Lies in Multiagent Subadditive Task Oriented Domains

Rivka Yagodnick
Jeffrey S. Rosenschein
Institute of Computer Science
The Hebrew University
Jerusalem, Israel
{rivki,jeff}@cs.huji.ac.il

ABSTRACT

1 Introduction

Coalition formation is one of the great unexploited potentials of the Internet. Despite a fair amount of work on coalitions in the Distributed Artificial Intelligence (DAI) community in recent years [35, 18, 10, 32, 37, 36], practical consequences of this research remain largely invisible to users. The untapped power of coalition formation is an important opportunity for negotiation technology in the near term.

Dynamically formed internet coalitions hold open the promise of harnessing the power of groups in many contexts:

- Coalitions of consumers could form to present an ecommerce vendor with a more attractive group purchase; the price of one car might be considerably higher than the per car price in a sale involving 100 cars. Such group purchasing coalitions, manually constructed, are an existing reality on the internet [33]. Automatic coalition formation among agents could make this process automatic and far more effective.

- Another type of coalition would be that of contributors to a group effort, for example, a coalition of computer users who contribute processing time to break a cryptographic puzzle. These kinds of coalitions, manually constructed, have also become popular on the internet, and there have been architectures developed for their semi-automatic formation [2].

- Dynamically formed coalitions of vendors could respond to a buyer's specialized request. For example, a school district might want to construct a computing infrastructure involving multiple vendors' equipment. Interested companies could form a coalition to satisfy the school district's request for proposals.

Again, coalitions and communities of interest already exist on the Web, but their formation is manual, not automatic. The automation of coalition formation is the key to unleashing coalition power.

There are a large number of architectural and theoretical challenges to be overcome before dynamic coalitions can throw their weight around on the internet. Like-minded participants must have a way of finding one another; the definition of "like-minded" in a given context must be determined. The division of labor, and the reintegration of sub-results, may need to be handled. The contribution of each coalition member to the group effort might need to be ascertained and rewarded appropriately [34, 38] (techniques like the Shapley Value [21, 20] may be useful in these cases). There may be issues of privacy and security as well (for example, I may be willing to enter into a group purchase of cars, but I'd rather that only the dealer knew exactly which optional accessories I was ordering).

Another important issue that needs to be resolved in the formation of coalitions is the veracity of the participants. Can a potential coalition partner gain an advantage by lying, either about its goals or capabilities? We have been examining this question, building on previous work in two-agent Task Oriented Domains [17, 39].

2 Task Oriented Domains

A task oriented domain (TOD) is an environment in which agents interact, that has the following features. There exists a set of tasks that can be executed in the domain, and there is a cost function that maps any set of tasks into an associated cost of execution. In an agent encounter in a TOD, each agent is given a set of tasks to accomplish. There is no interaction (either positive or negative) among tasks; each task can be executed without regard to any others, and there are no action side-effects. Typical TODs include the Postmen Domain, Database Queries, and the Fax Domain.

Sometimes, agents in an encounter may find it to their benefit to exchange tasks, thereby lowering their overall cost of execution. When agents are able to agree on such an exchange, they have reached a deal. The original work on TOD's considered only two-agent encounters, and explored the potential for deception as agents tried to reach a deal. For example, an agent could sometimes benefit by hiding some of its tasks, or by creating a fake task, that would improve its position in the negotiation.

Certain DAI work [18, 10, 37] has recently focused on more-than-two-agent domains, and general coalition formation. Groups of agents may agree to work together for the benefit of all the members of the group. The challenges here are to find these coalitions of agents, and to determine how the joint reward should be divided among them. Most of this work assumes complete knowledge among the interacting agents and/or truthful agents. These assumptions are not realistic in many domains, including standard internet domains where dynamic coalitions are being formed by agents representing user interests. The possibility of incomplete information among agents opens up the possibility of deception as part of the negotiating strategy of an agent, as was examined for TODs in the two-agent case.

What we are examining in this paper is thus the extension of the original TOD work from the two-agent case to the multiple agent case. Although restricted to the simplest of domains, it begins to uncover the opportunities for deception in coalition formation, which has direct bearing on the dynamic, automatic formation of coalitions on the internet.

In this paper, we present the main theorems and results of this work (including proofs). We build freely on the previous work of Zlotkin and Rosenschein [17] on Task Oriented Domains, using their concept of the domain, encounters, deals, and protocols. (These results are taken out of a larger work, and the numbering of figures and theorems sometimes reflects this fact.)

3 Mechanisms that Maximize the Product of Utilities

In this section, we present the concept of a *negotiation protocol*. The protocol specifies the rules of the negotiation, the rules by which the agents will come to a consensus, agreeing to carry out one of the deals in the negotiation set.

