中央研究院
資訊科學研究所
Institute of Information Science, Academia Sinica • Taipei, Taiwan, ROC

TR-IIS-05-020

Coalition Formation for Resource Co-allocation Using BDI Assignment Agents

Kiam Tian Seow, Kwang Mong Sim and Yuan Chia Kwek

December 2005 || Technical Report No. TR-IIS-05-020
http://www.iis.sinica.edu.tw/LIB/TechReport/tr2005/tr05.html

Coalition Formation for Resource Co-allocation Using BDI Assignment Agents

Kiam Tian Seow, Kwang Mong Sim and Yuan Chia Kwek

***Abstract*— A new distributed agent algorithm for resource co-allocation to different tasks is proposed. The algorithm extends a BDI assignment algorithm with resource capability reasoning. It enables resource agents to form coalitions via iterative BDI reasoning and negotiation given the limited capabilities of the resources vis-à-vis task requirements, without directly limiting the coalition size. In the worst case analysis, the number of negotiation rounds required by the algorithm is shown to be of a polynomial order in the number of agents. Empirical evidence from simulations shows that the algorithm yields favorable results in terms of the number of effective coalitions formed for different tasks. Fundamental differences between the proposed algorithm and related work are also discussed.**

***Index Terms*— Multiagent Systems, Software Agents, Resource Co-allocation, Problem Solving, Planning**

I. Introduction

Central to many real world applications in a non-centralized environment is the fundamental problem of assigning tasks to resources. The problem becomes even more challenging in many practical settings where, instead of just assigning every task to a different resource [1], some tasks need to be co-allocated with more than one different resource, under different constraints of resource (service) capability and task requirement.

Perhaps the most basic is the linear (sum) assignment problem (LAP) which deals with the question of how to *concurrently* assign N distinct tasks to N distinct resources on a one-to-one basis, with maximizing a summation objective function as the optimal *goal*. LAP manifests itself in a diverse range of interesting applications, either as a resource allocation problem or a subproblem of resource co-allocation, in personnel management, vehicle transportation, manufacturing and telecommunication, for which centralized algorithms have been applied [2].

Our research aims to develop techniques to address distributed versions of these LAP (or LAP-based) applications, emerged to exploit recent advancement in computer and internet technology that has made it possible to have situated agents collaboratively plan the assignments *by* themselves. This is in contrast to a centralized algorithm planning *for* them. Solved this way, the basic problem is termed a collaborative LAP (CLAP). While the centralized approach was acceptable in the past, it limited active involvement of distributed agents in incremental planning or problem solving.

Our recent work [1] has developed a BDI negotiation algorithm MA^3 to address CLAP. Using distributed agent reasoning, agents representing different resources negotiate for different tasks[1], to optimize the sum-total of the A-QoS (application quality-of-service) of one resource for one task. This paper addresses a more general CLAP (G-CLAP) that relaxes the allocation of strictly '*one resource to one task*' to '*several resources to one task*', motivating the idea of *coalition formation* [3]. A coalition is a team of resource agents formed to jointly service a task. G-CLAP seeks an efficient allocation of resources (e.g., limited manpower in a company or sensors of different capabilities in the same network) through coalition formation, to allocate resource coalitions, each to *effectively* handle a different task. The problem demands cooperation among the resource agents involved. The main benefit of coalition formation is either to accomplish tasks not possible with resource agents acting alone, or to achieve effective handling of tasks and a resulting increase in resource efficiency with the agents working on teams, for different tasks.

In addressing G-CLAP as a distributed agent problem to realize the benefit of coalition formation, an agent faces the basic issue of deciding whom to team up with for which tasks, given its resource capability vis-à-vis task requirements. In so doing, each resource agent *not only* needs to reason communicatively about its beliefs and preferences (in terms of desires and intentions) as well as its collaborating agents', *but also* locally reason about its A-QoS to offer for each task. The agent's A-QoS or capability reasoning entails a local model that would eventually decrease its A-QoS with successive task commitments, and depends on various other parametric constraints that characterize its resource capability. Such constraints include maximum task loading and working relationship with other agents for different tasks.

To develop such agents for G-CLAP, the key contribution of the paper includes extending MA^3 (Section II) to $\mathcal{C}$-MA^3 (Section III) that inherits the BDI negotiation model in MA^3 for communicative reasoning, and incorporating a local (capability) reasoning model (Section III-A.3) to compute successive A-QoS offers. $\mathcal{C}$-MA^3 will find application in numerous online planning domains, including a sensor network application, where targets to be concurrently monitored (the *tasks*) may each require different sensors (the *resources*) of limited capa-

K.T. Seow is with the Division of Computing Systems, School of Computer Engineering, Nanyang Technological University, Republic of Singapore 639798. Email: asktseow@ntu.edu.sg

K.M. Sim was with the Institute of Information Science, Academia Sinica, Taipei, Taiwan (until August 2005). Currently, he is with the Department of Computer Science, Hong Kong Baptist University, Kowloon Tong, Kowloon, Hong Kong. Email: bsim@comp.hkbu.edu.hk

Y.C. Kwek is currently a sales consultant with an engineering company in Singapore. Email: shanekwek@yahoo.com.sg

[1] In the original MA^3, task agents negotiate for different resources instead.

bility which diminishes with increasing task commitments. For instance, where three high value targets have been singled out for concurrent monitoring, it is desirable that a higher level command could check with three specific sensor agents, $\mathcal{C}$-MA^3-enabled to negotiate and report if their coalitions formed could effectively monitor the targets. Importantly, $\mathcal{C}$-MA^3 is found to achieve, empirically, a high success rate in terms of the number of effective coalitions formed for different tasks (Section IV), and in the worst case, does so in a number of negotiation rounds of a polynomial order in the number of agents, as analytically established (Section III-D). Fundamental differences that distinguish the proposed $\mathcal{C}$-MA^3 from related work are also examined (Sections V and VI).

II. CLAP & Solution MA^3: A Review

This section presents CLAP and the relevant details of its solution MA^3 [1].

A. Problem Statement

Let $T = \{t_0, t_1, \cdots, t_{|T|-1}\}$ and $R = \{r_0, r_1, \cdots, r_{|R|-1}\}$ denote a set of tasks and resources respectively; and $d_{ij} = d[t_i, r_j]$, for $t_i \in T, r_j \in R$ be a measure of the application quality of service (A-QoS) that a resource $r_j \in R$ can offer to a task $t_i \in T$ upon allocation. Assume $|T| \leq |R|$. Then formally, the objective of the $|T| \times |R|$ LAP is to find the particular (total) assignment mapping $\Pi : T \rightarrow R$ such that for $t_i, t_j \in T$, $i \neq j$ implies $\Pi(t_i) \neq \Pi(t_j)$, and the total A-QoS $S_{tot} = \sum_{i=0}^{|T|-1} d[t_i, \Pi(t_i)]$ is maximized over all possible permutations of Π. Intuitively, Π is a one-to-one mapping of tasks to resources. Each permutation represents an assignment (or allocation) set.

The basic approach [1] is to have each task agent represent (i.e., assume the responsibility of selecting a resource $r \in R$ for) a task $t_i \in T$ and initially possess A-QoS knowledge of task $t_i \in T$ only, namely, the individual A-QoS values $d[t_i, r]$ for all resources $r \in R$.

The objective of LAP thus becomes the joint (social) goal of these task agents, and the resulting problem is called a collaborative LAP (CLAP). In attempting to reach an optimal assignment solution, the basic issue a task agent faces is reasoning what resource exchange option to propose, as embodied in MA^3.

B. BDI Negotiation Algorithm MA^3: A Model Overview

Among the agent architectures/models (see [4, Ch. 1]), the BDI model [5], [6] is one of the best known and studied model of practical reasoning. Based on a philosophical model of human practical reasoning, originally developed by M. Bratman [7], the basic model guides us to develop an agent to decide moment by moment which action to take in the furtherance of a goal. We adapt this model, motivated by its appropriateness in allowing us to conceptualize and metaphorically describe an agent's reasoning mechanism, moment by moment, in terms of the agent' mental attitudes B (Belief), D (Desire) and I (Intention) to solve CLAP. However, two aspects clearly differentiate our work from existing BDI models. In the first is our approach to modelling. Existing BDI models are developed without concisely formulating the problems they attempt to solve while in our work, the BDI model is developed with a clear formulation of the problem it addresses, namely, CLAP. In the second, each *moment* is not a moment of reasoning in reaction to changes in its environment, but a negotiation round of collaborative reasoning - in fact, existing BDI models give no architectural consideration to explicitly multiagent aspects of behaviour [8] that is essential for addressing CLAP.

