Degree of Local Cooperation and its Implication on Global Utility

Jiaying Shen*
Department of Computer Science
University of Massachusetts
Amherst, MA 01003-4610
USA
jyshen@cs.umass.edu

Xiaoqin Zhang
Computer and Information Science Department
University of Massachusetts Dartmouth
North Dartmouth, MA 02747-2300, USA
x2zhang@umassd.edu

Victor Lesser
Department of Computer Science
University of Massachusetts
Amherst, MA 01003-4610
USA
lesser@cs.umass.edu

Abstract

In a cooperative multi-agent system that is situated in an evolving environment, agents need to dynamically adjust their negotiation attitudes towards different agents in order to achieve optimal system performance. In this paper, we construct a statistical model for a small cooperative multi-linked negotiation system. It presents the relationship between the environment, the level of local cooperation and the global system performance in a formal and clear way that allows us to explain system behavior and predict system performance. The analysis results in a set of design equations that can be used to develop distributed mechanisms that optimize the performance of the system dynamically. It helps us more concretely understand the important issue of distraction and provides us with the local attitude parameter to handle distraction effectively. This research demonstrates that sophisticated probabilistic modelling can be used to understand the behaviors of a system with complex agent interactions, and provide guidelines for the development of effective distributed control mechanisms.

1. Introduction

In Multi-Agent systems, agents negotiate over task allocation, resource allocation and conflict resolution problems. In a cooperative system, agents work together to achieve optimal global utility. Unfortunately, when the environment is evolving over time, it is virtually impossible for the agents to always obtain and process all the necessary non-local information in order to achieve the optimal performance. Since centralized control is costly and impractical, most related research has been focusing on mechanisms that use local cooperation to approximate global cooperation. Only recently have researchers looked at different levels of local cooperation and its impact on the global performance.

There are different degrees of local cooperation when an agent is considering whether to cooperate with other agents on an external task or the use of a local resource (Figure 1(a)). An agent is *completely self-directed* when it does not take into consideration how much utility the other agent can potentially gain if it commits to the requested task. In contrast, an agent is *completely externally-directed* if it sees the other agent's gain as its own when negotiating. In this paper, we distinguish the notion of "self-interested" versus "cooperative" from "self-directed" versus "externally-directed". We call an agent *self-interested* if its local goal is to maximize only its local utility and an agent is *cooperative* if it is intent on maximizing the final social utility. Self-interestedness and cooperation illustrate the goal of an agent, while self-directness and externally-directness is the local mechanism used to achieve the goal. In a complex distributed system, where the environment is evolving over time, an agent has to dynamically choose the level of local cooperation that is optimal for its organizational goals based on its limited local vision and the information provided by other agents. Recent experimental work [8] found that different degrees of local cooperation have different impacts on global cooperation level and it is not always beneficial for an agent to be completely externally-directed. Understanding this relationship between local cooperation and global cooperation formally is very important for designing appropriate mechanisms to achieve optimal system performance.

In this paper, we construct a statistical model to formally analyze the relationship between the degree of local cooperation, the environment characteristics and the global utility achieved in a simple multi-linked negotiation setting. The model is verified by simulations and is used to explain and predict the system performance for the degree of local coop-

* The first author is a student.

eration and different environment parameters. Furthermore, we show that a simple learning mechanism based on this model can be used by each agent so that it can dynamically adjust its local cooperation level in response to the evolving environment so that optimal performance is achieved.

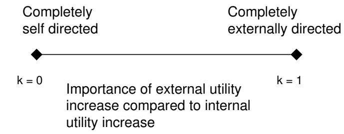
In an environment with uncertainty, the information provided by other agents may be inaccurate and prove a distraction for an agent's goal [2]. [1] proved that in a market system where agents are self-interested, if the trust an agent has for the other agent equals its trustworthiness, then the social welfare and the agents' utility functions are maximized. Our formal study described in this paper shows similar results in a cooperative system in the sense that the level of uncertainty directly affects the amount of self-directness that an agent should have in order to optimize the social utility. An agent should put appropriate weight on external information provided by other agents in an uncertain environment in order to deal with distraction. When there is more uncertainty related to the external information, an agent should be more self-directed. It should be more externally-directed if the external information has more certainty.

Research in Multi-Agent Systems community has been largely heuristic and experimental. Most formal work is done in systems with self-interested agents [6, 7, 5]. [3] analyzes the need for meta level communication in constructing a dynamic organizational structure. Our experience in building a formal model for a small cooperative multi-agent system demonstrates that sophisticated probabilistic techniques are useful in modelling complex interaction among agents. The analytical model is useful in understanding the behavior displayed in simulation results, and can be used as a base for designing distributed control mechanisms to improve the system performance in a dynamic environment.

