

and

$$x_1, x_2, \ldots, x_o \geq 0.$$

We assume that agents have to pool their resources together at a coalition member's location to produce these goods. Thus agents' cooperation incurs some costs, e.g., transportation cost, etc. The cooperation cost among agents is specified by the matrix $C = [c_{kl}]_{m\times m}$, which assigns a cooperation cost between each pair (a_k, a_l) of agents such that

$$c_{kl} \in \begin{cases} \mathbb{Z}^+ & \text{if } k \neq l \\ \{0\} & \text{if } k = l \end{cases}$$

We assume that all of the resources of agents are pooled at one location, which can be the location of any agent in the coalition. A singleton coalition yields cooperation cost of 0. For a coalition of size two, $S = \{a_1, a_2\}$, pooling coalition resources at any of the two sites yield the same cost for the coalition (i.e. the cooperation cost matrix is symmetric). The total cost for cooperation incurred by a coalition will be taken to be the sum of the pairwise cooperation costs between the agent at whose location coalition resources are pooled, and the other members of coalition. For a coalition of size three or larger, there is at least one agent, a_k, such that

$$\sum_{k'=1}^{m} c_{kk'} \leq \sum_{l'=1}^{m} c_{ll'}$$

for all $a_l \in S$. We shall call a coalition member a_k who yields the minimal cooperation cost for the coalition a *coalition center*.

Agents in the coalition S have to find a vector x to maximize the revenue accruing to a coalition. Let $P_S = \sum_{l=1}^{o} p_l x_l$ be the maximal revenue the coalition can generate. Let $C_S = \sum_{l\in S} c_{kl}$ be the minimal cooperation cost for the coalition (obtained by selecting the optimal coalition center). Obviously, the ultimate objective of agents in the coalition is to maximize profit, i.e., the coalition value υ_S, where $\upsilon_S = P_S - C_S$.

The linear inequalities referred to above, together with this objective function constitutes a linear programming problem. We shall call the solution, the vector $\langle x_1, x_2, \ldots, x_o\rangle$ that represents the optimal quantities of goods $g_1, g_2, \ldots, g_o$ *optimal product mix*.

2.2 Optimal Coalition Structure

Coalition formation problems can also be considered as a set partitioning problem. The set of all agents will be partitioned into mutually disjoint and proper subsets. Each instance of a partition is known as a *coalition structure*(CS) (Dang and Jennings, 2004; Sandholm et al., 1999; Kahan and Rapoport, 1984), while each subset is known as a coalition S. The *value* of each coalition structure

$$V(CS) = \sum_{S\in CS} \upsilon_S$$

indicates the system' utility yielded by that partitioning. The goal of cooperative agents in coalition formation (Sandholm et al., 1999; Dang and Jennings, 2004) is to maximize the system's utility.

Computing the optimal coalition structure in a non-superadditive environment is non-trivial (Sandholm et al., 1999). Previous studies (Sandholm et al., 1999; Dang and Jennings, 2004) assumed the existence of a characteristic function and considered algorithms for computing the optiamal coalition structure. Such an assumption is impractical in the real world—each coalition value may not be known a priori. Thus agents have to compute all coalition values first. Given a set of m agents, there are 2^m possible subsets, hence the complexity of computing all coalition structures is substantially worse.

Here, we consider a distributed algorithm that allows agents to compute coalition values and approach the optimal coalition structure as they proceed. Each agent has to do to two tasks: i) Delibarating: deliberate over what coalitions it might form by incrementally improving the initial set of coalitions, and ii) Forming coalitions: exchange information to form coalitions such that those coalitions yield maximal profit to the system. The sets of such coalitions are the optimal coalition structures. The main goal of the algorithm is to reduce search space for finding the optimal coalition structure. This can be achieved by reducing the number of coalitions to be considered. In our setting, the optimal coalition structure must yeild a profit, a non-nagative utility, to the system. In the worst case, the system's profit is 0—each agent is a singleton coalition and cannot produce anything at all.

3 DISTRIBUTED ALGORITHM FOR COALITION FORMATION

3.1 Deliberating Process

In the following, we will indentify a coalition by the indentifier of its coalition centre agent. Thus the coalition S^k will have agent a_k as its centre. Hence b^S represents the resource vector of S^k. The reasoning described below is conducted by the coalition centre agent for each coalition. Given a coalition S^k, let G^k refer to the set of goods whose resource requirements are fully or partially satsified by b^S, the resources available in S^k (excluding goods whose resource requirement might be trivially satisfied because these are 0). For each good $g_j \in G^k$, the coalition centre agent a_k ranks agents not currently in its coalition on a per good basis. For each resource r_i of good g_j, agent a_k ranks non-member agents by computing for each $a_l \notin S^k$, whose $b^l_i > 0$, the value π^j_i—its pro-