The proposed distributed agent algorithm MA^3 divides the agents' reasoning process into negotiation rounds, and in each round, performs negotiative means-end reasoning, where the *end* is to increase the social value, i.e., the total allocated A-QoS, using the *means* of resource exchange between two task agents. In each round, each task agent locally accesses and directly acts only on its own row of A-QoS data, and determines its *belief* set - the information or evidence that indicates all the possible options - the alternative resources - a task agent can exchange its current resource selection for, to achieve its end. Every task agent then begins *negotiating* by communicating with one another to acquire A-QoS data from any task agent whose current resource selection is in the agent's belief set. In collaborating, any such agents will respond with the required A-QoS values, using which the agent would deliberate to determine its own *desire* set - the means of exchanging its current resource selection for options that survive the deliberation (in that the social value will increase when exchanged for any of these options) with the respective agents (currently holding on to these options). As a final step in a negotiation round, the agent will select the best (local) desire - the one that offers a net exchange gain that is the highest from the agent's perspective - as its *intention*, which it would then use as the basis for a resource exchange proposal. All the agents' resource exchange intentions (or the lack thereof) would undergo arbitration to decide which two agents to proceed with the resource exchange, before negotiation is concluded, and the next round begins. MA^3 terminates when simultaneously, all task agents have no (more) intention to exchange resources.

C. BDI Concept Formalization

To formally ground the BDI concepts for CLAP, the following CLAP-specific data structures are formally defined in such a way that they can be naturally interpreted as a task agent's beliefs, desires and intentions computed in an arbitrary round of collaborative negotiation. In these definitions, the current resource selections of all agents refer to those made under an arbitrary permutation of Π.

Definition 1 (Belief Set B_i): Given that an agent $t_i \in T$'s current resource selection is $r^i \in R$. Then its (current) belief set B_i is given by

$$\mathsf{B}_i = \{r \in R \mid d[t_i, r] > d[t_i, r^i]\} \quad (1)$$

If $\mathsf{B}_i \neq \emptyset$, this means that agent $t_i \in T$ has at least one alternative resource selection $r \in \mathsf{B}_i$ that may lead to increase in total A-QoS S_{tot} (when made in exchange with an agent whose current selection is $r \in R$).

Definition 2 (Desire Set D_i): Given that an agent $t_i \in T$'s current resource selection is $r^i \in R$ and its belief set is B_i, $\mathsf{B}_i \neq \emptyset$. An arbitrary agent $t_j \in T$ whose current resource selection is $r^j \in R$ is said to accept agent $t_i \in T$'s beliefs B_i if $r^j \in \mathsf{B}_i$. To generate the desired exchange options or desires D_i, agent $t_i \in T$ broadcasts its beliefs B_i and current selection $r^i \in R$, and an arbitrary agent $t_j \in T$ who accepts the beliefs would respond with a pair of A-QoS values $d[t_j, r^j]$ and $d[t_j, r^i]$, so that for each of the $|\mathsf{B}_i|$ responses received, the corresponding resource exchange option $[(t_i, r^j), (t_j, r^i), \rho] \in \mathsf{D}_i$ (i.e., is agent $t_i \in T$'s desire) if $\rho > 0$, where ρ is defined by

$$\rho = -d[t_i, r^i] + d[t_i, r^j] - d[t_j, r^j] + d[t_j, r^i] \tag{2}$$

If $\rho > 0$, it means that there is a net exchange gain if agent $t_i \in T$ gives up its current selection $r^i \in R$ and selects $r^j \in R$, and in exchange, agent $t_j \in T$ gives up its current selection $r^j \in R$ and selects $r^i \in R$. Thus, any desire $d \in \mathsf{D}_i$, when carried out, will definitely lead to an increase in total A-QoS without violating Π. Quite naturally, it provides the motivation for agent $t_i \in T$ to want to exchange its current resource selection.

Definition 3 (Intention I_i): Given that an agent $t_i \in T$'s desire set is D_i, $\mathsf{D}_i \neq \emptyset$. Then, agent $t_i \in T$'s intention I_i is given by

$$\begin{aligned} I_i = [(t_i, r^j), (t_j, r^i), \rho] \in \mathsf{D}_i, \text{ for which} \\ \rho = \max\{\rho' \mid [-, -, \rho'] \in \mathsf{D}_i \ \} \end{aligned} \tag{3}$$

Agent $t_i \in T$'s decisive stance or intention has to be I_i since it is the best exchange option that the agent can propose. It is said to have no intention if either $\mathsf{B}_i = \emptyset$ or $\mathsf{D}_i = \emptyset$.

Finally, in the role of arbitration, an intention with the highest exchange gain, i.e., one that contributes to the highest increase (in total A-QoS) if carried out, is selected from all the agents' intentions $I_i \in \mathsf{I}$ gathered.

With the above formalization, the distributed agent algorithm MA^3 may be specified; it handles the simple role of arbitration through a dedicated agent.

D. Distributed Agent Algorithm

MA^3 assumes that $|T| = |R| = N$, and consists of an arbitration agent (or arbiter) and a team of task agents, $t \in T$.

1) Algorithmic Details: The generic BDI reasoning mechanism of a task agent and the simple role of the arbitration agent in an arbitrary round of collaborative negotiation can now be described as follows:

MA^3 : Collaborative (Task) Agent

1) If agent believes that there are alternative resource selections which may lead to increase in total A-QoS, it would, based on its (local) beliefs, generate the desired exchange options or desires, from which the best option will be chosen as its intention.
2) Agent submits its intention (or the lack thereof) to the arbitration agent.
3) Concurrent with Step 1 and Step 2, it responds to any requesting task agent whose beliefs it accepts, by sending to the requesting agent the A-QoS values as required for computing the requesting agent's desire.
4) Agent changes its resource selection (and then acknowledges it), proceeds to next round of negotiation or quit, as decided by the arbitration agent.

Table (a)

	r_0	r_1	r_2
t_0	$\{14\}$	5	8
t_1	2	$\{6\}$	4
t_2	8	7	$\{3\}$

$\xrightarrow{\mathsf{MA}^3}$

Table (b)

	r_0	r_1	r_2
t_0	$\{14\}$	5	8
t_1	2	6	$\{4\}$
t_2	8	$\{7\}$	3

Fig. 1. Example to illustrate MA^3

MA^3 : Arbitration Agent

1) Agent first receives the intentions (or the lack thereof) of all the task agents.
2) If agent sees that all task agents have no intention to exchange, it terminates the negotiation by telling all task agents to quit.
3) Otherwise, it
   a) selects an intention with the highest exchange gain and instructs the two agents concerned to proceed with the resource exchange.
   b) receives acknowledgement of resource exchange made as instructed (from the two agents concerned), before telling all task agents to proceed to next round of negotiation.

It has been established that MA^3 always terminates in a finite number of negotiation rounds. Although it does not guarantee an optimal agreement on termination, it has been empirically shown to produce one within about 10% of the optimal almost all the time.

2) An Example: To illustrate the working mechanisms of the proposed MA^3, consider an example problem: Fig. 1 shows two assignment tables or matrices for the problem, in which the resource selection of each agent $t_i \in \{t_0, t_1, t_2\}$ is represented by enclosing the corresponding A-QoS value within $\{\}$.

Referring to Fig. 1, Table (a) represents a randomly selected (initial) assignment and Table (b) represents a solution assignment. The following illustrates how the solution can be obtained by collaborative negotiations.

- Round 1
  - Agent t_0 selects r_0 and agent t_1 selects r_1 (as initialized). Both agents believe that they have the best selection because their belief sets are empty, hence no desire, and therefore no intention.
  - Agent t_2 selects r_2 (as initialized) but believes that there are alternative resource selections that may increase the total A-QoS, namely resources r_0 and r_1. It therefore generates its desired exchange options as follows:
    - ∗ for exchange with agent t_0, the exchange gain is
      $$-d[t_2, r_2] + d[t_2, r_0] - d[t_0, r_0] + d[t_0, r_2]$$
      which is equal to $-3 + 8 - 14 + 8 = -1 \leq 0$.
    - ∗ for exchange with agent t_1, the exchange gain is
      $$-d[t_2, r_2] + d[t_2, r_1] - d[t_1, r_1] + d[t_1, r_2]$$
      which is equal to $-3 + 7 - 6 + 4 = 2 > 0$.
    - ∗ Hence, its only desire is $[(t_2, r_1), (t_1, r_2), 2]$ and is therefore also its intention.