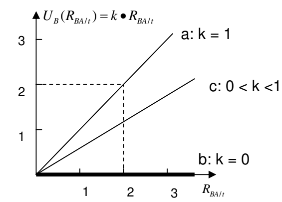
2. Integrative Negotiation

[8] introduced an integrative negotiation mechanism which enables agents to interact over a spectrum of different local cooperation degrees. During a negotiation session, an agent's attitude can vary to reflect how important its own utility increase is compared to the other agents' gains. When the agent only attaches importance to its own utility increase and not to the other agents', its attitude toward negotiation is completely self-directed; when it attaches the same degree of importance to the utility increase of other agents as it does to its own, its attitude is completely externally-directed (Figure 1(a)).

Let us take task allocation for example. There are two types of rewards that are transferred from agent A to agent B with the successful accomplishment of task t: real reward R_B and relational reward $R_{BA/t}$. Real reward R_B has positive benefits to agent B's utility. The agent collects real reward for its own utility increase and it is calculated into the social welfare increase as well. In contrast, the relational reward $R_{BA/t}$ does not contribute to agent B's actual utility increase, and it is not included in the social utility computation. Instead, it is transferred to reflect how important task t is for agent A and makes it possible for agent B to consider A's utility increase when it makes its negotiation decision. How $R_{BA/t}$ is mapped into agent B's virtual utility depends on agent B's negotiation attitude towards task t with agent A. Figure 1(b) shows different mapping functions for agent B. During its negotiation session with agent A about task t, agent B calculates its virtual utility for the task as $U_B(t) = U_B(R_B) + U_B(R_{BA/t}) = R_B + k \cdot R_{BA/t}$ and uses $U_B(t)$ to compare t against conflicting tasks if any.



(a) Attitude parameter as a measure of local cooperation level. The more weight an agent puts on the utility increase of the other agent, the more externally-directed it is.



(b) Different mapping from relational reward to local virtual utility reflects different degrees of local cooperation.

Figure 1. Degrees of local cooperation

Experimental work showed that it is not always beneficial for the agents in a cooperative system to be completely externally-directed [8]. When the uncertainty associated with the utility increase is high, it is better for the agent to be more self-directed. This indicates that complete local cooperation does not always lead to optimal global cooperation. Understanding the relationship between the local cooperation level and social welfare will help us better design a distributed system where agents can locally adjust their negotiation attitude and optimize the global utility.

3. General Problem

Let us formally define the class of problems we study.

There are a group of agents $A_1, A_2, \ldots, A_n$ and a set of tasks $T_1, T_2, \ldots, T_t$. Each task has a number of parameters that observe a distribution:

- r_i: task T_i arrives at an agent at time t with a probability of $1/r_i$.
- e_i: the difference between the arrival time of a task T_i and its earliest start time est_i.
- dur_i: the duration of the task T_i.

- sl_i: the difference between the earliest possible finish time of a task T_i and the deadline dl_i.
- R_i: the reward of a task T_i if it's finished.

The relationship of e_i, est_i, dur_i, sl_i and dl_i is illustrated in Figure 2.

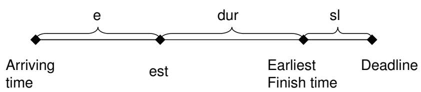


Figure 2. The relationship of different parameters of a task

Each task $T_i, 1 \leq i \leq t$ can be decomposed into a set of subtasks: $T_{i1}, T_{i2}, \ldots, T_{im_i}$, where m_i is the number of subtasks of T_i. All of the subtasks need to be completed in order for the agent A_s at whom T_i arrives to collect the reward. The agent can contract out some or all of the subtasks to other agents or it can finish the task on its own. As a special case, A_s can contract out the entire task T_i. Each subtask $T_{ij}, 1 \leq i \leq t, 1 \leq j \leq m_i$ has a set of parameters as well, and they have to observe certain relationships with each other and with the original task T_i:

- r_{ij}: $r_{ij} = r_i$.
- e_{ij}: the difference between the arrival time of a subtask T_{ij} and its earliest start time est_{ij}.
- dur_{ij}: the duration of the subtask T_{ij}.
- sl_{ij}: the difference between the earliest possible finish time of the subtask T_{ij} and its deadline dl_{ij}.
- R_{ij}: the reward of the subtask T_{ij} if it is finished. $\sum_j R_{ij} + R_{i0} = R_i$, where R_{i0} is the reward A_s gets after handing out the rewards to each subtask if all of the subtasks are completed.

For each subtask T_{ij} there is a set of agents AS_{ij} who can perform T_{ij}. When a task T_i arrives at agent A_s, A_s needs to do the following for each subtask T_{ij}:

1. Start to negotiate with one of the agent(s) in AS_{ij}.
2. Transmit the related parameters $e_{ij}, dur_{ij}, sl_{ij}, R_{ij}$. In addition, also transmit R_{i0}, i.e., the reward A_s itself will get if T_i is finished successfully.

