Chapter 1

A STABLE AND EFFICIENT SCHEME FOR TASK ALLOCATION VIA AGENT COALITION FORMATION *

Cuihong Li
Robotics Institute
Carnegie Mellon University
cuihong@cs.cmu.edu

Katia Sycara
Robotics Institute
Carnegie Mellon University
katia@cs.cmu.edu

Abstract Task execution in a multi-agent, multi-task environment often requires allocation of agents to different tasks and cooperation among agents. Agents usually have limited resources that cannot be regenerated, and are heterogeneous in capabilities and available resources. Agent coalition benefits the system because agents can complement each other by taking different functions and hence improve the performance of a task. Good task allocation decision in a dynamic and unpredictable environment must consider overall system optimization across tasks, and the sustainability of the agent society for the future tasks and usage of the resources. In this paper we present an efficient scheme to solve the real time team/coalition formation problem. Our domain of applications is coalition formation of various UAVs for cooperative sensing and attack. In this scheme each agent bids the maximum affordable cost for each task. Based on the bidding information and the cost curves of the tasks, the agents are split into groups, one for each task, and cost division among the group members for each task is calculated. This cost sharing scheme provably guarantees the stability in cost division within each

*This research was supported in part by AFOSR grant F49620-01-1-0542 and by AFRL/MNK grant F08630-03-1-0005.

coalition in terms of the core in game theory, therefore achieves good sustainability of the agent society with balanced resource depletions across agents. Simulation results show that, under most conditions, our scheme greatly increases the total utility of the system compared with the traditional heuristics.

Keywords: Coalition formation, Task allocation, Multi-agent coordination

1. Introduction

Task execution in a multi-agent, multi-task environment often requires allocation of agents to different tasks and coalition formation among agents. Good task allocation and coalition formation decisions must consider overall system optimization across tasks as well as agent heterogeneity in resources and capabilities. The cost division among coalition members is also important to sustain a well-functioning agent society in a dynamic and uncertain environment.

Consider a fleet of UAVs in missions over time to destroy targets that appear dynamically. The UAVs have different specializations in capabilities, although each can perform multiple functions subject to different costs (fuel). UAVs have limited resources (fuel) and cannot be refuelled during the process. The available resources of the UAVs are different because of the consumption of fuel on different levels. Coalitions of UAVs are desirable in executing the tasks because UAVs can complement each other by undertaking different functions that could be done more effectively with more participating UAVs. At each time there may be multiple targets that require different capabilities of UAVs to destroy. A UAV is capable of executing more than one task to destroy a target, and the UAVs have to be allocated to different coalitions/teams for different targets. The cost of, or the resource to be consumed by, a coalition to execute a task is deterministic. A coalition formation scheme decides the allocation of the UAVs into different coalitions, one for each target. A cost division scheme determines how much cost/effort a UAV should pay in participating in a coalition to destroy a target. We want the coalition configuration to be efficient so that the total performance of executing the multiple tasks is optimized. The cost sharing rule should be fair so that the resource depletions of the UAVs are balanced, and as many as possible UAVs can survive as long as possible through the usage horizon and complement each other in executing future tasks.

The task allocation and coalition formation problem can be characterized by the following important properties or requirements:

- Coalition formation: Agents can form a coalition and execute a task together. Agent coalitions benefit the system because agents can improve the task performance by taking different complementary functions. When there are more agents in a coalition, the average cost per agent for executing the task decreases. The relation between the total cost for executing a task and the number of agents[1] is characterized by a *cost curve*. Although the average cost decreases, the total cost for executing a task may increase with the number of agents in the coalition because more agents are involved. But the marginal cost imposed by an agent does not increase with the number of agents, in other words, the total cost is a non-decreasing concave function of the number of agents. Because of the contribution made by an agent to the task performance, it is always efficient to include an agent in a coalition if the agent can afford the marginal cost.

- Heterogeneity: The heterogeneity is in both the tasks and agents. The tasks are heterogeneous because the capability requirements and cost curves are different. An agent may be qualified in capabilities for participating in some tasks but not in others. The efficiency of an agent in executing a task is also different from executing other tasks. Agents are heterogeneous in capabilities and available resources. We use the *maximum affordable cost* of an agent as the measurement of the suitability of an agent to execute a task. The maximum affordable cost is a function of both the capability and the available resource. The maximum affordable cost is higher when an agent has more resources available, or the agent is more specialized in the capability desired for the task. In a quasi-linear form the maximum affordable cost can be expressed as the available resources plus a function of the capability that calculates the resource saving based on the capability. The index of the maximum affordable cost and the cost curve allow the comparison of the efficiency of allocating different agents with different capabilities and resources to different tasks.

- Sustainability: We want as many as agents to participate in the tasks to improve the efficiency, and also to minimize to the extent possible the depletion of resources across agents so as to retain agents for future tasks. It is not desirable to have some agents consume their resources much faster than the others. Sustainability does not mean that all agents share the cost equally. The agents that can afford more cost are reasonably assumed to share more cost because they have a larger base.

The objectives of the coalition formation scheme include: (1) to optimize the total performance of the tasks, and (2) to divide the cost among agents in a fair way to achieve good sustainability. The first objective is important since it assures the efficiency in executing the current tasks by forming coalitions and matching agents with the tasks. The second objective considers the efficiency in executing future tasks by balancing the resource depletions across agents for current tasks.

If we consider the performance of a task as the value of the coalition for that task, the coalition formation problem can be translated into a weighted set packing problem, modelled as a set covering problem, which is well known as a NP-complete problem [Arkin and Hassin, 1999]. Task allocations often involve a large agent group in a scale of thousands or much higher. Additionally, time for calculating a solution is usually limited so the coalition formation must be performed in real time. Therefore an efficient algorithm is desired to ensure the real-time application for large scale problems. We present an efficient coalition formation scheme in polynomial time for the coalition formation problem. In this scheme each agent bids the maximum affordable cost for each task that it is capable of. Based on the bidding information and the cost curves of the tasks, the coordinator splits the agents into groups, one group for each task.