- To get the required pairs of A-QoS values $\{d[t_0,r_0], d[t_0,r_2]\}$ and $\{d[t_1,r_1], d[t_1,r_2]\}$ for computing its desire set as done above, agent t_2 broadcasts its belief set $\{r_0,r_1\}$ and current selection $r_2 \in R$. The respective agents whose current resource selection is in agent t_2's belief set respond with those values. In subsequent rounds, such broadcasts and responses are deemed understood and will not be mentioned again.
- Agents t_0, t_1 and t_2 send their intentions (or the lack thereof) to the arbitration agent.
- The arbitration agent tells agent t_1 to change its resource selection to r_2 and agent t_2 to change it to r_1.
- Once both agents t_1 and t_2 inform the arbitration agent that they have changed the selections as instructed, the arbitration agent tells all agents to proceed to next round of negotiation.

- Round 2
  - Agent t_0 selects r_0 and believes that it has the best selection because its belief set is empty, hence no desire, and therefore no intention.
  - Agent t_1 selects r_2 but believes that an alternative resource selection r_1 may increase the total A-QoS. It therefore generates its desired exchange options as follows:
    - for exchange with agent t_2, the exchange gain is
    $$-d[t_1,r_2] + d[t_1,r_1] - d[t_2,r_1] + d[t_2,r_2]$$
    which is equal to $-4+6-7+3 = -2 \leq 0$.
    - Hence, it has no desire, and therefore no intention.
  - Agent t_2 selects r_1 but believes that an alternative resource selection r_0 may increase the total A-QoS. It therefore generates its desired exchange options as follows:
    - for exchange with agent t_0, the exchange gain is
    $$-d[t_2,r_1] + d[t_2,r_0] - d[t_0,r_0] + d[t_0,r_1]$$
    which is equal to $-7+8-14+5 = -8 \leq 0$.
    - Hence, it has no desire, and therefore no intention.
  - Agents t_0, t_1 and t_2 send their lack of intentions to the arbitration agent.
  - The arbitration agent tells all agents t_0, t_1 and t_2 to quit. The final resource selections of the agents yield the solution as shown in Table (b) of Fig. 1.

E. Negotiation Complexity

We conclude the review with a complexity result for MA^3 in terms of the number of negotiation rounds, and an approach to reducing communication time per negotiation round. The proof of the complexity result is presented elsewhere [9].

Theorem 1: Given an arbitrary $N \times N$ CLAP instance, the worst-case complexity of MA^3 in terms of the number of negotiation rounds is $O(N^2)$.

For conceptual clarity, the basic model has presented *full* BDI reasoning for every round of negotiation. In an actual implementation, a task agent would need to (and can easily) avoid full BDI reasoning whenever possible since the determination of especially its desires requires communication to request for the necessary A-QoS data from a number of its collaborating agents, and this could incur a lot of communication time. It turns out that if we allow the arbitration agent to announce the resource exchange intentions that were executed in a previous negotiation round, and this can be easily implemented, a task agent need not always have to do full BDI reasoning beyond the first negotiation round; only two task agents, say agents t_x and t_y, involved in the resource exchange per arbitrated intention would need to perform full BDI reasoning in the current round, say round k; and we note that the worst case complexity of computing an agent t_i's beliefs $r \in \mathsf{B}_i^k \subseteq R$ according to Eq. (1) is a linear order $O(N)$. As for each of the rest, say agent t_z, its intention would remain the same as its previous one and so no BDI reasoning is needed, if any resource exchanged in round $k-1$ is not in its beliefs; otherwise, with its beliefs unchanged, i.e., $\mathsf{B}_z^k = \mathsf{B}_z^{(k-1)}$, it would only need to carry out DI reasoning, by updating its desire set D_z^k with D_z^{k-1} (i.e., desire set determined in the previous round $k-1$) but with 1) all desires that involve a resource exchanged in round $k-1$ deleted; and 2) new desires determined in round k added. A new desire of each agent t_z is deliberated using Eq. (2) for exchange gain computation, using A-QoS data acquired not through the agent broadcasting its current beliefs as in the first round, but directly requesting the required A-QoS data from only one or both the agents t_x and t_y per executed intention, provided their current resource selections are in B_z^k. Such a negotiation is said to involve *non-redundant BDI reasoning*, which helps reduce the average communication time per round.

III. G-CLAP & Proposed Solution $\mathcal{C}$-MA^3

The characteristic constraint of exactly $1:1$ assignment in CLAP may be too stringent for many applications. Some tasks might need to be assigned to a set of resources, and more generally, these resource sets need not be disjoint (i.e., are *overlapping*), meaning some resources service more than one task. This generalizes a strictly $1:1$ assignment to $1:q$ assignment (or $p:1$ assignment), for $p,q \geq 1$.

To illustrate, consider an example with $T = \{t_0,t_1,t_2\}$ and $R = \{r_0,r_1,r_2\}$. A T-centered optimal solution could consist of the assignment set

$$\{\ (t_0,\{r_0,r_1,r_2\}),\quad (t_1,\{r_1,r_2\}),\quad (t_2,\{r_2\})\ \},$$

which contains $1:3$, $1:2$ and $1:1$ assignment elements, respectively. The resource subsets $\{r_0,r_1,r_2\}$, $\{r_1,r_2\}$ and $\{r_2\}$ are termed agent *teams* or *coalitions* for the respective tasks t_0,t_1 and t_2. Note that the allocated resource sets (between any two tasks') are not necessarily disjoint. An equivalent R-centred solution set is

$$\{\ (\{t_0\},r_0),\quad (\{t_1,t_0\},r_1),\quad (\{t_2,t_1,t_0\},r_2)\ \},$$

which contains $1:1$, $2:1$ and $3:1$ assignment elements, respectively. Equivalently, the assigned task sets are not necessarily disjoint. A collaboration problem concerned with such assignments is called a general CLAP (G-CLAP), which seeks a concurrent allocation of m different resources, $m \geq 1$, for every task to optimize the sum-total A-QoS of the concurrent allocations.

To address a version of G-CLAP, we propose a distributed agent algorithm, $\mathcal{C}$-MA^3. Essentially, it uses MA^3 iteratively, but with task set T and resource set R exchanged. In so doing, we instead have $\mathcal{C}$-MA^3 resource agents, each representing a different resource; and they negotiate for tasks to form a team of resource agents $r \in R$ for every task $t \in T$.

This more complex G-CLAP is formulated and addressed under different parametric constraints that, importantly, characterize the capabilities of the resources vis-à-vis task requirements without directly limiting the coalition size, as presented in the next section.

A. Terminology & Problem Formulation

The statement for a $|R| \times |T|$ G-CLAP can be formally specified following the definition and characterization of 1) a non-increasing A-QoS function under task loading, modelling a resource capability, 2) team forming and task commitments in ordered sets, and 3) team effectiveness in terms of task requirements and member compatibility.

1) A-QoS Function, Committed Tasks & Resource Team: Let $d[t_i, r_j, l_j] \geq 0$ denote the available A-QoS - the optimal level of effectiveness - that resource $r_j \in R$ can offer to an arbitrary task $t_i \in T$, when it is H_j^c-committed. $H_j^c = \{t_{j_0}, t_{j_1}, \cdots, t_{j_y}, \cdots, t_{j_{(l_j-1)}}\} \subseteq T$. Let variable l_j denote the number of tasks in set H_j^c, $l_j \geq 0$. Where it is necessary to explicitly indicate the number of tasks l_j in H_j^c, we also rewrite it as $H_j^c\{l_j\}$. $H_j^c\{l_j\}$ is *ordered* in that a task is selected and included as the y-th element $t_{j_y} \in H_j^c$ when it was offered an A-QoS of $d[t_{j_y}, r_j, y]$ by resource $r_j \in R$ at a stage when the resource was already assigned with y tasks. As will be made clearer later, ordering is needed to index in set H_j^c the tasks committed and added to it at successive negotiation stages when they were offered different A-QoS values by the same resource $r_j \in R$.

Let $R_i = \{r_{i_0}, r_{i_1}, \cdots, r_{i_x}, \cdots, r_{i_{|R_i|-1}}\} \subseteq R$ denote a subset allocated to task $t_i \in T$, and is said to be a resource team for task $t_i \in T$. Similarly, R_i is ordered, i.e., $r_{i_x} \in R_i$, the x-th element in R_i, offered an A-QoS of $d[t_i, r_{i_x}, x]$ for task $t_i \in T$, by which it was added to the team R_i at a negotiation stage[2] indexed by $(x+1)$.