In this chapter we introduce a class of mechanisms called the Product Maximizing Mechanisms (PMMs). This class of mechanisms was already introduced by Rosenschein and Zlotkin [17] for the two-agent case. Here, we extend their work for the general case.

3.1 Mechanism Definition

Definition 14: A *product maximizing mechanism,* or a PMM is a mechanism that satisfy the following conditions:

- The protocol is symmetrically distributed;
- The strategy is in equilibrium with itself;
- Given the protocol, if the agents play the strategy they will agree on a deal that maximizes the product of their utilities. If there is more than one product-maximizing deal, they will agree on a deal (among the product maximizers) that maximizes the sum of utilities. If there is more than one such sum-maximizing deal, the protocol will choose among those deals arbitrarily.

In this section we assume that the agents are using some kind of product-maximizing protocol, such as in Nash bargaining theory. For the purposes of this work, it really does not matter which protocol the agents use as long as it is a PMM. There are all sort of examples of different protocols that will bring the agents to an agreement that maximizes the product of their utilities. For example, a *one-step protocol,* where all the agents can simultaneously make a single offer, and the offer that maximizes the product of utilities will be chosen. If there is more than one offer that maximizes the product of utilities, one of the “maximizers” is chosen arbitrarily. Another example is a *monotonic concession protocol,* where each agent first starts with a deal that is best for it and then iteratively makes compromises to the other agents.

Now, let’s return to the last stage in the coordination process. Given the space of possible deals and the negotiation protocol, the next question is what negotiation strategy the agents will use? When the agents have complete information, strategy is not really important because once we have decided on the protocol, and all agents’ tasks are commonly known, there’s a well-defined agreement point. This well-defined point tells the agents when they have maximized the product of their utilities. They can just move to this point in one way or another. It doesn’t really matter how.

3.2 Evaluation of the Negotiation Mechanism

Here we consider the attributes of the above negotiation process with regard to its efficiency, stability, simplicity, distribution and symmetry. These are the attributes we want our mechanisms to exhibit.

3.2.1 Efficiency of the mechanism

<u>Theorem 10</u>: All PMMs are efficient.

<u>Proof</u>:

Let δ be a deal that the agents agree on after negotiation when they are using a PMM. In order to show that all PMMs are efficient, it is sufficient to show that δ is pareto optimal. Using proof by contradiction, let's assume that there exists another deal δ'∈ NS such that δ'≻δ.

This means that ∀i 1≤i≤n $Utility_i(\delta')\geq Utility_i(\delta)$ and that there exists j 1≤j≤n such that $Utility_j(\delta')>Utility_j(\delta)$.

Therefore, $\sum_{i=1}^{n} Utility_i(\delta') > \sum_{i=1}^{n} Utility_i(\delta)$.

However, from the "constant-sum equality" theorems (8 for mixed deals and 9 for all-or-nothing deals) we know that for any deal λ∈ NS, $C \equiv \sum_{i=1}^{n} Utility_i(\lambda) = \sum_{i=1}^{n} c(T_i) - c\left(\bigcup_{i=1}^{n} T_i\right)$.

Thus, $\sum_{i=1}^{n} Utility_i(\delta') = \sum_{i=1}^{n} Utility_i(\delta)$, which is a contradiction.

3.2.2 Symmetry of the Mechanism

<u>Theorem 11</u>: [Equal Division of Utility]: A PMM over mixed deals (or over all-or-nothing deals) in subadditive multi-agent TODs divides the available utility equally between the agents.

<u>Proof</u>:

For any deal δ∈ NS, $C \equiv \sum_{i=1}^{n} Utility_i(\delta) = \sum_{i=1}^{n} c(T_i) - c\left(\bigcup_{i=1}^{n} T_i\right)$ because of the "constant-sum equality" theorems (8 for mixed deals and 9 for all-or-nothing deals).

From the definition of a PMM we know that the agreement that will be reached will maximize the product of the agents' utilities. Let δ be a deal that the agents agree on after negotiation when they are using a PMM.

If C=0, then ∀i $Utility_i(\delta)=0$ and therefore δ divides the available utility equally between the agents.

Otherwise ($C \neq 0$), if $\exists i$ such that $Utility_i(\delta)=0$, then $\prod_{i=1}^{n} Utility_i(\delta) = 0$, which is certainly not the maximum. Therefore, we may assume that $\forall i$ $Utility_i(\delta)>0$.