We use the *core*, a concept from cooperative game theory [Moulin, 1988], to measure the fairness of cost division in a coalition. If the cost division is in the core, there are no agents that can get more total utility by deviating from the coalition and forming a coalition by themselves. Therefore a fair cost division scheme in the core achieves the *stability* of a coalition. In the task allocation situation the utility of an agent from a coalition is defined as the maximum affordable cost minus the cost to share. Agents in a coalition may pay different costs according to their maximum affordable costs.

As optimizing the total coalition values is, in general, computationally too complex as mentioned above, we take the following approach. When forming a coalition configuration, we try to maximize the value of the most valuable coalition, then maximize the value of the second valuable one, and continue recursively. Then we divide each coalition's cost within the coalition. We prove that our coalition formation scheme based on this approach guarantees the stability of cost division within each coalition in terms of the core in game theory. Simulation results show that, under most conditions, our scheme greatly increases the total utility of the system compared to the traditional heuristics.

This paper is organized as follows. Section 2 describes prior work. In section 3 the problem is formulated. In section 4 we present the coalition formation scheme in detail. Section 5 analyzes the stability of the coalition formation scheme. Section 6 provides the experimental results. We conclude in section 7.

2. Prior Work

Works in game theory and microeconomics such as [Moulin, 1988, Moulin, 1995] have provided concepts of coalition and its stability. A coalition is a set of agents which cooperate to achieve a common goal, and the stability requirement is that the outcome of a coalition be immune to deviations by individual agents or subsets of agents. Those concepts are important as criteria of coalition formation schemes, and we justify our scheme based on the core, one of the stability concepts in game theory. However, game theory does not provide efficient algorithms for coalition formation.

Finding the maximum total utility of coalitions can be translated into the weighted set packing problem [Arkin and Hassin, 1999]: Given a set B and collection of its subsets $Col = \{C_0, ..., C_n\}$ such that each $C_i$has its value $v(C_i)$, find a sub collection $SubCol \subset Col$ of pairwise disjoint sets such that $\sum_{C_i \in SubCol} v(C_i)$ is the maximum among all sub collections. We can interpret B as the set of agents, SubCol as a collection of coalitions, and v as a coalition's value. The weighted set packing problem is NP-complete, and several optimization algorithms have been proposed [Arkin and Hassin, 1999, Chandra and Halldorsson, 1999]. However, these algorithms rely on the assumption that the maximum size of subsets in $SubCol$ is bounded by a relatively small number k. In the context of task allocation, bounding the coalition size by a small number is impractical.

Research on multi-agent systems also has investigated coalition formation of agents. [Sandholm et al., 1999] proved that, for a given set A, searching the best coalition configuration among $\{\{A\}\} \cup \{\{A_1, A_2\} \mid A_1 \cup A_2 = B, A_1 \cap A_2 = \emptyset\}$ guarantees that the largest coalition value found is within a bound from the optimal one by $|B|$, and that no other search algorithm can establish any bound while searching only $2^{|A|-1}$ coalition configurations or fewer. This result means, without some kind of heuristics or assumptions, bounding the group's total utility is virtually impossible because $|A|$ could be large.

[Shehory and Kraus, 1996, Shehory et al., 1997] have provided distributed coalition formation schemes for multi-agent systems mainly focusing on increasing the group's total utility. They also limit the highest coalition size by an integer k, which means the algorithms proposed cannot be applied to large coalitions. [Shehory and Kraus, 1999] aims both to increase the total utility and to reach the stable payoff division among agents. Yet, the algorithms restrict the size of each coalition to guarantee the practical computation time.

[Lerman and Shehory, 2000] has proposed a new model of coalition formation, and applied it to coalition formation among buyer agents in an e-marketplace. Their model treats agents as locally interacting entities; an agent may create a coalition when it encounters another agent, join an existing coalition, or leave a coalition. The model describes global behavior of a set of agents from the macroscopic view point by differential equations, and simulates well how buyer coalitions evolve and reach the steady state. However, the model does not assist individual agents to form a coalition nor to negotiate surplus distribution.

3. Problem formulation

The terms and notations are defined and interpreted as follows.
Tasks and Cost Curves: $T = \{t_1, t_2, ..., t_m\}$ denotes the set of tasks. Let N and R be the set of natural numbers and real numbers respectively. A *cost curve* of $t_i$is represented as a descending function $p_i : N \to R$; $p_i(n)$ is the average cost per agent when n agents join the coalition for the task t_i.
Agents: Let $A = \{a_1, a_2, ..., a_n\}$ denote the group of agents to be allocated for the tasks. Agent a_k's maximum affordable cost for t_i is represented by $r_{ki} \geq 0$. The *maximum affordable cost* of an agent for a task comprises the agent's available resource, and the agent's capability for executing the task. An agent a_k's *utility* from participating in the task $t_i$at the cost p is defined as $u_{ki} = r_{ki} - p$.
Coalitions: Let $C_i \subset A$ denote a coalition for the task t_i. A coalition configuration is $Conf = \{C_1, ..., C_m\}$ such that $C_i \cap C_j = \emptyset$ for $i \neq j$. $C_i$can be empty. $Conf$ does not necessarily satisfy $\cup_{i=1,...,m} C_i = A$; some agents in A may not belong to any coalitions because their maximum affordable costs are too low.