2) Resource Compatibility & Team Effectiveness: Let resource subset $F_j \subseteq R$ denote only the members in set R that resource $r_j \in R$ can team up with, inclusive of itself, to service an arbitrary task, such that for two arbitrary sets $F_a \subseteq R$ and $F_b \subseteq R$,

$$r_a \in F_b \text{ iff } r_b \in F_a \tag{4}$$

Following (4), an arbitrary resource $r_a \in F_b \subseteq R$ is said to be a *mutual affiliate* of resource $r_b \in R$, in the sense that they are complementary resources such that the A-QoS values they offer for each task are *additive*.

Let $\gamma^i > 0$ denote the A-QoS threshold for task $t_i \in T$. It is the minimum level of A-QoS required of a team of resources to service the task. How a value like γ^i is determined is application dependent. It is important to note, as with most existing approaches, modelling specific application parameters before applying an approach is a design problem.

Definition 4 (Member Compatibility): In a team $R_i \subseteq R$, a team member $r_{i_x} \in R_i$ is said to be compatible with task $t_i \in T$ iff

$$d[t_i, r_{i_x}, x] \geq \left(\lambda_{ii_x} . \frac{\gamma^i}{m_i}\right), \text{with } \lambda_{ii_x} \leq 1 \tag{5}$$

λ_{ii_x} is said to define the compatibility factor of resource $r_{i_x} \in R$ (with respect to servicing task $t_i \in T$). $m_i \leq |R_i|$ denotes the number of mutual affiliates assembled in team R_i that offer non-zero A-QoS values to service $t_i \in T$. Henceforth, each such affiliate is called a *contributing* team member of R_i. The right-hand side of (5) defines the member-level threshold of a contributing resource $r_{i_x} \in R$ for $t_i \in T$.

Definition 5 (Team Effectiveness): An arbitrary resource team $R_i \subseteq R$ is said to be effective for task $t_i \in T$ iff all its contributing members are compatible with task $t_i \in T$ and its coalition value - the sum-total A-QoS contribution by these members - exceeds the threshold for task $t_i \in T$, i.e.,

$$d[t_i, R_i] \geq \gamma^i, \text{ where } d[t_i, R_i] = \sum_{x=0}^{|R_i|-1} d[t_i, r_{i_x}, x] \tag{6}$$

By Definition 5, we (implicitly) follow a standard assumption [10] that there are design means to transform the utility values held by different agents into common A-QoS (utility) units for interagent comparison.

Referring to right-hand side of (5), we have noted that $\left((1 - \lambda_{ii_x}) . \frac{\gamma^i}{m_i}\right)$ represents the tolerable deviation from the 'average' threshold (per required resource). Therefore, to satisfy (6), this deviation must be compensated for by the other contributing members in team R_i.

Let $n_j = |F_j|$. Then since contributing members in any team R_i must also be mutual affiliates by definition,

$$\min\{n_{i_x} \mid r_{i_x} \in R_i\} \geq m_i \tag{7}$$

Thus, that a contributing member $r_{i_x} \in R_i$ is compatible implies that

$$d[t_i, r_{i_x}, x] \geq \left(\lambda_{ii_x} . \frac{\gamma^i}{n_{i_x}}\right), \text{with } \lambda_{ii_x} \leq 1. \tag{8}$$

The right-hand side of (8) defines the *necessary* threshold of task $t_i \in T$ for resource $r_{i_x} \in R$. Not greater than that in (5), it is checked against during negotiation (as seen later in Condition 1 of Constraint Zeroize-A-QoS for the A-QoS model (9)), since m_i for a team R_i is not known *a priori* the negotiation, whereas n_j for a resource $r_j \in R$ can be made known.

3) A Model for Loaded A-QoS Function: Let L^j denote the load capacity, the limit in the number of tasks that resource $r_j \in R$ can concurrently service. A formula for an A-QoS function follows. It models (the reasoning that determines) a resource $r_j \in R$'s uncommitted available/remaining service capability for a task $t_i \in T$ when it has just been admitted

[2]This stage is the $(x+1)$-th negotiation session of a proposed solution algorithm for G-CLAP (Section III-B).

as the x-th element[3] in some team R_z (for a task $t_z \in T$), i.e., $z_x = j$ or equivalently, committed task $t_z \in T$ as its y-th element in H_j^c, i.e., $j_y = z$, and it is possible $z \neq i$. Note that (numerically) $x = y = (l_j - 1)$.

For all $t_i \in T$, $l_j \geq 1$,

$$d[t_i, r_j, l_j] = \begin{cases} 0 & \text{if Constraint Zeroize-A-QoS holds,} \\ \alpha_{ij} & \text{otherwise} \end{cases} \quad (9)$$

for which

$$\alpha_{ij} = \begin{cases} d[t_i, r_j, l_j - 1] - \triangle_{ij}[l_j^+] & \text{if } d[t_i, r_j, l_j - 1] > 0 \text{ and } t_i \neq t_{j_y} \in H_j^c \text{ and } d[t_{j_y}, r_j, l_j - 1] > 0, \\ d[t_i, r_j, l_j - 1] & \text{otherwise,} \end{cases}$$

where

- $\triangle_{ij}[l_j^+] \geq 0$, and is said to define the amount of loss in effectiveness of resource $r_j \in R$ for task $t_i \in T$, due to it committing to one more task $t_{j_y} \in T$, i.e., $H_j^c\{l_j\} = H_j^c\{l_j - 1\} \cup \{t_{j_y}\}$, $j_y \neq i$; $l_j^+ \leq l_j$ denotes the number of such committed tasks $t_{j_y} \in H_j^c\{l_j\}$ for which $d[t_{j_y}, r_j, l_j - 1] > 0$. Note that $l_j = y + 1$ since H_j^c is ordered.
- Constraint Zeroize-A-QoS is a disjunction of the following conditions:
  1) $\alpha_{ij} < \left(\lambda_{ij} . \frac{\gamma^i}{n_j}\right)$; resource $r_j \in R$ cannot meet the necessary threshold of task $t_i \in T$.
  2) $t_i = t_{j_y} \in H_j^c\{l_j\}$; resource $r_j \in R$ was allocated to $t_i \in T$.
  3) $l_j^+ = L^j$; the load capacity of resource $r_j \in R$ has been fully utilized.
  4) $d[t_i, V_i(x+1)] \geq \gamma^i$; allocated effectiveness for task $t_i \in T$ has reached or exceeded threshold γ^i, where $V_i(x+1) = \{r_{i_0}, r_{i_1}, \cdots, r_{i_x}\}$ denotes a partial team formed (for task $t_i \in T$), $x \geq 0$; $V_i(0) = \emptyset$.
  5) $r_{i_x} \in V_i(x+1)$ and $d[t_i, r_{i_x}, x] > 0$, but $r_{i_x} \notin F_j$; the contributing member $r_{i_x} \in R$ that just joined team $V_i(x+1)$ is not $r_j \in R$'s affiliate.

With (9), we see that $d[t_i, r_j, L^j] = 0 \leq d[t_i, r_j, L^j - 1] \leq \cdots \leq d[t_i, r_j, 1] \leq d[t_i, r_j, 0]$.

Definition 5 of team effectiveness and the non-increasing A-QoS reasoning model (9) imply an *additive* but not necessarily a super-additive environment[4]; the former definition suggests the bigger the coalition, the better it is, but this is constrained by the latter model which suggests the addition of agents to a task coalition is costly in the sense that doing so, these agents become subsequently less able to meet the A-QoS demands of the other tasks. Therefore expanding coalitions, especially up to a grand coalition (one that includes all the agents [12]) for a task may not be beneficial overall.

[3]This occurs immediately following the $(x+1)$-th negotiation session of the proposed algorithm for G-CLAP (Section III-B).

[4]In a super-additive condition [11], adding more agents into a coalition is always beneficial.

4) The $|R| \times |T|$ G-CLAP Statement: Assume $|T| \leq |R|$. Then formally, the objective, given A-QoS reasoning model (9) in an additive environment, is to find the particular (total) assignment mapping

$$\Pi : T \rightarrow 2^R \quad (10)$$

such that every team $\Pi(t_i)$ is effective (Definition 5) and the total A-QoS

$$S_{ctot} = \sum_{i=0}^{|T|-1} d[t_i, \Pi(t_i)] \quad (11)$$

is maximized over all possible permutations of Π.