When an agent A_l receives a request from agent A_s to do subtask T_{ij}, it does the following:

1. Decide whether T_{ij} can be fit onto its own schedule or can be contracted out (contracting out a subtask follows the afore mentioned procedure of a regular task); if yes, reply committed.
2. If there is a conflict between T_{ij} and A_s's own schedule and A_s cannot subcontract T_{ij} out to other agents, compare the utilities of the conflicting tasks and commit to the one with highest utility, and decommit from the other.

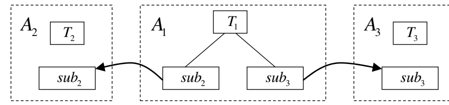


Figure 3. The simplest organization structure with the necessary inter-agent interactions

The utility of a subtask T_{ij} for A_l is calculated as $U_l(T_{ij}) = R_{ij} + k_{l,ij} * R_{io}$, where $k_{l,ij}$ is A_l's attitude parameter towards subtask T_{ij}. Here we use the reward R_{io} that will be received by agent A_s as the relational reward for A_l.

Each agent A_i is inherently cooperative, which means that its goal is to maximize the expected social utility.

We need to decide the relationship between the attitude parameter k, the environment parameters and the expected social utility. Furthermore, we need to design a mechanism that allows an agent to adjust its attitude parameters towards different agents and tasks in response to the ever changing environment and achieve optimal social utility.

4. A Statistical Analysis

4.1. An Example

Though relational reward provides agents with the information about the importance of the task to the other agent, this information may be inaccurate when the task requires cooperation from more than two agents. In this section, we describe a simple agent organization structure with the necessary inter-agent interactions during the negotiation process (shown in Figure 3) that exemplifies the class of problems and build an analytical model for this structure.

There are three agents in the system. A_1 has one type of task T_1 coming in, of which there are two different subtasks sub_2 and sub_3 that need to be contracted out to A_2 and A_3 respectively. Suppose at the same time, tasks T_2 and T_3 arrives at agents A_2 and A_3 and need to be completed. As a result, there may be conflicts between T_2 and sub_2, or between T_3 and sub_3, which force the agents to choose one task between the two. This decision depends on the real reward associated with each task, the relational reward from A_1, and also how the agent evaluates this relational reward, i.e., its attitude parameter toward it.

Let us consider the following example. Suppose agent A_2 faces the following situation: it receives a task T_2 with 15 units of real reward ($R_2 = 15$), and at the same time it receives a task proposal sub_2 with 6 units of real reward and 10 units of relational reward ($R_{12} = 6$ and $R_{11} = 10$). If A_2 is completely externally-directed towards A_1 regarding task sub_2 ($k = 1$), then the utility for sub_2 is $U_{sub_2} = 16 > 15 = U_{T_2}$. As a result, A_2 decides to accept task sub_2

and reject task T_2. However, A_3 rejects task sub_3 based on its situation, and therefore task T_1 can not be accomplished successfully. Since A_1 does not get the expected reward of 20 units, it cannot give A_2 the promised 6 units of real reward. At the same time, A_2 also loses the opportunity of accumulate 15 units real reward from T_2. The performance of this small organization is not optimized. The reason is that the information from A_1 is uncertain: the expected reward is based on the assumption of both subtasks are accomplished successfully, which depends on the local decisions of both A_2 and A_3.

How to deal with this uncertainty associated with multiple agents' local decision processes? We have built an analytical model presented in the following sections, so that agents can choose the appropriate attitude parameters to cope with the uncertainty.

4.2. Model Setup

We formally describe the three-agent organization (as shown in Figure 3) and the interaction among agents in this section. We chose this organization structure because it is minimal in its simplicity and yet complex enough to highlight the problem we are studying. We will extend this simple model to more complex structures as described in Section 3 in our future work.

There are three agents in the system. A_1 has task T_1 coming in, of which there are two subtasks sub_2 and sub_3 that need to be subcontracted to A_2 and A_3 respectively. At the same time, T_2 and T_3 arrive at A_2 and A_3.

1. T_i arrives at A_i with a probability of $1/r_i$ at each time unit. More formally, for any time t, $Pa_i(t) = 1/r_i$ represents the probability of there being a task T_i arriving at A_i at time t.
2. For task T_i, e_i, dur_i and sl_i are uniformly distributed:

$$\begin{aligned} P_{e_i}(x) &= \begin{cases} \frac{1}{be_i - ae_i}, & ae_i < x \le be_i \\ 0, & \text{o}therwise \end{cases} \\ P_{dur_i}(x) &= \begin{cases} \frac{1}{bd_i - ad_i}, & ad_i < x \le bd_i \\ 0, & \text{o}therwise \end{cases} \\ P_{sl_i}(x) &= \begin{cases} \frac{1}{bs_i - as_i}, & as_i < x \le bs_i \\ 0, & \text{o}therwise \end{cases} \end{aligned}$$

3. For the two subtasks sub_2 and sub_3, e_{1i}, dur_{1i} and sl_{1i} are uniformly distributed as well within the ranges $(ae_{1i}, be_{1i}]$, $(ad_{1i}, bd_{1i}]$ and $(as_{1i}, bs_{1i}]$, respectively. The parameters of the subtasks should bear some relationship with those of T_1 and with each other, and we should take care when setting these parameters so that it will reflect such a relationship. For example, the est of the subtasks cannot be earlier than T_1's est. There maybe an enable relationship between sub_2 and sub_3, and as a result, $est_{13} \ge dl_{12}$.