The *value* $v_i(C)$ of a coalition C for the task t_i is defined as

$$v_i(C) \stackrel{\text{def}}{=} \sum_{a_k \in C} r_{ki} - cost_i(C)$$

where $cost_i(C)$ is the cost paid by the coalition C to execute the task t_i, i.e., $cost_i(C) = |C| \cdot p_i(|C|)$. ($|C|$ denotes the cardinality of C.) Since the cost of the coalition $cost_i(C)$ is shared by the agents in C, the value of a coalition is equal to the sum of the utilities of the agents in the coalition. A coalition C is formed for the task t_i only if it can afford to execute the task t_i, i.e., $v_i(C) \geq 0$. The higher the value $v_i(C)$, the more efficient the allocation of the coalition C to the task t_i. It is because a higher value $v_i(C)$ means that the task t_i requires lower cost from the coalition C, or the agents in the coalition C are more capable of executing the task t_i. To maximize the values of the coalitions is consistent with the objective to maximize the overall performance of the tasks.

A cost division scheme c_k, $a_k \in C$ for the coalition C_i is in the *core* if and only if there does not exist $S \subset C_i$ so that the value $v_i(S)$ of the coalition S is greater than the sum of the utilities of agents in S from the coalition C_i, i.e., $v_i(S) \leq \sum_{a_k \in S} u_{ki}$ for any $S \subset C_i$.

The problem of coalition formation can be formulated as

$$\max_{C_i \subset A, i=1,\ldots,m} \sum_{a_k \in C_i} r_{ki} - cost_i(C_i)$$

so that $C_i \cap C_j = \emptyset$ for $i \neq j$; and for each $i = 1, \ldots, m$, find c_k for each $a_k \in C_i$ so that for any $S \subset C_i$

$$\sum_{a_k \in S} (r_{ki} - c_k) \geq \sum_{a_k \in S} r_{ki} - cost_i(S).$$

4. Coalition Formation Scheme

We give a simple example to illustrate the model and the approach. Assume there are three tasks which have the same cost curve shown in Figure 1.1. The horizontal axis shows the number of agents in the coalition for a task, and the vertical axis indicates the average cost per agent. For instance, if there are three agents in the coalition, the average cost goes down to 90. Table 1.1 shows five agents to be allocated to these tasks. Each row shows an agent's affordable cost for each task that the agent is capable of performing. For instance, $a4$ is capable of participating in $task1$ or $task2$ and the maximum costs are 85 and 95 respectively.

Table 1.1. Sample agents' maximum affordable costs

agent	task0	task1	task2
a0	100		70
a1	80	95	95
a2		95	
a3		65	
a4		85	95

The main issues we study are how to split the agents into coalitions, and how to distribute the cost of the group among agents. In this example, there are one possible coalition for task0 ($\{a0\}$), three for task1

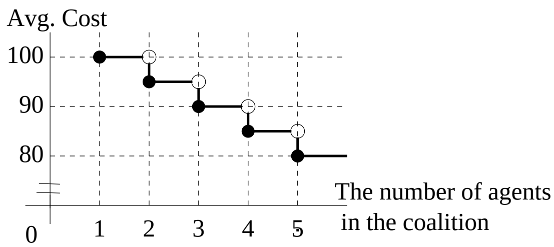

Figure 1.1. A sample cost curve

$(\{a1, a2\}, \{a1, a2, a4\}, \{a1, a2, a3, a4\})$, and one for task2$(\{a1, a4\})$. Our scheme derives the coalition configuration shown in Table 1.2; $\{a1, a2, a4\}$ as a 'task1' coalition has the largest surplus among all possible coalitions, and $\{a0\}$ as a 'task0' coalition is the only coalition which the rest of the agents can form. Each cell in the table contains the agents' cost to pay and the maximum affordable cost between parentheses. The costs to pay in a coalition differ depending on agents' maximum affordable costs. For example, $a1$ pays 92.5 ($a1$'s maximum affordable cost is 95), while $a4$ pays only 85 ($a4$'s maximum affordable cost is 85). If $a4$ did not join the coalition, $a1$ and $a2$ would have to pay 95 for executing task1. On the other hand, the coalition does not include $a3$ because $a3$ would bring no benefit to the coalition.

Table 1.2. A sample coalition configuration

agent	task0	task1	task2
a0	100(100)		
a1		92.5 (95)	
a2		92.5 (95)	
a3			
a4		85.0 (85)	

The rest of this section formally explains this coalition formation scheme.

4.1. Coalition Configuration Algorithm

As we have mentioned, it is not computationally feasible to obtain the optimal coalition configuration that maximizes the total coalition values. We design a computational heuristic to configure the coalitions that achieves fairly good efficiency in reasonable time. In the heuristic approach a coalition configuration $Conf = \{C_1, ..., C_m\}$ is formed so that the value of the most valuable coalition is maximized first, and then the utility of the second most one is maximized, etc. This algorithm is formalized as follows.

Algorithm 1: Coalition Configuration

1 Set $Conf = \emptyset$, $RestOfTaskIDs = \{1, 2, ..., m\}$ and $RestOfAgents = B$.

2 For every $i \in RestOfTaskIDs$, calculate a candidate coalition $C_i^* \subset RestOfAgents$, the set with the largest value as a $t_i$coalition, as follows.

$$AC_i \stackrel{\text{def}}{=} \{C \subset RestOfAgents \mid v_i(C) \geq 0\}$$
$$VC_i \stackrel{\text{def}}{=} \{C \in AC_i \mid v_i(C) \geq v_i(C') \, for \, \forall C' \in AC_i\}$$

(AC_i is the set of admissible coalitions, VCi the set of the most valuable coalitions.)

Select any one of $C_i^* \in VC_i$ if $VC_i \neq \emptyset$, $C_i^* = \emptyset$ otherwise. $Cand \stackrel{\text{def}}{=} \{C_i^* \mid i \in RestOfTaskIDs\}$ denotes the set of all candidates.

3 If every $C_i^* \in Cand$ is empty, stop this procedure.

4 If there exist non empty candidates in $Cand$, select one of them with the largest utility within $Cand$; that is, select C_i^*such that $v_k(C_k^*) \geq v_i(C_i^*)$ for $\forall C_i^* \in Cand$. Let $Conf = Conf \cup \{C_k^*\}$, $RestOfTaskIDs = RestOfTaskIDs \backslash \{k\}$, and $RestOfAgents = RestOfAgents \backslash C_k^*$.