In addressing (a softer version of) G-CLAP, unlike MA^3 for CLAP which is concerned with achieving high total A-QoS solutions, the proposed solution $\mathcal{C}$-MA^3, detailed in the next section, seeks to form as many effective resource teams $\Pi(t_i)$, $t_i \in T$ as possible and aimed towards, though not necessarily achieving, the (secondary) social goal $\max\{S_{ctot}(11)\}$. Importantly, trading the social goal for lower negotiation complexity (Section III-D) does not result in low success rate in forming effective teams (Section IV).

The original G-CLAP as formulated is a complex optimization problem. Formulating an approach to G-CLAP that also achieves its social goal (i.e., yields a globally optimal solution) is an important but challenging direction that should be pursued in future work.

B. Proposed Solution: $\mathcal{C}$-MA^3

$\mathcal{C}$-MA^3 assumes that $|T| = |R| = N$, and consists of an arbitration agent (or arbiter) and a group of r-agents, $r \in R$. A r-agent is the resource agent representing (i.e., having the responsibility of selecting tasks $t \in T$ only for) resource $r \in R$, and initially possesses only A-QoS knowledge of the resource it represents, initially $d[t, r, 0] > 0$ for all $t \in T$.

We call one complete run of the extended algorithm $\mathcal{C}$-MA^3 a *course*. A course consists of several *sessions* of negotiation, and one session involves several negotiation *rounds*. The reference data of agent $r_j \in R$ for the $(x+1)$-th session are summarized in Table I. The main idea, then, is to iteratively run each session using MA^3 on iteratively updated A-QoS values, until all A-QoS values reduce to 0.

TABLE I

AGENT $r_j \in R$'S REFERENCE DATA FOR $(x+1)^{th}$ SESSION

t_0	t_1	$\cdots$	$t_{(N-1)}$
λ_{0j}	λ_{1j}	$\cdots$	$\lambda_{(N-1)j}$
γ^0	γ^1	$\cdots$	γ^{N-1}
$d[t_0, V_o(x)]$	$d[t_1, V_1(x)]$	$\cdots$	$d[t_{N-1}, V_{N-1}(x)]$

constants: n_j, L^j

t_0	t_1	$\cdots$	$t_{(N-1)}$
$d[t_0, r_j, x]$	$d[t_1, r_j, x]$	$\cdots$	$d[t_{(N-1)}, r_j, x]$

1) Reasoning Mechanism: The purpose of negotiation is to allocate, for each task $t_i \in T$, an effective team R_i, $R_i = \Pi(t_i) \subseteq R$. In $\mathcal{C}$-MA^3, every resource agent $r_j \in R$ negotiates to form its committed task set H_j^c first, from which the teams R_i can be easily determined if desired. Every resource agent $r_j \in R$ will commit to a task $t_i \in T$ selected after each session, by adding it to its set H_j^c. As a result, after the 1^{st} session, updating the allocable A-QoS values using (9) for the 2^{nd} session yields values of $d[t_i, r_j, 1]$ for all $t_i \in T$, $r_j \in R$. Inductively, updating the allocable A-QoS values for the $(x+1)$-th session yields values of $d[t_i, r_j, x]$ for all $t_i \in T$, $r_j \in R$. In forming a team $\Pi(t_i)$ such that every resource agent needs to technically add a task to its set H_j^c after every session, non-contributing team members (allocating zero A-QoS values for task $t_i \in T$) may also be included. Only contributing members in a team R_i can service task $t_i \in T$; the rest can be removed.

A $\mathcal{C}$-MA^3 resource agent's reasoning mechanism may be presented as follows:

$\mathcal{C}$-MA^3 : Collaborative (Resource) Agent $r_j \in R$

---

1) **Initialize** according to

$$d[t_i, r_j, 0] := \begin{cases} 0 & \text{if } d[t_i, r_j, 0] < \left(\lambda_{ij} \cdot \frac{\gamma^i}{n_j}\right), \\ \text{unchanged} & \text{otherwise.} \end{cases}$$

2) Iteratively
   a) **run** a session of negotiation via MA^3;
   b) **add** task selected to committed-task set H_j^c as the last element (counting from left);
   c) **share** and **accumulate** allocated (partial team) effectiveness for every task;
   d) **update**, for every task, the A-QoS values according to (9);

   until it has set all its A-QoS values to 0, and is informed to end the course.
3) **Compute** to **check** its compatibility (Definition 4) with each committed task $t_{j_x} \in H_j^c$, and **verify** team R_{j_x}'s effectiveness (Definition 5).

Implicit in $\mathcal{C}$-MA^3 is an arbitration role which can be assumed by a dedicated agent or one of the resource agents, referred to as the arbitration agent. To start the next session, every resource agent involved will first inform the arbitration agent if it is interested to proceed; *it is not interested if it has set all its A-QoS values to* 0. The arbitration agent will inform all the resource agents to end the negotiation course when all resource agents have indicated the lack of interest to proceed. Within each session, the arbitration agent proceeds as per that in MA^3.

As a final remark, in running MA^3 iteratively, $\mathcal{C}$-MA^3 is attempting to attain the best possible allocation (i.e., one with maximum total A-QoS) of one different resource to every task in each session. In doing so, the coalition size might tend to be minimized since the cumulative team effectiveness for every task should reach or exceed the task threshold sooner.

C. An Example

To illustrate the mechanism of $\mathcal{C}$-MA^3, consider an example of a 3×3 G-CLAP, with $T = \{t_0, t_1, t_2\}$ and $R = \{r_0, r_1, r_2\}$. All resources are mutual affiliates, i.e., $R = F_0 = F_1 = F_2$, hence $n_j = 3$. The other constant values for γ^i, λ_{ij}, $\triangle_{ij}[l_j]$ and L^j, are given as follows:

Task requirements	Resource task-capability: Task-resource characteristics	Task loading
$\gamma^0 = 6$	$\lambda_{ij} = 1$	$L^0 = 2$
$\gamma^1 = 10$	$\triangle_{i0}[1] = 1.8$,	$L^1 = 2$
$\gamma^2 = 8$	$\triangle_{i1}[1] = 0.0$, $\triangle_{i2}[1] = 2.0$	$L^2 = 2$

The resource agents' data for session 1 are listed in Table II.

TABLE II
REFERENCE DATA OF AGENTS $r_j \in R$ ON ALL TASKS $t_i \in T$ FOR 1^{st} SESSION, $x = 0$

(a) Data	(b) r_0: t_0	t_1	t_2	(c) r_1: t_0	t_1	t_2	(d) r_2: t_0	t_1	t_2
λ_{ij}	1	1	1	1	1	1	1	1	1
γ^i	6	10	8	6	10	8	6	10	8
$d[t_i, V_i(0)]$	0	0	0	0	0	0	0	0	0
$n_j = 3$, $L^j = 2$, $d[t_i, r_j, 0]$	(6)	9	7*	10*	(3)	4	8	5*	(2)

TABLE III
REFERENCE DATA OF AGENTS $r_j \in R$ ON ALL TASKS $t_i \in T$ FOR 2^{nd} SESSION, $x = 1$

(a) Data	(b) r_0: t_0	t_1	t_2	(c) r_1: t_0	t_1	t_2	(d) r_2: t_0	t_1	t_2
λ_{ij}	1	1	1	1	1	1	1	1	1
γ^i	6	10	8	6	10	8	6	10	8
$d[t_i, V_i(1)]$	10	5	7	10	5	7	10	5	7
$n_j = 3$, $L^j = 2$, $d[t_i, r_j, 1]$	0	7.2*	(0)	(0)	3	4*	0*	(0)	0

TABLE IV
REFERENCE DATA OF AGENTS $r_j \in R$ ON ALL TASKS $t_i \in T$ FOR 3^{rd} SESSION, $x = 2$

(a) Data	(b) r_0: t_0	t_1	t_2	(c) r_1: t_0	t_1	t_2	(d) r_2: t_0	t_1	t_2
λ_{ij}	1	1	1	1	1	1	1	1	1
γ^i	6	10	8	6	10	8	6	10	8
$d[t_i, V_i(2)]$	10	12.2	11	10	12.2	11	10	12.2	11
$n_j = 3$, $L^j = 2$, $d[t_i, r_j, 2]$	0	(0)	0	0	0	(0)	(0)	0	0

In one operational implementation, the arbitration agent first informs all the three agents about the three tasks and inform them to get ready for negotiation. Upon A-QoS initialization, each agent would then respond to the arbitration agent that it is ready. Once the arbitration agent has received all the three responses, it would inform the three agents to proceed with Session 1.