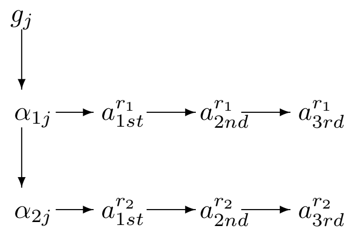

Figure 1: Agents are ranked by their potential profit per each resource of a good.

portional contribution to the profit of the good (using its fraction of the resource requirements for that good provided by the a_l) minus the (pair-wise) collaboration cost between a_l and a_k, i.e.,

$$\pi^j_i = \frac{b^l_i}{\alpha_{ij}} p_j - c_{kl}.$$

The agent a_k uses this proportional contribution π^j_i to construct a binary tree for each g_j. Figure 1 illustrates the tree T^j of g_j. The only child of the root g_j is the first resource α_{1j}, whose left child is the second resource α_{2j}, and so on. For each α_{ij}, its right child is either i) null if $\alpha^i_j = 0$, or ii) the agent $a^{r_i}_{1st}$, whose pi^j_i value is the greatest. The right child of $a^{r_i}_{1st}$ is the agent $a^{r_i}_{2nd}$, whose π^j_i value is the second greatest, and so on. Every time a_k wants to produce additional units of g_j, it traverses the tree down to the appropriate resource r_i and add more agents into its coalition based on b^S.

The agent a_k uses b^S to determine additional resources needed to produce additional units of a good g_j. For each $g_j \in G^k$ and resource r_i,

$$\beta^j_i = I(\alpha_{ij}) - b^S_i,$$

where $I \in \mathbb{Z}^+$ is the smallest integer such that $\beta^j_i > 0$, represents the amount of r_i that coalition S^k lacks to produce good g_j, provided the amount is non-nagative ($\beta = 0$ otherwise). The *indicative vector*, $\beta^j = [\beta^j_i]_{1\times n}$, represents un-met requirements for each resource r_i of good g_j.

The agent a_k uses the indicative vector β^j to help collecting additional coalition members into its coalition. If the agent a_k wants to produce an additional unit of g_j, it identifies the resource that is needed the most, $\beta^j_{i^*} = max^n_{i=1}(b^j_i)$, from the indicative vector. It locates the node $\beta^j_{i^*}$ in T^{g_j} and collects the next available agent $a^{i^*}_l$ into the coalition. The total resources of the coalition b^S is updated. Each β^j_i of indicative vector will be subtracted by it corresponding b^l_i. The agent a_k keeps adding more agents into its coalition until there are enough resources to produce an additional unit of g_j, i.e., $\beta_i^j > 0\ \forall i$. This algorithm to collect additional agents into the coalition is shown in 1.

Algorithm 1 Select additional agents

Require: the present coalition S
Require: the focused good g
initialize additional agents $S' = \emptyset$
get the coalition's resource b^S
get the indicative vector $\beta^{g'}$
identify the most needed resource r_{i^*}
while $r_{i^*} > 0$ **do**
  locate next available agent $a_l^{r_j}$
  if $a_l^{r_j}$ ¡0 **then**
    break
  end if
  set $S' = S' \cup a_l^{r_j}$
  for all β_i^j **do**
    set $\beta_i^j = \beta_i^j - b_i^l$
  end for
  identify the most needed resource r_{i^*}
end while
return S'

So far, the agent a_k knows that if it wants to produce at least an additional unit of g_j, it needs to add agents from S' into its S^k. The agent a_k create a trial coalition by merging S' into S. Since each new agent may posses other resources not required for producing g_j, the trial coalitions may find a better solution for producing goods. Hence the profits υ of trial coalitions vary. The additional agents $S*$ are those S' that provide the highest additional profit υ^* and are kept as the basis for further growing coalition, S^*. The subalgorithm for selecting the most profitable members is shown in algorithm 2.

Algorithm 2 Select the most profitable members

Require: A coalition S
Require: ranking trees T^G
set highest profit $\upsilon^* = 0$
set most profitable members $S^* = null$
for all $g_j \in G$ **do**
  if S is not capable of producing g_j **then**
    continue
  end if
  get additional agents S'
  set trial coalition $S'_j = S \cup S'_j$
  compute trial coalition's profit $\upsilon_{S'_j}$
  if $\upsilon_{S'_j} > \upsilon_{S^*}$ **then**
    set $S^* = S'_j$
  end if
end for
return S^*

At the begining of deliberating, the agent a_k considers itself a singleton coalition. It create the ranking tree T^G of all agent for each good. Then it keeps adding the most profitable agents, S^*, into the coalition. This will keep the coalition's marginal profit grows while the size of the coalition is growing. It also reduces the number of coalitions each agent a_k has to maintain. The new coalition S will be added to a list L^+ of profitable coalitions. This process repeats until there are no more agents left or it cannot find any more profitable agents. The main algorithm is shown in algorithm 3.