5 Go back to Step 2 if $RestOfTaskIDs \neq \emptyset$ and $RestOfAgents \neq \emptyset$. Otherwise, stop this procedure.

This algorithm can be considered as a variation of the greedy algorithm for the weighted set packing problem [Chandra and Halldorsson, 1999]. In general, finding a subset of A which has the largest value among all subsets could require $O(2^n)$ computations at worst.

However, we have an efficient algorithm to calculate our coalition configuration with order $O(n \cdot \log n)$, where n is the number of agents in B, and we assume the number of tasks can be bounded from above by a positive number K independently from n. This assumption makes sense even for very large coalitions. The complexity of searching C_i^*at each recursion is $O(n \cdot \log n)$ computations as explained below, each recursion includes at most K times of the search, and all coalitions are configured within K recursions. Thus, the entire complexity of the coalition configuration is $O(n \cdot \log n)$ computations.

To search C_i^*at each recursion, first arrange all agents in $RestOfAgents$ in the descending order in terms of the maximum affordable cost for

$t_i(O(n \cdot \log n)$ computations). Then calculate the utility of subsets $C_{ij} \subset RestOfAgents$ for $j = 1, ..., t$ (t is at most n) which includes the top j agents in terms of the maximum affordable cost for t_i, and select C_i^*out of $\{C_{i1}, ..., C_{it}\}$. This requires $O(n)$ computations. (This algorithm is supported by Proposition 2 in the next section.)

4.2. Cost Sharing in a Coalition

Agents in a coalition share the cost within the coalition. Let the cost shared by the agent $a_k \in C_i$ be $c_k \geq 0$. The cost sharing rule is defined as follows.

Definition 1: Cost Sharing Rule When a coalition $C_i$has value $v_i(C_i) > 0$, the cost c_k shared by an agent $a_k \in C_i$ is

$$c_k \stackrel{\text{def}}{=} \begin{cases} h_{C_i} & (a_k \in \overline{C_i}) \\ r_{ki} & (a_k \notin \overline{C_i}) \end{cases}$$

where $h_{C_i}$and $\overline{C_i}$satisfies the following conditions:

$$cost_i(C_i) = |\overline{C_i}| \cdot h_{C_i} + \textstyle\sum_{a_k \in C_i \setminus \overline{C_i}} r_{ki},$$
$$\overline{C_i} \stackrel{\text{def}}{=} \{a_k \in C_i \mid h_{C_i} \leq r_{ki}\}.$$

Figure 1.2 illustrates this definition. The graph shows each agent's maximum affordable cost, her share of cost, and her actual utility. Agents in $\overline{C_i}$pay h_{C_i}which is equal to or lower than their maximum affordable costs. Others in $C_i \setminus \overline{C_i}$ pay just their maximum affordable costs.

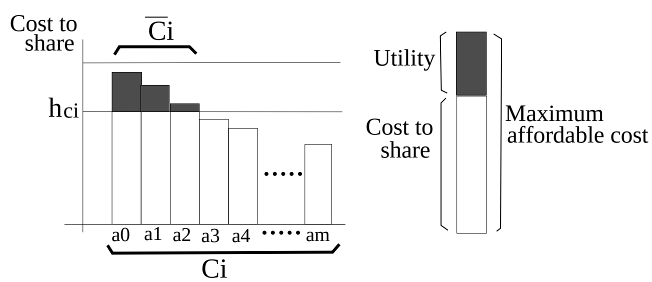

Figure 1.2. The cost sharing rule

5. Stability of Coalition Configuration

As agents in a coalition pay different costs under our scheme, a fair share of the cost is essential to sustain the agent society, and guarantee the stability of the coalitions if agents are autonomous to choose the tasks driven by self-interest. If agents do not trust the fairness, they may not join a coalition, nor provide their maximum affordable costs truthfully, which could prevent successful coalition formation.

In this section, we discuss our scheme's stability in terms of the core in game theory[Osborne and Rubinstein, 1994, Moulin, 1988]. The core is defined as follows.

Definition 2: The Core [Osborne and Rubinstein, 1994]
A coalitional game with transferable payoff consists of (1) a finite set C of players, and (2) a utility function v which associates with every nonempty subset $S \subset C$ a real number $v(S)$. The core of the coalitional game with transferable payoff $< C, v >$ is

$$Core = \{(u_a)_{a \in C} \mid v(C) = \sum_{a \in C} u_a, v(S) \leq \sum_{a \in S} u_a \, for \, \forall S \subset C\}$$

In general, the core may contain multiple elements, or it may be empty. In our case each coalition $C_i$has a nonempty core. We prove that the cost distribution calculated by our cost sharing rule is within the core as the next proposition states.

Proposition 1 (Stability of a coalition)
For $\forall C_i \in Conf$, the cost distribution $(c_k)_{a_k \in C_i}$ calculated using the cost sharing rule (Definition 1) is in the core of the coalitional game with transferable payoff $< C_i, v_i >$. That is, $v_i(S) \leq \sum_{a_k \in S} u_k$ holds for $\forall S \subset C_i$.

The stability condition defined by the core is that no subset of agents in a coalition can obtain utility that exceeds the sum of the current utility of the members in the subset. Thus, even self-interested agents in a coalition would not be motivated to deviate from the coalition.

There can be multiple cost distributions within the core. Proposition 2 and 3 below characterize our cost distribution, and we expect these propositions will encourage an agent to tell its maximum affordable cost truthfully. (Note that Proposition 1 above is proved via Proposition 2 and 3. The proof is provided in Appendix.)

Proposition 2 (Members in a coalition)
At each recursion of coalition configuration in Algorithm 1, for $\forall a_k \in RestOfAgents$ and $\forall i \in RestOfTaskIDs$, if $\exists a_h \in C_i^*$ such that $r_{ki} > r_{hi}$, then $a_k \in C_i^*$.