When an instance of T_1 arrives at A_1, A_1 will start negotiation processes with both A_2 and A_3. The two sessions are done in parallel, which means that the result of one negotiation does not affect the other. Associated with each T_1 is a reward R_1. Upon completion of T_1, A_1 will collect a part of the total reward R_{11} for itself, and hand the rest R_{12} and R_{13} to A_2 and A_3 respectively. $R_1 = R_{11} + R_{12} + R_{13}$. In order for the reward to be collected, both sub_2 and sub_3 have to be completed.

During the negotiation process with agent A_i about a subtask sub_i, A_1 promises a real reward R_{1i} for completing the task and tells A_i about the reward that A_1 itself will gain if the task is completed, i.e., the relational reward R_{11}. A_i's attitude parameter toward A_1 about doing sub_i is k_i. As a result, the utility of the subtask sub_i for A_i when it is making the negotiation decision is $Rn_i = R_{1i} + k_i \cdot (R_{11})$. R_1, R_{11}, R_{12} and R_{13} are all constants.

T_2 and T_3 both have a reward, R_2 and R_3 respectively, which are uniformly distributed within the range of $(ar_2, br_2]$ and $(ar_3, br_3]$.

Once having received a subtask (sub_i) request, the agent A_i sees whether there is a conflict between the new task and other tasks (both the previous commitment to A_1 and its local task T_i). These other tasks include the tasks that came in before the new one and those will come in after it. If there is no conflict, A_i will commit to the task. Otherwise, it will choose the task with higher reward.

There is no task failure or explicit decommitment once a commitment is made. The only time that a contract is breached is by A_1 if it receives a commitment from one of the agents and not the other. In this case, the committed agent will still execute the subtask as it promised but receive no promised reward for that. This is a simple negotiation protocol and some of the uncertainty we discuss above can be resolved by a more sophisticated negotiation protocol. We model this simple protocol for two reasons. First, some of the necessary techniques are developed in the process and can be extended to model other protocols. Second, we demonstrate that in an environment without sophiscated global design or with a tight communication restriction, local mechanisms such as attitude parameter can be used effectively to cope with the resulting uncertainty.

4.3. Probability of Conflict

An agent needs to choose between tasks to execute when and only when there is a conflict between tasks. A task of type i is in conflict with a task of type j (whether it came before task i or after) if and only if there exists a task of type j such that the following two inequalities are both true:

$$\begin{aligned} dl_i - est_j &\le dur_i + dur_j, \\ dl_j - est_i &\le dur_i + dur_j. \end{aligned} \quad (1)$$

Rewriting (1) in terms of est, dur and sl, we get

$$sl_i - dur_j \le est_j - est_i \le dur_i - sl_j \quad (2)$$

For a task of type i that arrives at a given time, we define Pc_{ij} as the probability of there being a task of type j that has conflict with it. Notice that for task i, we only know of its arriving time, not its other relevant parameters. In addition, we do not know any parameter of task j.

$$\begin{aligned} & Pc_{ij} \\ = \; & P(sl_i - dur_j \le est_j - est_i \le dur_i - sl_j) \\ = \; & \sum_{z=-\infty}^{+\infty} \sum_{y=z}^{+\infty} (1 - \prod_{x=z}^{y} (1 - P_{est_j - est_i}(x))) \cdot \\ & \qquad P_{dur_i - sl_j}(y) P_{sl_i - dur_j}(z) \end{aligned} \quad (3)$$

First let us look at $P_{est_j - est_i}(x)$, the probability of the difference between the earliest start time of tasks T_i and T_j being x. Since the arrival time of task i is fixed, without loss of generality, let us define the arriving time of task i as 0. As a result, $est_i = e_i$, and est_j can range from $-\infty$ to $+\infty$. Therefore, $P_{est_j - est_i}(x) = P(est_j - e_i = x)$, i.e., the probability of there existing a task j that satisfies $est_j - e_i = x$. We first solve the probability of there being a task whose est is at a specified time t, which we write as $P(est = t)$:

$$\begin{aligned} P(est = t) & = \sum_{x=-\infty}^{+\infty} Pa(t - x) P_e(x) \\ & = \sum_{x=ae+1}^{be} \frac{1}{r} \cdot \frac{1}{be - ae} \\ & = \frac{1}{r} \end{aligned} \quad (4)$$

Then we can further calculate $P_{est_j - est_i}(x)$:

$$\begin{aligned} P_{est_j - est_i}(x) & = \sum_{y=-\infty}^{+\infty} P_{e_i}(y) P(est_j = y + x) \\ & = \sum_{y=ae_i+1}^{be_i} \frac{1}{be_i - ae_i} \cdot \frac{1}{r_j} \\ & = \frac{1}{r_j} \end{aligned} \quad (5)$$