Algorithm 3 Main

set $L^+ = \emptyset$
create a singleton coalition $S = \{a_k\}$
set $A' = A - \{a_k\}$
create ranking trees T^G for all goods
select the most profitable members S^*
while $A' \neq \emptyset$ and $S^* \neq \emptyset$ **do**
  set $A' = A' - S^*$
  set $S = S \cup S^*$
  set $L^+ = L^+ \cup S$
  select the most profitable members S^*
end while

3.2 Coalition Formation Algorithm

Once each agent finishes its deliberation, it ranks all of its coalitions by profit. Let S^- be a non-profitable coalition, whose value $\upsilon_{S^-} \leq 0$. and S^+ be a profitable coalition, whose value $\upsilon_{S^+} > 0$.

Lemma 1 *Any S^- coalition can be replaced by a set of its members' singleton coalitions, whose $\upsilon_{a_{k \in S}} \geq 0$, such that the coalition structure's value will not be decreased.*

Therefore, all non-profitable coalitions can be ignored. Each agent will prune all of the non-profitable coalitions, if there is any. The remaining coalitions are profitable. In fact, our algorithm in deliberation process can simply prevent this happening using its tree T^G. It always generate profitable coalitions. Obviously, each singleton coalition is non-negative. Hence, non-profitable coalitions must not exist in the coalition structure. Given that the deliberation algorithm generates all profitable coalitions among agents inclusively, agents can i) exchange information about coalitions generated and their singleton coalitions, and ii) decide form coalitions that yield the optimal coalition structure value.

Proposition 1 *The optimal coalition structure can be constructed by profitable coalitions generated by agents and their singleton coalitions.*

Next step each agent sends information about coalitions it has generated to each other. For each coalition size, each agent can further reduce the number of coalitions it has by deleting non-centred coalitions and those whose values are non-maximal. Up to this point, the remaining coalitions are likely to be in the coalition structure. Agents exchange information again and compute optimal coalition structure using existing algorithm, e.g., (Sandholm et al., 1999). The algorithm is shown below.

1. Each agent a_k deletes non-profitable coalitions from its list
2. Agent a_k sends its list of profitable coalitions to each coalition member
3. For each coalition size, agent a_k deletes all coalitions that their centre are not the agent itself and those that do not yield the maximal value
4. Each agent sends the remaining coalitions to each member
5. Each agent compute the optimal coalition structure using existing algorithm ((Sandholm et al., 1999))
6. The optimal coalition structure will be recognized by agents.

4 EXPERIMENT

We conduct experiment of our algorithm within a range of $10-100$ agents. In each round, the agents number increases by 5. The number of goods and resources are equal and increase by 1 in every 2 rounds. In each round, the technology matrix, agents' resources and cooperation costs among agents are randomly generated with uniform distribution. The number of each resource α_{ij} in the technology matrix is in the range $0-10$. The prices of the goods are in the range of $10-20$ while the cooperation costs are in the range of 0 and the number of agents in that round, e.g., $10, 15, \ldots$. As our algorithm deals with non-superadditive environments, this setting tends to increase the cooperation cost of a coalition as its size grows. Hence it forces agents to work harder to form profitable coalitions and to achieve optimal coalition structure. Both algorithms uses the Simplex algorithm to find the optimal solution for each coalitions. The revenue generated is subtracted to achieve the coalition's profit.

The Table 1 compares the average deliberation time agents spent using exhaustive search and that using our algorithm. The time is measured in milliseconds. We experienced that exhaustive search hardly make progress after the number of coalitions generated exceeded 2.5 millions. As shown in the table, the time spent on deliberation using exhaustive search was approximately doubled as the number of agents increased by 1. With 20 agents, the time spent on deliberation using exhaustive search is far larger than that using our algorithm. Our computer system could not carry on simulations any further after we reached 35 agents using exhaustive search. We continued experiment using our algorithm until the number of agents reached 100. (Although we carried on the experiment up to 300 agents, the results are not shown here.) Since the number of coalitions generated are small, the optimal coalition structure can be found more rapidly.

Table 1: This table compares the average deliberation time of each agent using our algorithm against exhaustive search. Our algorithm outperforms exhaustive search after the number of agents exceeds 35 (exhaustive time not available—NA).