Proposition 2 means that C_i^*consists of the top $|C_i^*|$ agents in terms of the maximum affordable cost. The higher an agent's maximum affordable cost is, the more likely it will be able to join a coalition.

Proposition 3 (Cost sharing)
At each recursion of coalition configuration in Algorithm 1, for $\forall i \in RestOfTaskIDs$ and $\forall C \in AC_i,\ h_{C_i^*} \leq h_C$.

The last proposition assures that, at each recursion, the highest cost anybody in C_i^*pays, $h_{C_i^*}$, is the lowest among all the costs afforded by any sets of agents.

6. Evaluation

We have conducted a series of simulations to evaluate the effectiveness of our coalition formation scheme in increasing the system's performance. We simulated agents' behaviors under three coalition formation schemes (our scheme, a traditional scheme and an optimal scheme) under particular conditions, and compared them by the groups' total utility.

6.1. Assumptions

We make the following assumptions.

Tasks and Cost Curves: The cost curve for each task is a predetermined non-increasing step function. The highest value of the function is called the *highest average cost*. There is no limit to how many agents can join a coalition.

Agents: An agent has several choices of tasks. We model the distribution of capabilities for multiple tasks by RAMT (the Ratio of Agents who are capable of Multiple Tasks). RAMT is an array $(ra_1, ..., ra_m)$, where m is the number of tasks and $ra_1 + ... + ra_m = 1$ holds. ra_i denotes the ratio of agents who can participate in i tasks out of m tasks. For instance, in the example shown by Table 1.1 in Section 4, RAMT is (0.4, 0.4, 0.2); out of five agents, two agents can only participate in one task, two agents can work for two tasks, and one agent can take part in three tasks. RAMT does not specify which particular tasks each agent is qualified for. An agent randomly selects the tasks that it is capable of performing.

Some agents' maximum affordable costs (MAC) for a given task may be greater or equal to the highest average cost. These agents are sure to be included in the candidate coalitions because they do not need the joining of other agents to form a coalition with a non-negative value. Let the ratio of the number of the agents with MACs no less than the highest average cost be called RMH (the Ratio of Maximum affordable costs which are the Highest average cost). Other MACs for the task are randomly distributed between its highest average cost and a certain lower value. We denote the lowest possible MAC by LAC. The environment (other agents' behaviors, cost curves, etc.) does not affect agents' capabilities or maximum affordable costs.

An Optimal Scheme: At every simulation, we calculate an optimal coalition configuration for comparison. The optimal scheme exhaustively searches all possible coalition configurations and selects one of the configurations which has the largest value[2]. Agents in a coalition share their cost within the coalition, but the optimal scheme does not care about how to share.

A Traditional Scheme Under a traditional coalition formation scheme, each agent first selects one task, and then the agents that select the same task and can afford the cost are formed as a coalition. All agents in a coalition pay the same cost.

An agent can know the cost curve, current average cost and the number of agents in each coalition at any time. An agent $a_k$selects one task out of the tasks it is qualified for by following one of the selection rules listed below.

Random Rule: Randomly Select a task.

Lowest Price Rule: Select a task whose current average cost is the lowest in proportion to the highest average cost.

Highest MAC Rule: Select a task with the highest maximum affordable cost in proportion to the highest average cost.

Highest Utility Rule: Select a task which currently brings the highest utility (maximum affordable cost - current cost share).

6.2. Simulation and Parameters

For every set of parameters, we simulate agents' behavior under our scheme, the optimal scheme and the traditional scheme 1000 times, and calculate the average data for the evaluation criteria. For the traditional scheme, we simulate four experimental conditions. At every condition, all agents follow the same selection rule out of four rules listed above.

Table 1.3 summarizes the simulation parameters in the evaluation. The range of the number of tasks is 1,3 and 5. We assign the identical cost curve to all tasks such that the highest average cost is 100, the lowest is 80, and the average cost decreases by 5 in proportion to the number of agents. We only vary the average cost decreasing ratio (CDR), the ratio of 'the least number of agents which assures the lowest average cost' to 'the number of agents in a group.' CDR characterizes how steeply the average cost decreases. Figure 1.3 shows sample cost curves with CDR of 0.4 and 1.0, and 100 agents in a group. In the simulation, CDR varies among 0.2, 0.4, 0.6, 0.8 and 1.0.

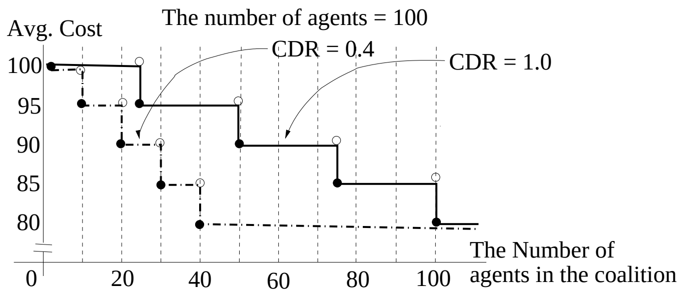

Figure 1.3. Sample cost curves

Table 1.3. Simulation Parameters

	Parameter	Range
Tasks	The number of tasks	1, 3, 5
Cost Curve	CDR (price decreasing ratio)	0.2, 0.4, 0.6, 0.8, 1.0
Agents	The number of agents	100, 200, 400, 800
	RAMT (the ratio of agents capable of multiple tasks)	(1), (1, 0, 0), (.7, .2, .1), (.5, .3, .2), (1/3, 1/3, 1/3), (1, 0, 0, 0, 0), (.7, .2, .05, .03, .02), (.5, .3, .1, .05, .05), (.2, .2, .2, .2, .2)
	RMH (the ratio of MACs which are no less than the highest average cost)	0, 0.25
	LAC (the lowest MAC)	70, 80

The range of the number of agents is 50, 100, 200 and 400. We also vary RAMT, RMH and LAC as shown in Table 1.3 so that the effect of the agents' capability and resource distributions can be observed. Note that the optimal scheme can handle only the cases with 50 agents and RAMT of (1), (1,0,0) or (0.7, 0.2, 0.1) because of its high computational complexity.