Now let us see what $P_{dur_i - sl_j}(y)$ is.

$$\begin{aligned} & P_{dur_i - sl_j}(y) \\ = \; & \sum_{x=-\infty}^{+\infty} P_{dur_i}(x) \cdot P_{sl_j}(x - y) \\ = \; & \begin{cases} \frac{bd_i - as_j - y}{(bd_i - ad_i)(bs_j - as_j)}, & \max(ad_i - as_j, bd_i - bs_j) < y < bd_i - as_j; \\ \frac{1}{bd_i - ad_i}, & ad_i - as_j \le y \le bd_i - bs_j; \\ \frac{1}{bs_j - as_j}, & bd_i - bs_j \le y \le ad_i - as_j; \\ \frac{bs_j + y - ad_i}{(bd_i - ad_i)(bs_j - as_j)}, & ad_i - bs_j < y < \min(ad_i - as_j, bd_i - bs_j); \\ 0, & otherwise. \end{cases} \end{aligned} \quad (6)$$

Similarly, we get

$$\begin{aligned} & P_{sl_i - dur_j}(z) \\ = \; & \begin{cases} \frac{bs_i - ad_j - z}{(bs_i - as_i)(bd_j - ad_j)}, & \max(as_i - ad_j, bs_i - bd_j) < z < bs_i - ad_j; \\ \frac{1}{bs_i - as_i}, & as_i - ad_j \le z \le bs_i - bd_j; \\ \frac{1}{bd_j - ad_j}, & bs_i - bd_j \le z \le as_i - ad_j; \\ \frac{bd_j + z - as_i}{(bs_i - as_i)(bd_j - ad_j)}, & as_i - bd_j < z < \min(as_i - ad_j, bs_i - bd_j); \\ 0, & otherwise. \end{cases} \end{aligned} \quad (7)$$

Now, we can put (5), (6) and (7) back to (3) and get the probability of there being a conflict for a task that comes in at a given time. Please note that this calculation of P_{ij} is an approximation, since we are only considering the probably of two tasks conflicting with each other. In reality, there might be three or more tasks that can not be scheduled successfully at the same time but any two of them can be. Therefore, the real probability of conflict may be slightly higher than our approximation.

4.4. Expected Reward

What we are really concerned about is the expected reward that the system may receive at any given time. Multiplying it by the time that the system has run yields the expected reward of the system.

For A_2 and A_3, there may be two types of tasks coming in at any moment: the local task T_i with a probability of $1/r_i$ and the non-local task sub_i with a probably of $1/r_1$. When a local task T_i for A_i arrives, it accumulates reward only under one of the following circumstances:

1. There is a conflict between it and one non-local task sub_i and there is no conflict with other local tasks. In addition, the local task reward is greater than the utility of the non-local task that it is in conflict with, i.e., $R_i > Rn_i = R_{1i} + k_i \cdot R_{11}$. Therefore,

$$\begin{aligned} & E(R_i | R_i > Rn_i) \\ = \; & \sum_{x=\lfloor Rn_i \rfloor + 1}^{br_i} P_{R_i}(x) \cdot x \\ = \; & \begin{cases} \frac{ar_i + br_i + 1}{2}, & \lfloor Rn_i \rfloor < ar_i; \\ \frac{(br_i - \lfloor Rn_i \rfloor)(br_i + \lfloor Rn_i \rfloor + 1)}{2(br_i - ar_i)}, & ar_i \le \lfloor Rn_i \rfloor < br_i; \\ 0, & \lfloor Rn_i \rfloor \ge br_i. \end{cases} \end{aligned} \quad (8)$$

The part of expected reward gained by executing the new task in this case is then:

$$ER_i^{(1)} = Pc_{1i,i} \cdot (1 - Pc_{ii}) \cdot E(R_i | R_i > Rn_i) \quad (9)$$

2. The only conflict caused by this task is with another local task T_i'. In addition, the new reward is higher than that of T_i'. The expected reward gained by executing this task under this condition is:

$$\begin{aligned} & ER_i^{(2)} \\ = \; & (1 - Pc_{1i,i}) \cdot Pc_{ii} \cdot [E(R_i | R_i > R_i') \\ & + \frac{1}{2} E(R_i | R_i = R_i')] \end{aligned} \tag{10}$$

where

$$E(R_i | R_i > R_i') = \sum_{y=ar_i+1}^{br_i} \sum_{x=y+1}^{br_i} x P_{R_i}(x) P_{R_i}(y)$$

and

$$\begin{aligned} \frac{1}{2} E(R_i | R_i = R_i') &= \sum_{x=ar_i+1}^{br_i} x (P_{R_i}(x))^2 \\ &= \frac{ar_i + br_i + 1}{4(br_i - ar_i)} \end{aligned}$$