According to the averages inequality, $\forall n \in N$ if $\forall 1 \leq i \leq n$ $x_i>0$ then $\sqrt[n]{\prod_{i=1}^{n} x_i} \leq \frac{\sum_{i=1}^{n} x_i}{n}$ and equality exist iff all the x_i's are equal.

Therefore, $\sqrt[n]{\prod_{i=1}^{n} Utility_i(\delta)} \leq \frac{\sum_{i=1}^{n} Utility_i(\delta)}{n} = \frac{C}{n}$ and equality exist iff all the $Utility_i(\delta)_i$'s are equal.

Thus, $\prod_{i=1}^{n} Utility_i(\delta) \leq \left(\frac{C}{n}\right)^n$ and the maximum is received when all the $Utility_i(\delta)_i$'s are equal.

Therefore δ divides the available utility equally between the agents.

Thus, *all* the agents would be assigned the same utility when using a PMM over mixed or all-or-nothing deals, and particularly *symmetric* agents would.

When using a PMM over pure deals, it may happen that two symmetric agents won't get the same utility. For example, consider the encounter from the Postmen Domain given in Figure 6.

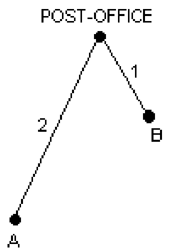

Figure 6

The length of each edge is written next to it. The post office is at the root of the tree. There are two agents. Both of them need to deliver letters to nodes A and B.

There are two pure deals that maximize the product of utilities. These deals are ({A},{B}) and ({B},{A}). ({A},{B}) gives agent 1 utility 2 and agent 2 utility 4, ({B},{A}) gives agent 1 utility 4 and agent 2 utility 2. Thus, no matter which of those deals the mechanism chooses, the two symmetric agents get different utility. However, the protocol chooses one of those deals with equal probability. Thus, both agents get expected utility of 3.

This phenomenon also happens in the general case. If a pure deal that maximizes the product of utilities doesn't give equal utility to two symmetric agents, there is a symmetric pure deal that makes the opposite discrimination. Thus, any PMM over pure deals gives two symmetric agents the same expected utility.

3.2.3 Stability of the Mechanism

- **Individual Rationality**

Since agreements are restricted to be in the negotiation set, and since the negotiation set is not empty (i.e., an agreement is guaranteed), the result will be individual rational.

- **Group Rationality**

This condition is equivalent to efficiency, and thus according to 5.2.1., all PMMs satisfy group rationality.

- **Coalition Rationality**

Since we assume that the full coalition will be formed, the coalition rationality means that no group of agents should have any incentive to deviate from the full coalition and creates a sub-coalition. This condition is very strong, and it is not always satisfied by PMMs.

Consider the encounter from a three-agent Postmen Domain that can be seen in Figure 7.

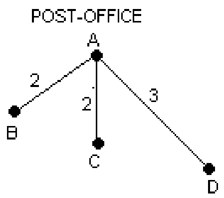

Figure 7

The Post-Office is at node A. The length of each arc is written next to it. The encounter is (T_1 = {B,C}, T_2 = {B,C}, T_3 = {D}). Thus, the costs of delivering the subsets of letters are as follows: $C(T_1) = 8$, $C(T_2) = 8$, $C(T_3) = 6$, $C(T_1 \cup T_2) = 8$, $C(T_1 \cup T_3) = 14$, $C(T_2 \cup T_3) = 14$ and $C(T_1 \cup T_2 \cup T_3) = 14$.

The joint utility of all the agents from any mixed deal in the negotiation set is $C(T_1) + C(T_2) + C(T_3) - C(T_1 \cup T_2 \cup T_3) = 8 + 8 + 6 - 14 = 8$.

Any PMM over mixed deals divides the available utility equally between the agents. Thus, in this example any PMM chooses a deal that gives all the agents utility of $2\frac{2}{3}$. In this case, the joint utility of agents 1 and 2 is $5\frac{1}{3}$. However, if agents 1 and 2 collude and form a sub-coalition, their joint utility will be (according to theorem 8): $C(T_1) + C(T_2) - C(T_1 \cup T_2) = 8 + 8 - 8 = 8$. Thus, agents 1 and 2 are motivated to deviate from the full coalition.

The above example is from a Postmen Domain that has a star topology, i.e., a modular TOD. Thus, even in modular TODs, no PMM over mixed deals satisfies the coalition rationality.

It is not very hard to show (using the same example) that the above conclusion is true over pure deals and all-or-nothing deals as well.

3.2.4 Simplicity of the Mechanism

- **Computational Complexity**

The agents don’t have to calculate the entire negotiation set, since all they need to know is the deal that maximizes utility and is best for them. Finding this deal has a complexity of $O(n^c)$, i.e., polynomial in the number of the agents.