6.3. Results

For a given number of agents and tasks, the three schemes showed common relations between agents' total utilities and the simulation parameters. The factors which affected the total coalition value favorably included smaller CDR, larger RMH and LAC, and more distributed RAMT (for instance, (1/3, 1/3, 1/3) brought larger a large objective value than (1, 0, 0) did). Among them, CDR brought a clear contrast between the three schemes. Here, we analyze the simulation results focusing on CDR.

Out of the four experimental conditions for the traditional scheme, the one where all agents followed the highest utility rule produced the highest objective value in almost all simulations. Thus, in this section we refer only to this condition as the traditional scheme's output.

Optimality: First, we compare our scheme to the optimal one by examining the case that the number of tasks is 3, the number of agents is 50 and RAMT=(0.7, 0.2, 0.1). In summary, (1) our scheme came out more than 80 percent of the optimal utility under all conditions on average, and (2) as CDR became larger, the difference between our scheme and the optimal one became smaller; when CDR = 1.0, our scheme's outputs were nearly the same as the optimal ones.

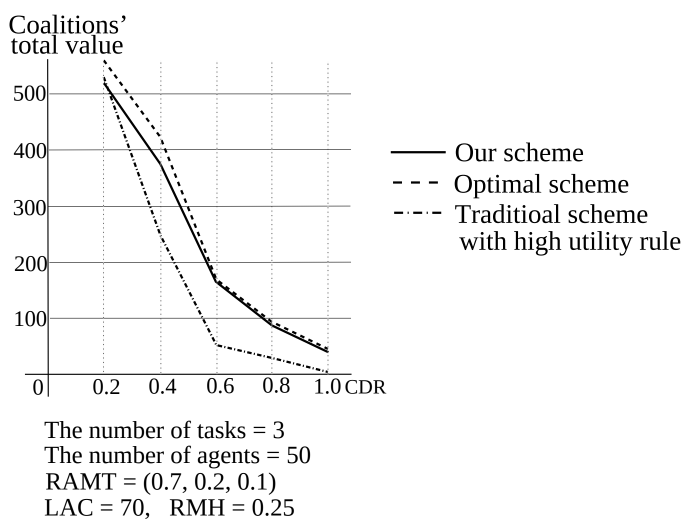

Figure 1.4. Comparison between our scheme, the optimal one and the traditional one

Figure 1.4 shows the average objective value under the conditions where LRP = 70 and RRMP = 0.25. [3] The horizontal axis is CDR, and the vertical axis is the total coalition value. When CDR is 0.2, the total utility gained by our scheme was slightly worse than the one by the optimal scheme and even the one by the traditional scheme. But, the average total utility under our scheme was still above 91 percent of the optimal one. As CDR became larger, our scheme performed better in the sense that the objective value became close to the optimal ones. When CDR ≥ 0.6, the objective value is within 96 percent of the optimal one. On the other hand, the traditional scheme became much worse when CDR was 0.4 or larger. When CDR $= 1.0$, the traditional scheme scarcely brought value to the system.

Cases with a large number of agents: Next, we examine the cases that 400 agents are involved in a group. (We compare only ours and the traditional scheme. Our implementation of optimal scheme could not handle such large number of agents.) Regardless of the number of agents, the comparison results showed the same tendency as the previous case of 50 agents: (1) when CDR=0.2, ours and the traditional scheme brought the best objective values, and the traditional scheme slightly outperformed ours under some conditions, and (2) as CDR became larger, our scheme performed better than the traditional one.

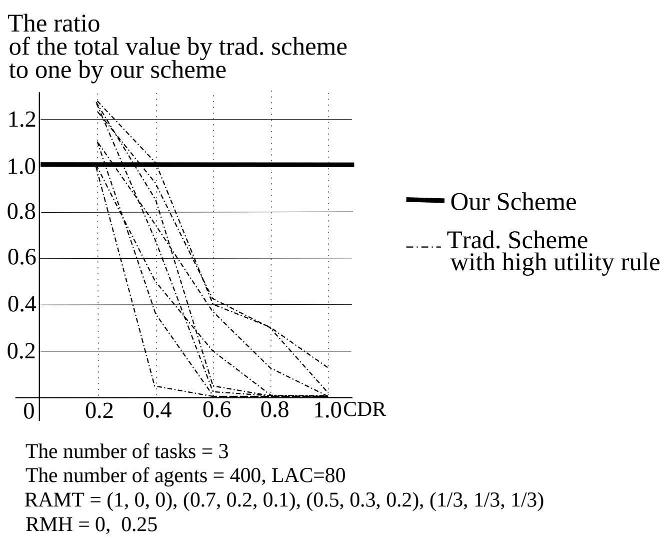

Figure 1.5. Comparison between our scheme and the traditional scheme

Figure 1.5 supports the above statements. The graph shows the ratio of the objective value by the traditional scheme to the one by our scheme. The horizontal axis of the graph is CDR. The vertical axis is the performance ratio. The value 1.0 means two schemes have the same performance, the value under 1.0 indicates our scheme is better, and the value above 1.0 does the opposite. The graph includes the data under eight conditions; RAMT = (1,0,0), (0.7, 0.2, 0.1), (0.5, 0.3, 0.2) or (1/3, 1/3, 1/3), and RMH = 0 or 0.25. Other parameters are fixed (three tasks, 400 agents, and LAC = 80). In terms of the total coalition value, the traditional scheme outperformed ours only when CDR = 0.2. When CDR ≥ 0.4, our scheme was better under all conditions.