3. There is a conflict with both another local task and a non-local task. In addition, the reward gained by the new local task is the highest.

$$\begin{aligned} & ER_i^{(3)} \\ = \; & Pc_{1i,i} \cdot Pc_{ii} \cdot [E(R_i | R_i > Rn_i \& R_i > R_i') \\ & + \frac{1}{2} E(R_i | R_i > R_{ni} \& R_i = R_i')] \end{aligned} \tag{11}$$

where

$$\begin{aligned} & E(R_i | R_i > Rn_i \& R_i > R_i') \\ = \; & \sum_{y=ar_i+1}^{br_i} \sum_{x=\max(\lfloor Rn_i \rfloor + 1, y+1)}^{br_i} P_{R_i}(x) P_{R_i}(y) x \\ = \; & \frac{1}{(br_i - ar_i)^2} \sum_{y=ar_i+1}^{br_i} \sum_{x=\max(\lfloor Rn_i \rfloor + 1, y+1)}^{br_i} x \end{aligned}$$

and

$$\begin{aligned} & \frac{1}{2} E(R_i | R_i > R_{ni} \& R_i = R_i') \\ = \; & \frac{1}{2} \sum_{x=\lfloor Rn_i \rfloor + 1}^{br_i} [P_{R_i}(x)]^2 \cdot x \\ = \; & \begin{cases} 0, & \lfloor Rn_i \rfloor \geq br_i; \\ \frac{(br_i - \lfloor Rn_i \rfloor)(br_i + \lfloor Rn_i \rfloor + 1)}{4(br_i - ar_i)^2}, & ar_i \leq \lfloor Rn_i \rfloor < br_i; \\ \frac{ar_i + br_i + 1}{4(br_i - ar_i)}, & \lfloor Rn_i \rfloor < ar_i. \end{cases} \end{aligned}$$

4. There is no conflict caused by the new task.

$$ER_i^{(4)} = (1 - Pc_{1i,i})(1 - Pc_{ii}) \cdot \frac{ar_i + br_i}{2} \tag{12}$$

Similarly, when a subtask sub_i arrives at A_i, A_i will choose to commit to it under four conditions, but it can accumulate this reward only when the other agent decides to commit to the other subtask as well. Therefore the expected reward will be:

$$ER_i^{(5)} = R_{1i} \cdot Pcommit_2 \cdot Pcommit_3 \tag{13}$$

where

$$\begin{aligned} & Pcommit_i \\ = \; & Pc_{1i,i}(1 - Pc_{11}) P(Rn_i \geq R_i) \\ & + \frac{1}{2} Pc_{1i,i} \cdot Pc_{11} P(Rn_i \geq R_i) \\ & + \frac{1}{2} (1 - Pc_{1i,i}) Pc_{11} \\ & + (1 - Pc_{1i,i})(1 - Pc_{11}) \end{aligned} \tag{14}$$

and

$$\begin{aligned} P(Rn_i \geq R_i) &= \sum_{x=ar_i+1}^{\lfloor Rn_i \rfloor} P_{R_i}(x) \\ &= \begin{cases} 1, & \lfloor Rn_i \rfloor \geq br_i \\ \frac{\lfloor Rn_i \rfloor - ar_i}{br_i - ar_i}, & ar_i \leq \lfloor Rn_i \rfloor \leq br_i \\ 0, & \lfloor Rn_i \rfloor \leq ar_i \end{cases} \end{aligned} \tag{15}$$

Now we have the expected reward that A_2 or A_3 collects at each time unit:

$$ER_i = \frac{1}{r_i}(ER_i^{(1)} + ER_i^{(2)} + ER_i^{(3)} + ER_i^{(4)}) + \frac{1}{r_1} ER_i^{(5)} \tag{16}$$

Let us have a look at the expected reward that A_1 collects at each time unit. There is only one type of task coming in to A_1. The reward can be collected if and only if both of the other two agents commit to the subtasks. As a result,

$$ER_1 = \frac{1}{r_1} \cdot R_{11} \cdot Pcommit_2 \cdot Pcommit_3 \tag{17}$$

Now that we have the expected reward for each of the agents, we can calculate the k_i that will maximize the social utility given the set of the parameters. More formally, we set k_2 and k_3 to be:

$$\arg \max_{k_2, k_3} (ER_1 + ER_2 + ER_3).$$

Please notice that when we are calculating the expected reward collected by each of the agents by executing T_1 we are assuming perfect knowledge of the other agent's model. This is useful from a system designer's perspective. Having a global view of the system, the designer can set the attitude parameter k of each agent such that the global utility can be maximized.