In illustrating, each agent begins the $(x+1)$-th session with a task selection whose A-QoS value is within $()$, and when the session ends, it has a new task selection whose A-QoS value is marked '*' on the same reference data table (e.g., Table II for the 1-st session).

After Session 1, we have: $H_0^c = \{t_2\}$, $H_1^c = \{t_0\}$ and $H_2^c = \{t_1\}$. The agents share and accumulate the allocated effectiveness values for every task, as reflected in Table III: $d[t_i, V_i(1)]$. Each agent then computes and updates the rest of the data accordingly for Session 2, as shown in Table III. Each agent then informs the arbitration agent if it is interested to proceed with Session 2; it is not interested if it has set all its A-QoS values set to to 0. Because at least one agent is interested, the arbitration agent informs the three agents to proceed with Session 2.

After Session 2, we have: $H_0^c = \{t_2, t_1\}$, $H_1^c = \{t_0, t_2\}$ and $H_2^c = \{t_1, t_0^?\}$. For easy identification in this illustration, we tag any task $t_{j_x} \in T$ in set H_j^c that has $d[t_{j_x}, r_j, x] = 0$ with a superscript '?', to indicate that resource $r_j \in R$ is non-contributing for task $t_{j_x} \in T$. As usual, the agents share and accumulate the allocated effectiveness values for every task, and then compute and update the data accordingly for Session 3, as shown in Table IV. Each agent then informs the arbitration agent if it is interested to proceed with Session 3. Because all agents are now not interested, the arbitration agent informs them to end the negotiation course.

For this negotiation course, after removing the tasks that will not be serviced (i.e. those superscripted with '?' in sets H_j^c), the task selections of the three resource agents are as follows: For agent $r_0 \in R$, the committed task set is $\{t_2, t_1\}$, for agent $r_1 \in R$, it is $\{t_0, t_2\}$, and for agent $r_2 \in R$, it is $\{t_1\}$. For this example, the overall assignment solution is found to be optimal, with a total A-QoS of $(7 + (9 - 1.8)) + (10 + (4 - 0)) + (5)$ = 33.2. All teams are effective because in team R_0, $d[t_0, \{r_1\}] = d[t_0, r_1, 0] = 10 \geq \gamma^0$; in team R_1, $d[t_1, \{r_2, r_0\}] = d[t_1, r_2, 0] + d[t_1, r_0, 2] = 5 + (9 - 1.8) = 12.2 \geq \gamma^1$, and in team R_2, $d[t_2, \{r_0, r_1\}] = d[t_2, r_0, 0] + d[t_2, r_1, 1] = 7 + (4-0) = 11 \geq \gamma^2$, respectively with $m_0 = 1$, $d[t_0, r_1, 0] = 10 \geq \left(\lambda_{01}.\frac{\gamma^0}{m_0}\right)$, $m_1 = 2$, $d[t_1, r_2, 0] = 5 \geq \left(\lambda_{12}.\frac{\gamma^1}{m_1}\right)$, $d[t_1, r_0, 1] = (9 - 1.8) \geq \left(\lambda_{10}.\frac{\gamma^1}{m_1}\right)$, and $m_2 = 2$, $d[t_2, r_0, 0] = 7 \geq \left(\lambda_{20}.\frac{\gamma^2}{m_2}\right)$, $d[t_2, r_1, 1] = (4 - 0) \geq \left(\lambda_{21}.\frac{\gamma^2}{m_2}\right)$.

D. Worst-case Complexity of Coalition Formation

This section presents a complexity result for $\mathcal{C}$-MA^3 in terms of the number of negotiation rounds. Proof of the result requires the following lemma.

Lemma 1: Given an arbitrary $N \times N$ G-CLAP instance, the maximum number of sessions required by $\mathcal{C}$-MA^3 is of the order $O(N)$.

Proof: In the worst case, each A-QoS value of a resource for a task is *non-zero* initially, and in every session, for every resource agent $r_j \in R$, only the A-QoS value of the task $t_i \in T$ selected following every session is *zeroized* (i.e., only Condition 2 of Constraint Zeroize-A-QoS is satisfied). So the (minimum) number of A-QoS values *zeroized* after the first session is N.

If N is even, all N agents will make mutual exchanges at least once for *non-zero* A-QoS values in every subsequent session. It follows that after k sessions, $k \geq 1$, there are k zeroized A-QoS values, with respect to both a resource agent and a task. Thus, the (minimum) number of non-zero A-QoS values that are *zeroized* per session is N. Since the algorithm will terminate when all N^2 A-QoS values have been reduced to 0, the maximum number of sessions required is N.

If N is odd, say $2r + 1$, $r \geq 1$, then after the first session, the (minimum) number of A-QoS values *zeroized* is $2r$ per session for the next $(2r-1)$ sessions. Thus, after $2r$ or $(N-1)$ sessions, a minimum of $N+2r(2r-1)$ or $N+(N-1)(N-2)$ A-QoS values would have been zeroized. Subsequently, it is 2 per session until all N^2 A-QoS values have been *zeroized*, giving an addition of $\frac{N^2-[N+(N-1)(N-2)]}{2}$ or $(N - 1)$ more sessions. This leads to a maximum total of $2(N-1)$ sessions. Hence the result. ■

Theorem 2: Given an arbitrary $N \times N$ G-CLAP instance, the worst-case complexity of $\mathcal{C}$-MA^3 in terms of the number of negotiation rounds is $O(N^3)$.

Proof: The complexity of $\mathcal{C}$-MA^3 is determined by the core computation of iterating MA^3 in sessions. Since, from Theorem 1, the worst-case complexity of MA^3 is $O(N^2)$ in terms of the number of negotiation rounds, and by Lemma 1, the number of sessions required by $\mathcal{C}$-MA^3 is bounded by $O(N)$, it follows that the worst-case complexity of $\mathcal{C}$-MA^3 is $O(N^3)$. Hence the result. ■

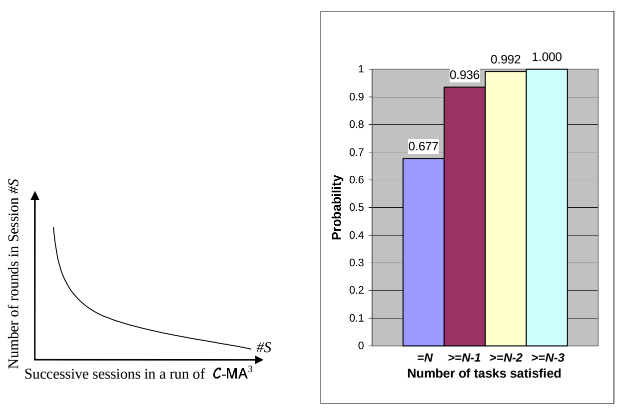
IV. $\mathcal{C}$-MA^3 Simulations and Discussion

To conduct a performance evaluation, we first prototyped a simulator for algorithm $\mathcal{C}$-MA^3. The simulator consists of a centralized program running on a personal computer. For an $N \times N$ problem instance, the program generates and inputs each of the $N!$ initial assignment solutions to a reasoning mechanism which computes the agents' task selections which would have resulted from the distributed agent algorithm $\mathcal{C}$-MA^3.

In principle, $\mathcal{C}$-MA^3 can handle an arbitrary problem size N. But for a complete unbiased simulation, the number of simulation runs or courses carried out is $N!$ per problem instance. Clearly for a big N, it can become intractably time consuming to simulate for a large number of problem instances. For experimental purposes, we limit $N = 6$, requiring 6! (or 720) simulation runs per problem instance. Despite this limit, we note that the simulation results also provide a practical reference for addressing larger problems that can be decomposed into smaller subproblems for $\mathcal{C}$-MA^3. Problem decomposition, however, is usually done based on application-specific criteria; this is beyond the scope of this paper.

The problem instances were generated under the following settings: For all $t_i \in T$, $r_j \in R$, $F_j = R$ (hence $n_j = N$), $\lambda_{ij} = 1$, $L^j = N$, and $\triangle_{ij}[l_j^+] = 0$; $d[t_i, r_j, 0]$ and γ^i were randomly generated. A problem is said to be 'solvable' if there is at least one initial assignment (in Session 1) out of $N!$ on which the simulator arrives at an assignment solution in which all teams are effective. We ran the simulations until 100 solvable problem instances were generated and simulated. Only the simulation results for solvable instances were recorded.