7. Conclusions and Future Work

In this chapter, a coalition formation scheme was proposed to allocate agents to different tasks and divide the task execution cost among coalition members, considering heterogeneity of agents and tasks. We showed that our scheme has enough scalability to handle a large number of agents, guarantees the stability in cost division within each coalition, and performs better in increasing the system's performance compared to a traditional coalition formation scheme.

Future work includes to investigate strategies of agents and the mechanism design. In the evaluation reported in this paper, we simply assumed agents truthfully reveal their maximum affordable costs. Agents, however, may underreport the maximum affordable costs to share less cost in a coalition. We need to examine the relations between the mechanism design and agents' strategies to effectively solve the task allocation problem when agents are self-interested and strategic.

Notes

1. Precisely the cost of a coalition depends on the functions of the agents and the coordination mechanism. Since we consider the task allocation on a high level and do not consider the specific function allocation or scheduling of the agents, the cost of a coalition is approximated as a function of the number of participants.
2. Exhaustive search is only computationally possible for a small problem size.
3. We got similar results for other combinations of LAC (70 or 80) and RMH (0 or 0.25).

References

[Arkin and Hassin, 1999] Arkin, E. M. and Hassin, R. (1999). On local search for weighted k-set packing. In *Proceedings of the 7th Annual Europe Symposium on Algorithms.*

[Chandra and Halldorsson, 1999] Chandra, B. and Halldorsson, M. M. (1999). Greedy local improvement and weighted set packing approximation. In *Proceedings of the 10th Annual SIAM-ACM Symposium on Discrete Algorithms (SODA).*

[Lerman and Shehory, 2000] Lerman, K. and Shehory, O. (2000). Coalition formation for large-scale electronic markets. In *Proceedings of the 4th International Conference on Multiagent Systems (ICMAS-2000)*.

[Moulin, 1988] Moulin, H. (1988). *Axioms of cooperative decision making*. Cambridge University Press.

[Moulin, 1995] Moulin, H. (1995). *Cooperative Microeconomics: A Game-Theoretic Introduction*. Princeton University Press.

[Osborne and Rubinstein, 1994] Osborne, M. J. and Rubinstein, A. (1994). *A Course in Game Theory*. MIT Press.

[Sandholm et al., 1999] Sandholm, T., Larson, K., Andersson, M., Shehory, O., and Tohme, F. (1999). Coalition structure generation with worst case guarantees. *Artificial Intelligence*, 111(1-2):209–238.

[Shehory and Kraus, 1996] Shehory, O. and Kraus, S. (1996). Formation of overlapping coalitions for precedence-ordered task-execution among autonomous agents. In *Proceedings of the 2nd International Conference on Multiagent Systems (ICMAS-96)*.

[Shehory and Kraus, 1999] Shehory, O. and Kraus, S. (1999). Feasible formation of coalitions among autonomous agents in non-superadditive environments. *Computational Intelligence*, 15(3):218–251.

[Shehory et al., 1997] Shehory, O., Sycara, K., and Jha, S. (1997). Multi-agent coordination through coalition formation. In Rao, A., Singh, M., and Wooldridge, M., editors, *Intelligent Agents IV (Lecture Notes in Artificial Intelligence no. 1365)*. Springer-Verlag.

Appendix: Proof of Propositions

Proof of Prop.2. Suppose $\exists a_k \not\in C_i^*, \exists a_h \in C_i^*$ such that $r_{ki} > r_{hi}$. From the definition of v_i, $v_i(C_i^* \cup \{a_k\} \backslash \{a_h\}) > v_i(C_i^*)$ holds, which contradicts the definition of C_i^* ($v_i(C_i^*)$ be the largest).

Lemma 1. For $\forall C \subset B$ and $\forall a_k \not\in C$, if $h_C \leq r_{ki}$ then (1) $h_{C \cup \{a_k\}} \leq h_C$, and (2) $a_k \in \overline{C \cup \{a_k\}}$, where h_X and $\overline{X}$ for any X are calculated as a $t_i$coalition.

Proof of Lemma 1 (1). Suppose $h_{C \cup \{a_k\}} > h_C$, and we will show it leads to a contradiction, $cost_i(h_{C \cup \{a_k\}}) < cost_i(h_{C \cup \{a_k\}})$.

Let $D \stackrel{\text{def}}{=} C \cup \{a_k\}$. Then we have

$$
\begin{aligned}
& cost_i(D) \stackrel{\text{def}}{=} sum_{a_h \in C_i \backslash \overline{C_i}} r_{hi} + |\overline{D}| \cdot h_D \\
&= \sum \{r_{hi} \mid a_h \in D, r_{hi} < h_D\} + |\overline{D}| \cdot h_D \\
&> \sum \{r_{hi} \mid a_h \in D, r_{hi} < h_D\} + |\overline{D}| \cdot h_C \qquad (\text{ since } h_C < h_D) \\
&= \sum \{r_{hi} \mid a_h \in D, r_{hi} < h_D\} \\
&\quad + \sum \{r_{hi} \mid a_h \in D, h_C \leq r_{hi} < h_D\} + |\overline{D}| \cdot h_C \\
&= \sum \{r_{hi} \mid a_h \in C, r_{hi} < h_C\} + \sum \{r_{hi} \mid a_h \in D, h_C \leq r_{hi} < h_D\}
\end{aligned}
$$

$+ |\{a_h \in D \mid \{h_D \le r_{hi}\}| \cdot h_C$ (by $h_C \le r_{ki}$ and Def. of $\overline{D}$)
$\ge \sum\{r_{hi} \mid a_h \in C, r_{hi} < h_C\} + |\{a_h \in D \mid h_C \le r_{hi}\}| \cdot h_C$
$\ge \sum\{r_{hi} < h_C \mid a_h \in C\} + (|\overline{C}| + 1)h_C$
$= cost_i(C) + h_C$ (from Def. of $\overline{C}$)
$\ge |C| \cdot p_i(|C|) + p_i(|C|)$ (from Def. of $cost_i$ and $p_i(|C|) \le h_C$)
$\ge |C| \cdot p_i(|D|) + p_i(|D|) = |D| \cdot p_i(|D|)$ (since $|D| = |C| + 1$)
$= cost_i(D)$.