No. of Agents	No. of Goods Resources	Exhuastive Search	Our Search
10	4	781	121
15	4	42269	123
20	5	1272703	197
25	5	5092317	234
30	6	19384629	607
35	6	80429663	1608
40	7	NA	1696
50	8	NA	4730
60	9	NA	13346
70	10	NA	24298
80	11	NA	23276
90	12	NA	26933
100	12	NA	81845

Having pruned a large number coalitions, the number of remaining coalitions are small. Hence the number of coalition structures are small. Applying existing algorithm can intuitively achieve optimal coalition structure in timely fashion.

5 RELATED WORK

Shehory et. al (Shehory and Kraus, 1995) propose an algorithm to allocate tasks to agents in distributed problem solving manner, i.e., agents try to maximise the utility of the system. They consider a domain where a task composed of multiple subtasks, each of which requires specific capacity. These tasks have to be carried out by agents who have specific capacities to carry out tasks. Each agent prepares its list of candidate coalitions and proposes to other agents.

Shehory et. al. (Shehory and Kraus, 1996) study overlapping coalition formation in distributed problem solving systems in non-superadditive environments. Althhough agents can belong to multiple coalitions at the same time, agents execute one task at a time. The task allocation process is completed prior to the execution of the tasks. Agents are group-rational, i.e., they form coalition to increase the system's payoff.

Sandholm et. al. (Sandholm and Lesser, 1995) analyze coalition formation among self-interested agents who are bounded-rational. They consider deliberation cost in terms of monetary cost. The agents' payoffs are directly affected by deliberation cost. In their work, agents agree to form coalition and each of the agents can plan to acheive their goals.

Soh et. al. (Soh and Tsatsoulis, 2002) propose an integrated learning approach to form coalition in real time, given dynamic and uncertain environments. This work concentrates on finding out potential coalition members by utilising learning approach in order to quickly form coalitions of acceptable quality (but possibly sub-optimal.)

Sandholm et. al. (Sandholm et al., 1999) study the problem of generating coalition structure generation. Since the number of coalition structure can be very large for exhaustive search, they argue whether the optimal coalition structure found via a partial search can be guaranteed to be within a bound from optimum. They propose an anytime algorithm that estiblishes a tight bound withing a minimal amount of search.

6 CONCLUSION AND FUTURE WORK

Coalition formation is an important area of research in multi-agent system. The problem of generating optimal coalition structure, the partitioning of a set of agents such that the sum of all coalitions' values within the partitioning is maximal, is an important issue in the area. The small number of existing studies assume each coalition value is known a priori. Such assumption is impractical in realworld settings. Furthermore, finding all coalition values becomes intractable for a relatively small number of agents.

We study coalition formation among fully cooperative agents where each coalition value is not known a priori. We proposes a distributed algorithm to generate optimal coalition structure by reducing the number of coalitions to be involved. Since they do not help increasing coalition structures' values, the non-profitable coalitions are not generated by the deliberation algorithm. If there is any, each agent delete them first. Then the information of remaining coalitions will be exchanged among agent. For each coalition size, each agent prunes its list of coalitions again deleting those, whose centres are not the agent itself and those whose values are not maximal within their coalition sizes. Remaining coalitions will be exchanged among agents again. Lastly, each agent uses existing algorithm (Sandholm et al., 1999) to compute optimal coalition structure.

Although this algorithm helps reducing number of coalitions involved in generating optimal coalition structure, there is always rooms to improve. We want to further reduce the number of coalitions generated by each agent and want to make sure that coalitions generated are highly likely to be in the coalition structure. Furthermore, we want to improve the coalition algorithm rathan using existing one.

REFERENCES

Dang, V. D. and Jennings, N. R. (2004). Generating coalition structures with finite bound from the optimal guarantees. In *Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 2 (AAMAS'04), pp. 564-571*.

Kahan, J. P. and Rapoport, A. (1984). *Theories of Coalition Formation*. Lawrence Erlbaum Associates, Hillsdale, New Jersey.

Owen, G. (1975). On the core of linear production games. *Mathematical Programming 9 (1975) 358-370*.

Sandholm, T., Larson, K., Andersson, M., Shehory, O., and Tohmé, F. (1999). Coalition structure generation with worst case guarantees. *Artif. Intell.*, 111(1-2):209–238.

Sandholm, T. and Lesser, V. (1995). Coalition Formation among Bounded Rational Agents. *14th International Joint Conference on Artificial Intelligence*, pages 662–669.

Shehory, O. and Kraus, S. (1995). Task allocation via coalition formation among autonomous agents. In *Proc. of IJCAI*, pages 655–661.

Shehory, O. and Kraus, S. (1996). Formation of overlapping coalitions for precedence-ordered task-execution among autonomous agents. In *ICMAS-96*, pages 330–337.

Soh, L.-K. and Tsatsoulis, C. (2002). Satisficing coalition formation among agents. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 1062–1063. ACM Press.