- **Communication Complexity**

The complexity of the mechanism depends on the exact protocol we use. However, we can use one-step protocol in which the complexity is very low. (Each agent suggests a deal that maximizes the product of utilities, and one of these deals is chosen arbitrarily).

3.2.5 Distribution of the Mechanism

A PMM is defined to be symmetrically distributed (definition 14).

3.3 Incentive Compatible Mechanisms

In the previous sections we assumed complete knowledge of all relevant information. As mentioned, this is not always realistic in the real world. In order to overcome the lack of information, we use a “-1 negotiation phase” in which agents simultaneously declare their goals before the beginning of the negotiation. This raises the question of whether the PMM is an incentive compatible mechanism (i.e., the strategy of telling the truth is in equilibrium).

In this section, we provide a characterization of the relationship between kind of lies, domain attributes, and deal types. There are three kind of lies in TODs (hidden tasks, phantom tasks, decoy tasks), and we have considered three domain attributes (subadditivity, concavity, modularity) and three classes of PMMs, based on pure, all-or-nothing, and mixed deals. The resulting three-dimensional table is represented in Figure 8. Its notation is described below.

Modular	H	P	D
Pure	↙L_7	T	⇦T_4
A/N	↗T	↗T	↗T
Mixed	↙L_8	↗T	↗T

Concave	H	P	D
Pure	↙L↙	↙L_5⇨	↙L
A/N	↗T↗	T↗	T↗
Mixed	↙L↙	T ⇧↗	⇦T_3⇧↗

Subadditive	H	P	D
Pure	L↙	L↙	L↙
A/N	T_1↗	T/P	L_6 ⇩
Mixed	L↙	T/P_2⇧	L

Figure 8

Consider the entry under [Subadditive, A/N (All-or-Nothing deal), D (Decoy lie)]. The entry L at that position means that for every PMM that is based on all-or-nothing deals, there exists a subadditive domain and an encounter such that at least one agent has an incentive to lie with a decoy lie (L means that lying may be beneficial). The entry T at position [Concave, Mixed (Mixed deal), D (Decoy lie)] means that for every concave domain and every encounter within this domain, under any PMM based on mixed deals, agents do not have an incentive to lie with decoy lies (T means telling the truth is always beneficial).

The entries in the table marked T/P (such as [Subadditive, A/N (All-or-Nothing deal), P (Phantom lie)] refer to lies, which are not beneficial because they may always be discovered (in any encounter within the domain). If the agent tells the truth, it is because he is afraid of the penalty that we be levied is his lie is discovered. Thus, T/P can be transformed to T if the PMM includes a sufficiently high penalty for discovered lies.

In the table, there is relationship between cells. The fact that entry [Subadditive, A/N, H] is T implies that entry [Concave, A/N, H] will also be T (this is denoted by the ↗ single arrow in the first cell, and the ↗ arrow going into the T in cell [Concave, A/N, H]. Similarly, [Concave, A/N, H] being T implies that entry [Modular, A/N, H] will be T (and is also denoted by arrows). This is because modular domains are concave, and concave domains are subadditive; if there is no incentive to lie even in subadditive domains, there will certainly be no incentive to lie in concave or modular domains (which are sub-classes of the subadditive domains).

Similarly, the entry L in [Modular, Pure, H] implies that [Concave, Pure, H] will also have an L entry: if a beneficial lie can be found in a modular domain, then it can certainly be found in a concave domain (a superset). These downward influences of L are also marked in the table, with ↙ arrows.

There is also a relationship between certain table entries with the same domain attribute (these relationships are denoted by double arrows like ⇦). For example, if there is no incentive to lie in general mixed deals, there is no incentive to lie in all-or-nothing deals (which are a subset). Thus, the T in [Concave, Mixed, D] implies the T in [Concave, A/N, D] (it also implies [Concave, Mixed, P], which in turn implies [Concave, A/N, P],…).

To fill out the table, therefore, we need only demonstrate a small number of “fixed points”, which in turn implies all the other table entries. The fixed points that need to be demonstrated are numbered in the table from 1 to 8. We demonstrate the values for these 8 cells, and present some other theorems that make general statements about PMMs.

5.3.1. Incentive Compatible Fixed Points

The four incentive compatible fixed points are determined by the theorems below.

Theorem 12: [Fixed point 1]: For any encounter in multi-agent subadditive TOD, and any PMM over all-or-nothing deals, every “hiding” lie is not beneficial.