Proof of Lemma 1 (2). From $h_C \le r_{ki}$ and $(1) h_D \le h_C$, we have $h_D \le r_{ki}$, which means $a_k \in \overline{D} = \overline{C \cup \{a_k\}}$.

Lemma 2 (A general form of Lemma 1). For $\forall C \subset B$ and $\forall D \subset \{a_k \in B \mid h_C \le r_{ki}\}$, $(1) h_{C\cup D} \le h_C$, and $(2) D \subset \overline{C \cup D}$, where h_X and $\overline{X}$ for any X are calculated as a $t_i$coalition.

Proof of Lemma 2. The proof of Lemma 2 (1) is by induction on the cardinality of D. Begin with the first step. When $|D| = \{a_k\}$ and $a_k \in C$, (1) is trivial. If $a_k \notin C$, (1) is supported directly by Lemma 1. For the inductive step, suppose (1) holds for all D such that $|D| \le n$, and we will show that (1) holds for $D \cup \{a_k\}$ where$a_k \notin D$ and $h_C \le r_{ki}$. By the induction hypothesis, we have $h_{C\cup D} \le h_C \le r_{ki}$. In the case $a_k \notin C$, the above inequation and $a_k \notin C \cup D$ lead $h_{C\cup D\cup\{a_k\}} \le h_{C\cup D}$ by using Lemma 1 (1). In the case $a_k \in C$, $h_{C\cup D\cup\{a_k\}} \le h_{C\cup D}$ also holds since $C \cup D \cup \{a_k\} = C \cup D$. Using the induction hypothesis again, we have $h_{C\cup D\cup\{a_k\}} \le h_C$. (2) follows trivial by (1).

Proof of Prop.3. Suppose $\exists C \in AC_i$ s.t. $h_C < h_{C_i^*}$ ... (1). By applying $\overline{C_i^*}$ to D in Lemma 2, we have $h_{C\cup\overline{C_i^*}} \le h_C$ ... (2), and $\overline{C_i^*} \subset \overline{C \cup \overline{C_i^*}}$ ... (3). Using (1), (2) and (3), we see $v_i(C \cup \overline{C_i^*}) > C_i^*$ as follows, which contradicts that C_i^*be the largest by its definition.
$v_i(C \cup \overline{C_i^*}) = \sum_{a_k \in \overline{C\cup\overline{C_i^*}}} (r_{ki} - h_{C\cup\overline{C_i^*}})$
$> \sum_{a_k \in \overline{C\cup\overline{C_i^*}}} (r_{ki} - h_{C_i^*})$ (by combining (1) and (2))
$\ge \sum_{a_k \in \overline{C_i^*}} (r_{ki} - h_{C_i^*}) = v_i(C_i^*)$ (from (3)).

Lemma 3. For any coalition C_i and any subset $S \subset C_i$, $cost_i(S) \ge |S \cap \overline{C_i}| \cdot h_{C_i} + \sum_{a_k \in S\setminus\overline{C_i}} r_{ki}$.

Proof of Lemma 3. By Prop. 3, $h_{C_i} \le h_S$...(1) holds. Then, the following two equations are straightforwardly proved using (1): $\overline{S} = \overline{S} \cap \overline{C_i}$, and $(S\setminus\overline{S})\setminus\overline{C_i} = S\setminus\overline{C_i}$. Therefore,
$cost_i(S) = |\overline{S}| \cdot h_S + \sum_{a_k \in S\setminus\overline{S}} r_{ki}$
$= |\overline{S}| \cdot h_S + \sum_{a_k \in (S\setminus\overline{S})\cap\overline{C_i}} r_{ki} + \sum_{a_k \in (S\setminus\overline{S})\setminus\overline{C_i}} r_{ki}$
$\ge |\overline{S}| \cdot h_{C_i} + \sum_{a_k \in (S\setminus\overline{S})\cap\overline{C_i}} h_{C_i} + \sum_{a_k \in (S\setminus\overline{S})\setminus\overline{C_i}} r_{ki}$
$= |\overline{S} \cap \overline{C_i}| \cdot h_{C_i} + |(S\setminus\overline{S}) \cap \overline{C_i}| \cdot h_{C_i} + \sum_{a_k \in (S\setminus\overline{S})\setminus\overline{C_i}} r_{ki}$
$= |S \cap \overline{C_i}| \cdot h_{C_i} + \sum_{a_k \in (S\setminus\overline{S})\setminus\overline{C_i}} r_{ki}$
$= |S \cap \overline{C_i}| \cdot h_{C_i} + \sum_{a_k \in S\cap(C_i\setminus\overline{C_i})} r_{ki}$.

Proof of Prop.1. By Lemma 3 and the definition of group utility $v_i(S) \stackrel{\text{def}}{=} \sum_{a_k \in S} r_{ki} - cost_i(S)$, we have
$\sum_{a_k \in S} r_{ki} - v_i(S) \ge |S \cap \overline{C_i}| \cdot h_{C_i} + \sum_{a_k \in S\setminus\overline{C_i}} r_{ki}$.
Using Definition 2, this inequation yields
$v_i(S) \le \sum_{a_k \in S} r_{ki} - \sum_{a_k \in S\setminus\overline{C_i}} r_{ki} - |S \cap \overline{C_i}| \cdot h_{C_i}$

$= \sum_{a_k \in S \cap \overline{C_i}} r_{ki} - |S \cap \overline{C_i}| \cdot h_{C_i}$
$= \sum_{a_k \in S \cap \overline{C_i}} (r_{ki} - h_{C_i}) = \sum_{a_k \in S} x_k.$