The simulation results show that the maximum number of sessions per run was about $4 < N$ and the maximum number of rounds per course was $21 << N^3$. In terms of the number of negotiation rounds, MA^3 has a worst-case complexity of $O(N^2)$ (Theorem 1). However, running as a session in $\mathcal{C}$-MA^3, it is empirically lower since, due to A-QoS zeroization, the number of negotiation rounds decreased (or the negotiation speed increased) in each successive session This is depicted in Fig. 2(a). All these suggest that practically, $\mathcal{C}$-MA^3 might frequently form coalitions in a number of rounds of a lower order than its worst-case analytical bound of $O(N^3)$ (Theorem 2).



(a) Number of negotiation rounds in successive Session $\#S$ of a run

(b) Probability of a run converging with a specific number of tasks effectively satisfied, $N = 6$

Fig. 2. $\mathcal{C}$-MA^3 simulations: Some empirical (average) results for 100 randomly generated 6×6 solvable problem instances

The increase in speed in successive sessions means that if $\mathcal{C}$-MA^3 were to be halted in the midst, the leverage on this feature will be lost. However, it does not imply that the algorithm is not anytime ready; a solution (a team for each task) is still anytime available whenever $\mathcal{C}$-MA^3 needs to be halted before it would have normally terminated. But as with any anytime ready algorithm, high solution quality cannot be guaranteed whenever it is halted this way; herein unlike that for MA^3 which is solely concerned with efficiency in resource utilization, *solution quality* includes effectiveness in handling tasks, i.e., is also defined in terms of the number of tasks effectively satisfied by the resource teams formed.

It is also found that the average number of tasks satisfied was $0.9N$. A general observation is that the larger the number of sessions required, the larger the number of tasks found to be effectively satisfied. Finally, in Fig. 2(b), we present, graphically, the average probabilities obtained of $\mathcal{C}$-MA^3 converging with N, at least $(N-1)$, $(N-2)$, and $(N-3)$ tasks satisfied. It can be inferred that for $N = 6$, a solvable problem instance is almost guaranteed (99.2%) to have at least $(N-2)$ tasks satisfied.

V. Related Work

$\mathcal{C}$-MA^3 provides an anytime ready solution for G-CLAP, and is simple and easy to implement. It is perhaps the only attempt that uses the solution concept of BDI assignment in negotiation for coalition formation. A unique feature is that as negotiation progresses, the $\mathcal{C}$-MA^3 resource agents, following every session, check for satisficing solutions, i.e., whether the coalitions being formed are effective enough for each task, while also reasoning to update their capability status according to their A-QoS models (9).

1) Coalition Formation: Under different problem scenarios, coalitions can be formed, either by *centralized* reasoning *for* self-interested agents [3], [13], [14] within a game-theoretic framework, or by *distributed* reasoning *among* agents within a multiagent framework which can be cooperative [15] or non-cooperative [16]. The coalitions formed are either disjoint [13], [14] or overlapping [15].

In [17], [15], coalition formation has been utilized as a means for cooperative agents to coordinate the order of execution of sub-tasks, so as to maximize some system utility outcome (reminiscent of the total A-QoS in our context), but not necessarily reducing the execution time. In our work, it is utilized as a means for cooperative (resource) agents to have their resources effectively co-allocated to handle independent tasks concurrently (i.e., tasks that need to be done in the same time period). Each resource agent in $\mathcal{C}$-MA^3 has a task loading limit - a necessary and intuitive characterization of resource capability - without directly limiting the number of agents on a team as imposed in [15] to reduce algorithmic computations. Unlike with $\mathcal{C}$-MA^3, the optimal solution that may be attained by the latter given a limit on agent team size can be arbitrarily far from the actual optimal solution (without the limit), as noted in [14].

Other related work has focussed on coalition formation in competitive, game-theoretic environments (e.g., among different companies or self-interested individuals). In this setting [16], [14], coalition formation involves three activities: 1) coalition structure generation (grouping the agents into coalitions), 2) solving each coalition's (optimization) problem within the coalition, and 3) dividing the value of each coalition among member agents. In ours, it involves, roughly speaking, interleaving the first two activities among the $\mathcal{C}$-MA^3 resource agents to form agent coalitions during problem solving, concurrent with a different activity 3) which checks, as a necessity for team effectiveness, each cumulative coalition value - the sum-total A-QoS contribution by all contributing members in a coalition as it is being formed - against a (minimum) task requirement. Activity 3) helps determine if the coalition formation is successful, and is completed after coalition members' compatibility with assigned tasks and the effectiveness of their fully formed coalitions have been verified. The existing approach differs fundamentally from ours on activity 3), since coalition formation is by *self-interested* agents concerned with how gains from cooperation are to be distributed [16], while ours is by cooperative task-centric resource agents concerned with whether the coalitions formed provide enough A-QoS's (the gains) to satisfy task requirements.

The proposed $\mathcal{C}$-MA^3 complements related research work with *overlapping* coalition formation by *distributed* agent reasoning in a *cooperative* multiagent framework for a different resource co-allocation problem, G-CLAP.

2) Distributed Constraint Reasoning: There are some efforts not cast in the context of coalition formation but appear to have addressed a similar problem in the context of *distributed constraint reasoning*, notably, work on distributed constraint optimization problem (DCOP) (e.g., [18], [19]). In principle, CLAP and G-CLAP are DCOPs. However, being perhaps overly general, existing DCOP techniques seek to *indirectly* minimize the cost values associated with satisfying constraints between agents, not *directly* the values associating the variables and their individual domain values. In reformulating CLAP as a DCOP, T becomes a set of 'variables', R becomes a finite discrete domain for every variable $t_i \in T$, but the function to maximize[5] becomes a rather unintuitive one given by $\sum_{i=0}^{N-1} f_i(A)$, where $A = \{(t,r) \mid t \in T, r \in R\}$ is a solution set and $f_i(A)$ is a (*valued* constraint) function of $(N-1)$-ary constraint, $(t_i \neq t_k)\ \forall t_k \in T, k \neq i$, that when satisfied with $t_i = r$, returns the element $d[t_i, r]$ of the standard CLAP formulation.

In any case, current DCOP techniques are still largely *algorithmic*; they do not exploit important agent concepts such as BDI, agent coalition and capability reasoning to provide conceptually clearer agent-based solutions that aid in the better understanding of fundamental application problems. DCOP techniques also do not lend themselves readily to incremental problem solving, so their solutions are not guaranteed to be anytime ready, unlike MA^3 and $\mathcal{C}$-MA^3.

3) Contracting: In [20], the idea of *contracting* to address a task allocation problem [20] is proposed. The problem definition [20, p. 68, Definition 1] is similar to G-CLAP but is also not cast in the context of coalition formation. More importantly, the individual agents considered therein are self-interested, and each deals with a cost function of handling subsets of tasks. This contrasts with $\mathcal{C}$-MA^3 resource agents which are cooperative, and informed with a resource capability model (9). This model explicitly characterizes the non-increasing capability level of a collaborative agent for uncommitted tasks, as negotiation progresses and the agent continually commits to more tasks. Technically, *contracting* strives to achieve an optimal task allocation whereas our approach enables resource capability reasoning through iterative BDI assignment negotiation [1] to arrive at as many effective coalitions as possible; global efficiency in resource utilization (social goal) is treated as a secondary objective.

4) Other Related Approaches: Outside the realm of coalition formation, there are several other approaches to overlapping problems on allocation of objects, be they resources, tasks or roles. For example, in [21], BDI and POMDP models are combined to address role allocation in teams. In [22], the idea of capability matching to perform allocation is proposed. In [23], [24], different ideas of auctions to decide on how tasks should be assigned or allocated are developed. As with $\mathcal{C}$-MA^3, these different techniques treat modelling an application a design issue outside their jurisdiction. In [25], a role (re)allocation algorithm is developed to enable autonomy of role reallocation to shift between a human supervisor and the agents. Other related approaches include reconfiguration methods for reforming a team [26], self-adapting organizations [27] and dynamic re-organizing groups [28].

The main idea that distinguishes from existing allocation approaches is our introduction of resource capability (reasoning) models for the formation of effective coalitions: this enables negotiation that dynamically allocates, taking into explicit account of individual capability of agents that would naturally lower as the agents commit to additional tasks in the process of their joining and forming task coalitions. The negotiation process is accelerated by Constraint Zeroize-A-QoS for the A-QoS model (9), which decrements an agent's balance capability accordingly upon its task commitment that concludes a session, up to or before its task loading limit is reached.

[5]We assume that current DCOP techniques for minimizing cost can be easily modified for maximizing utility (or A-QoS).