We ran a set of simulations in the integrative negotiation framework with different parameter settings (Table 1)

	r	est	dur	dl	R
T_2	t2	14	6	26+td2*[1,3]	2+tr2*[1,3]
T_3	t3	24	7	34+td3*[1,3]	2+tr3*[1,3]
sub_2	15	12	7	20+[0,2]	3
sub_3	15	23+[0,2]	6	35+[0,2]	3
T_1	15	12		35+[0,2]	25

Table 1. Simulation parameter setting

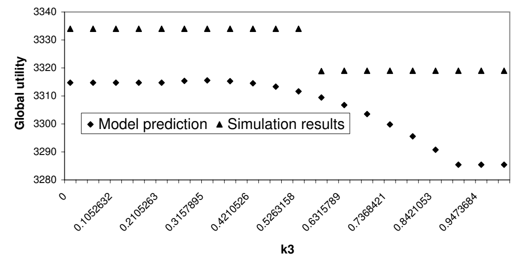


Figure 4. Comparison of the model prediction and the simulation results. $t2 = t3 = 10$**,** $td2 = td3 = 1$**,** $tr2 = tr3 = 6$**.**

to verify the model. We vary the arrival rate, deadline and reward of the tasks and record the social utility generated by the system after 950 time units for different attitude parameters k_2 and k_3. As seen in Figure 4, the simulation results and the theoretical prediction match well with each other, with a utility difference of around 1%. The difference in the two curves are mainly caused by the two major differences between the simulator and our theoretical model. First, the tasks in the simulator arrives at the agents every r_i time step instead of with a probability of $1/r_i$ at each step. Though these two settings are statistically equivalent, the simulator has less chance of the same type of tasks conflicting with each other, and results in a higher utility generated by the simulation. Second, the simulator uses a scheduler that schedules all the tasks in a fixed time window together and resolves the conflicts among them. Once a task is successfully scheduled, it will not be removed from the schedule or shifted to accomodate tasks arriving in the next time window. As a result, the simulator is not as sensitive to slight parameter changes as the model is, which leads to the gadual drop in utility in the theoretical model versus the step function drop in the simulator. Other parameter settings show a similar correlation between the simulation results and the model prediction. As tasks become less flexible (varied by r and dl), conflicts become increasingly likely and global utitlity is reduced. The higher a local task's reward is compared to that of the subtask, the less likely T_1 will be finished and the more self-directed the other agent should be for the system to collect more reward. These behaviors are both predicted and explained by the model and the resultant equations.

5. Adjusting Local Attitude

In a real system, the environment may evolve over time. In such situations, it is unlikely that a static organization will remain optimal as the environment changes. Furthermore, it is impractical for the agents to always have a global view of the system without significant communication cost. Fortunately, an agent can often learn the other agents' behavior through past interactions with them. If agents can dynamically adjust their relationships with other agents based on observations of each other, then the system can achieve more global utility than a static system.

The agents A_2 and A_3 can learn the probability of the reward being actually collected from A_1 by recording the interaction history between them. From these statistics, they are able to choose their own attitude parameters (k_i) in order to maximize the total utility that may be collected by them and A_1. Expressed more formally, if agent A_2 observes the probability of A_1 handing out the reward for sub_2 as P_2, then $ER_2^{(5)}$ and ER_1 are written for A_1 as follows:

$$ER_2^{(5)} = R_{12} \cdot Pcommit_2 \cdot P_2; \qquad (18)$$

$$ER_1 = \frac{1}{r_1} \cdot R_{11} \cdot Pcommit_2 \cdot P_2. \qquad (19)$$

In order to maximize the social utility as its vision of the environment allows, A_2 should set k_2 as:

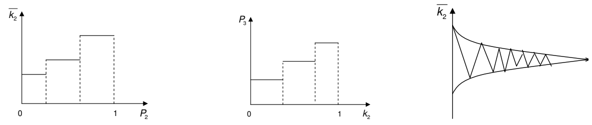
$$k_2 = \arg\max_{k_2}(ER_1 + ER_2). \qquad (20)$$

It is the same for A_3.

There are two cases of environment change to consider. First, there is a change happening at A_2 or A_3 which makes the corresponding agent adjust its k_i. Second, there is a change of the local parameters at A_1 that leads to a change in k_i in one or both of the agents' attitude. When such change happens, one or both of the agents initiate the adjustment in their attitude parameters k_i in response, which leads to a change in the other agent's observation of P_i and further adjustment of k_i. We prove the following theorem:

Theorem 1 *The local adjustment of the attitude parameters is stable, i.e., the process will converge.*

PROOF. If we fix the parameters other than k_2 and denote the utility that A_2 is trying to maximize as U_2, we can write it as a function of $x_2 = \lfloor Rn_2 \rfloor$: $U_2 = ER_1 + ER_2 = -a \cdot x_2^2 + (b + d \cdot P_2) \cdot x_2 + c$, when $ar_2 \le \lfloor Rn_2 \rfloor \le br_2$, where a, b, c, d are all constants. Then we have the optimal $\lfloor Rn_2 \rfloor$ as $\overline{x_2} = \frac{b + d \cdot P_2}{2a}$. Since $x_2 = \lfloor Rn_2 \rfloor = R_{12} + k_2 \cdot R_{11}$, the optimal $\overline{k_2}$ changes monotonically as P_2 changes (shown in Figure 5(a)). When A_2 sets its new k_2, A_3's observation of P_3 changes accordingly: $P_3 = e \cdot \lfloor Rn_3 \rfloor + f$ when $ar_3 \le \lfloor Rn_3 \rfloor \le br_3$, where e and f are constants.