VI. Conclusion & Future Work

We have presented a BDI coalition formation algorithm $\mathcal{C}$-MA^3 as an approximate but effective solution of low negotiation complexity to $N \times N$ G-CLAP, a resource co-allocation problem. $\mathcal{C}$-MA^3 extends MA^3 [1] by inheriting its BDI negotiation model for communicative reasoning, and incorporating a local A-QoS (capability) reasoning model, exploited to avoid computing all possible coalitions and directly limiting the coalition size. To the best of our knowledge, no other coalition formation research has attempted to solve a resource co-allocation problem that explicitly considers resource capability reasoning.

Empirical evidence from $\mathcal{C}$-MA^3 simulations on solvable $N \times N$ G-CLAP instances shows that $\mathcal{C}$-MA^3 yields favorable results in terms of the number of effective coalitions formed for different tasks. An objective assessment shows that in the worst case, the algorithm can form coalitions in a number of negotiation rounds of a polynomial order $O(N^3)$ (Theorem 2), with empirical evidence suggesting that practically, the complexity order can frequently be lower. Future work includes gathering more insights of $\mathcal{C}$-MA^3 with varied parametric settings, conducting a probability evaluation on the various extents (e.g., within 5%, 10%) that the total A-QoS of a solution produced deviates from the optimal one (social goal) achievable, and seeking ways to speed up coalition formation and improve global efficiency. To demonstrate the practical applicability of $\mathcal{C}$-MA^3, it is also expected to include empirical studies to be carried out 1) in a context specific to an application (e.g., resource allocation in Grid computing), and 2) on a large scale.

Acknowledgements

The authors would like to thank the Associate Editor and three anonymous referees for their comments and suggestions

which helped improve the paper significantly.

K. M. Sim gratefully acknowledges financial support from the Institute of Information Science (IIS), Academia Sinica, Taiwan, ROC under a project entitled *Collaborative BDI Mechanism Design for Negotiation Agents* for funding K.T. Seow's visit at IIS from January 1, 2005 to July 14, 2005.

References

[1] K. T. Seow and K. Y. How, "Collaborative assignment : A multiagent negotiation approach using BDI concepts," in *Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'02)*. Palazzo Re Enzo, Bologna, Italy: ACM Press, July 2002, pp. 256–263.

[2] R. E. Burkard and E. Cela, "Linear assignment problems and extensions," in *Handbook of Combinatorial Optimization, Vol.4*, P. M. Pardalos and D. Z. Du, Eds. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1999, pp. 75–149.

[3] A. Rapoport and J. P. Kahan, *Theories of Coalition Formation*. Hillsdale, NJ: Lawrence Erlbaum Associates, 1984.

[4] G. Weiss, Ed., *Multiagent System : A Modern Approach to Distributed Artificial Intelligence*. The MIT Press, London, U.K, 1999.

[5] M. E. Bratman, D. J. Israel, and M. E. Pollack, "Plans and resource-bounded practical reasoning," *Computational Intelligence*, vol. 4, no. 4, pp. 349–355, 1988.

[6] M. P. Georgeff and A. L. Lansky, "Reactive reasoning and planning," in *Proceedings of the Sixth National Conference on Artificial Intelligence (AAAI'87)*, Seattle, WA, USA, July 1987, pp. 677–682.

[7] M. E. Bratman, *Intentions, Plans and Practical Reason*. Harvard University Press, Cambridge, M.A, 1987.

[8] M. P. Georgeff, B. Pell, M. E. Pollack, M. Tambe, and M. Wooldridge, "The Belief-Desire-Intention model of agency," in *Intelligent Agents V: Agent Theories, Architectures, and Languages (ATAL): Proceedings of the 5th International Workshop, ATAL'98*, J. P. Müller and A. S. R. Munindar Singh, Eds. Paris, France: Springer Verlag, London, UK, July 1998, pp. 1–10.

[9] K. T. Seow, K. M. Sim, Y. S. Ong, and E. P. Sulaiman, "A BDI Assignment Protocol with New Cooperative-Concession Strategies," Institute of Information Science, Academia Sinica, Taipei, Taiwan, Technical Report, 2005, to appear.

[10] J. S. Rosenschein and G. Zlotkin, *Rules of Encounter: Designing Conventions for Automated Negotiation among Computers*. Cambridge, M.A, USA: The MIT Press, 1994.

[11] G. Zlotkin and J. S. Rosenschein, "Coalition, cryptography, and stability: Mechanisms for coalition formation in task oriented domains," in *Proceedings of the Twelfth National Conference on Artificial Intelligence*. Menlo Park, California, USA: AAAI Press, August 1994, pp. 432–437.

[12] L. S. Shapely, "A value for N-person games," in *Contributions to the Theory of Games, No. 28 in Annals of Mathematics Studies*, H. W. Kuhn and A. W. Tucker, Eds. Princeton, NJ, USA: Princeton University Press, 1953, pp. 307– 317.

[13] T. Sandholm, K. Larson, M. Andersson, O. Shehory, and F. Tohmé, "Coalition structure generation with worst case guarantees," *Artificial Intelligence*, vol. 111, no. 1-2, pp. 209–238, 1999.

[14] V. D. Dang and N. R. Jennings, "Generating coalition structures with finite bound from the optimal guarantees," in *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'04)*. New York City, New York, USA: IEEE Computer Society, July 2004, pp. 564–571.

[15] O. Shehory and S. Kraus, "Methods for task allocation via agent coalition formation," *Artificial Intelligence*, vol. 101, no. 1-2, pp. 165–200, 1998.

[16] F. Tohmé and T. Sandholm, "Coalition formation processes with belief revision among bounded-rational self-interested agents," *Journal of Logic and Computation*, vol. 9, no. 6, pp. 793–815, December 1999.

[17] O. M. Shehory, S. P. K. Sycara, and S. Jha, "Multi-agent coordination through coalition formation," in *Intelligent Agents IV: Agent Theories, Architectures and Languages, Lecture Notes in Artificial Intelligence, Vol. 1365*, A. Rao, M. Singh, and M. Wooldridge, Eds. Providence, RI, USA: Springer Verlag, London, UK, 1997, pp. 143–154.

[18] R. Mailler and V. Lesser, "Solving distributed constraint optimization problems using cooperative mediation," in *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'04)*. New York City, New York, USA: IEEE Computer Society, July 2004, pp. 438–445.

[19] P. J. Modi, W.-M. Shen, M. Tambe, and M. Yokoo, "An asynchronous complete method for distributed constraint optimization," in *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'03)*. Melbourne, Victoria, Australia: ACM Press, July 2003, pp. 161–168.

[20] T. W. Sandholm, "Contract types for satisficing task allocation: I theoretical results," in *Proceedings of AAAI Spring Symposium Series: Satisficing Models*, Sanford University, Stanford, CA, USA, March 1998, pp. 68–75.

[21] R. Nair and M. Tambe, "Hybrid bdi-pomdp framework for multiagent teaming," *Journal of Artificial Intelligence Research*, vol. 23, pp. 367–420, 2005.

[22] G. Tidhar, A. Rao, and L. Sonenberg, "Guided team selection," in *Proceedings of the Second International Conference on Multiagent Systems (ICMAS'96)*, 1996, pp. 369–376.

[23] L. Hunsberger and B. J. Grosz, "A combinatorial auction for collaborative planning," in *Proceedings of the Fourth International Conference on Multiagent Systems (ICMAS'00)*, 2000, pp. 151–158.

[24] S. S. Fatima and M. Wooldridge, "Adaptive task and resource allocation in multi-agent systems," in *Proceedings of the Fifth International Conference on Autonomous Agents (Agents'01),*, 2001, pp. 537–544.

[25] P. Scerri, D. Pynadath, L. Johnson, P. Rosenbloom, M. Si, N. Schurr, and M. Tambe, "A prototype infrastructure for distributed robot-agent-person teams," in *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'03)*, 2003, pp. 433–440.

[26] B. Dunin-Keplicz and R. Verbrugge, "A reconfiguration algorithm for distributed problem solving," *Engineering Simulation*, vol. 18, pp. 227–246, 2001.

[27] H. Bryan, B. Brett, and L. Victor, "Using self-diagnosis to adapt organizational structures," in *Proceedings of the Fifth International Conference on Autonomous Agents (Agents'01),*, 2001, pp. 529–536.

[28] K. S. Barber and C. E. Martin, "Dynamic reorganization of decision-making groups," in *Proceedings of the Fifth International Conference on Autonomous Agents (Agents'01)*, 2001, pp. 513–520.