(a) (b) (c)

Figure 5. (a) $\overline{k_2}$ changes monotonically as P_2 changes. (b) P_3 changes monotonically as k_2 changes. (c) $\overline{k_2}$ converges over time even when k_2 and k_3 change in different directions at the same time.

As shown in Figure 5(b), P_3 changes monotonically as k_2 changes as well.

No matter what change in the environment causes the change in local parameter k_i, the value of k_i either increases or decreases. If the changes of both agents are towards the same direction, i.e., both of them increase, both decrease, or one of them stays the same, then as Figures 5(a) and 5(b) show, both k_2 and k_3 change monotonically without oscillation. Since there are only limited number of different values for $\lfloor Rn_i \rfloor$, k_i will converge to a certain value.

On the other hand, if k_2 and k_3 start changing towards different directions, they will both oscillate, as the directions of change caused by the two agents are different. Fortunately, the oscillation is bounded by the curves of change in k_i in Figure 5(a) (as shown in Figure 5(c), and the process will converge in the end. $\square$

Theorem 1 tells us that it is safe for the agents to adjust their attitude parameters locally and reach a global equilibrium. We can add a simple learning component to each agent A_i which observes the probability of A_1 handing out the reward for sub_i as P_i and adjust k_i to the optimal value related to P_i dynamically.

In an environment with uncertainty, the information provided by other agents may be inaccurate and prove a distraction for an agent's goal [2]. [2, 4] suggest that mechanisms that appropriately handle distraction in a complex multi-agent system are important to improving the overall system performance. In the three agent multi-linked negotiation system we are modelling in this paper, there is uncertainty related to the rewards that A_1 promises to A_2 and A_3 and may prove distracting. P_2 and P_3 are good measures of this uncertainty. The proof of Theorem 1 shows that the level of uncertainty in the external information received from A_1 directly affects the amount of self-directness that an agent should have in order to optimize the social utility. As seen in Figure 5(a), the greater the value of P_i is, the higher the optimal k_i is, which means the more externally-directed A_i should be towards A_1 regarding sub_i. Likewise, when there is more uncertainty related to the external information, an agent should be more self-directed. Therefore the attitude parameter of an agent can be seen as an effective way to handle distraction introduced by uncertain external information.

6. Conclusions

In this paper, we successfully constructed a statistical model for a small cooperative multi-link negotiation system. It shows us the relationship between the environment, the level of local cooperation and the global system performance in a formal and clear way that allows us to explain system behavior and predict system performance. The analysis also results in a set of design equations that can be used directly to design distributed local mechanisms that optimize the performance of the system dynamically. Finally, it helps us more concretely understand the important issue of distraction that was first discovered and studied by [4, 2] and provides us with the local attitude parameter to handle distraction effectively. This research demonstrates that sophisticated probabilistic modelling can be used to understand the behaviors of a system with complex agent interactions, and provide guidelines for the development of effective distributed control mechanisms. Though what we present in this paper is a model of a simple three agent system, both the model itself and the techniques we use can be extended to more interesting and larger systems with more complex inter-agent interactions.

References

[1] S. Braynov and T. Sandholm. Contracting with uncertain level of trust. *Computational Intelligence*, 18(4):501–514, 2002.

[2] M. H. Chia, D. E. Neiman, and V. R. Lesser. Poaching and distraction in asynchronous agent activities. In *Proceedings of the Third International Conference on Multi-Agent Systems*, pages 88–95, 1998.

[3] K. Decker and V. Lesser. An Approach to Analyzing the Need for Meta-Level Communication. *International Joint Conference on Artificial Intelligence*, 1, January 1993.

[4] V. Lesser and L. Erman. Distributed Interpretation: A Model and an Experiment. *IEEE Transactions on Computers, Special Issue on Distributed Processing*, C-29(12):1144–1163, December 1980.

[5] S. Saha, S. Sen, and P. S. Dutta. Helping based on future expectations. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 289–296, Melbourne, Australia, 2003. ACM Press.

[6] T. Sandholm, S. Sikka, and S. Norden. Algorithms for optimizing leveled commitment contracts. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 535–540, Stockholm, Sweden, 1999.

[7] S. Sen. Believing others: Pros and cons. *Artificial Intelligence*, 142(2):179–203, December 2002.

[8] X. Zhang, V. Lesser, and T. Wagner. Integrative negotiation in complex organizational agent systems. In *Proceedings of the 2003 IEEE/WIC International Conference on Intelligent Agent Technology (IAT 2003)*, pages 140–146, 2003.