Proof:

Using theorems 9 and 11, we know that the agents will agree on a deal δ that satisfies

$$\forall i\; Utility_i(\delta) = \tfrac{1}{n} \times \left[\sum_{j=1}^{n} c(T_j) - c\left(\bigcup_{j=1}^{n} T_j \right) \right].$$

Negotiating over all-or-nothing deals means that they will agree on probabilities (p_1,…, p_n) that satisfy the following relationship:

$$\forall i\ Utility_i(\delta) = c(T_i) - p_i \times c\left(\bigcup_{j=1}^{n} T_j\right).$$

Thus,

$$\forall i\ c(T_i) - p_i \times c\left(\bigcup_{j=1}^{n} T_j\right) = \frac{1}{n} \times \left[\sum_{j=1}^{n} c(T_j) - c\left(\bigcup_{j=1}^{n} T_j\right)\right].$$

Particularly,

$$c(T_1) - p_1 \times c\left(\bigcup_{j=1}^{n} T_j\right) = \frac{1}{n} \times \left[\sum_{j=1}^{n} c(T_j) - c\left(\bigcup_{j=1}^{n} T_j\right)\right].$$

Thus,

$$p_1 = \frac{(n-1) \times c(T_1) - \sum_{j=2}^{n} c(T_j) + c\left(\bigcup_{j=1}^{n} T_j\right)}{n \times c\left(\bigcup_{j=1}^{n} T_j\right)}.$$

If agent A_1 lies and hides some tasks $L \subseteq T_1$ and broadcasts $T_1^* = T_1 - L$, then (because of the subadditivity of the domain) $c(T_1) \le c(T_1^*) + c(L)$ and $c\left(\bigcup_{i=1}^{n} T_i\right) \le c\left(T_1^* \cup \bigcup_{i=2}^{n} T_i\right) + c(L)$.

If the agents negotiate over T_1^*,T_2,…,T_n they will agree on:

$$p_1^* = \frac{(n-1) \times c(T_1^*) - \sum_{j=2}^{n} c(T_j) + c\left(T_1^* \cup \bigcup_{j=2}^{n} T_j\right)}{n \times c\left(T_1^* \cup \bigcup_{j=2}^{n} T_j\right)}.$$

Agent A_1 will perform $c\left(\bigcup_{i=1}^{n} T_i\right)$ with probability $p_1^* = \dfrac{(n-1) \times c(T_1^*) - \sum_{j=2}^{n} c(T_j) + c\left(T_1^* \cup \bigcup_{j=2}^{n} T_j\right)}{n \times c\left(T_1^* \cup \bigcup_{j=2}^{n} T_j\right)}$ or perform only L with probability $1 - p_1^* = \dfrac{\sum_{j=2}^{n} c(T_j) - (n-1) \times c(T_1^*) + (n-1) \times c\left(T_1^* \cup \bigcup_{j=2}^{n} T_j\right)}{n \times c\left(T_1^* \cup \bigcup_{j=2}^{n} T_j\right)}$, and his expected cost will be:

$$\frac{(n-1) \times c(T_1^*) - \sum_{j=2}^{n} c(T_j) + c\left(T_1^* \cup \bigcup_{j=2}^{n} T_j\right)}{n \times c\left(T_1^* \cup \bigcup_{j=2}^{n} T_j\right)} \times c\left(\bigcup_{j=1}^{n} T_j\right) + \frac{\sum_{j=2}^{n} c(T_j) - (n-1) \times c(T_1^*) + (n-1) \times c\left(T_1^* \cup \bigcup_{j=2}^{n} T_j\right)}{n \times c\left(T_1^* \cup \bigcup_{j=2}^{n} T_j\right)} \times c(L).$$

To complete the proof, it will be sufficient to show that

$$\frac{(n-1)\times c(T_1^*)-\sum_{j=2}^{n}c(T_j)+c\left(T_1^*\cup\bigcup_{j=2}^{n}T_j\right)}{n\times c\left(T_1^*\cup\bigcup_{j=2}^{n}T_j\right)}\times c\left(\bigcup_{j=1}^{n}T_j\right)+\frac{\sum_{j=2}^{n}c(T_j)-(n-1)\times c(T_1^*)+(n-1)\times c\left(T_1^*\cup\bigcup_{j=2}^{n}T_j\right)}{n\times c\left(T_1^*\cup\bigcup_{j=2}^{n}T_j\right)}\times c(L)\geq\frac{(n-1)\times c(T_1)-\sum_{j=2}^{n}c(T_j)+c\left(\bigcup_{j=1}^{n}T_j\right)}{n}$$

.

$$\left[(n-1)\times c(T_1^*)-\sum_{j=2}^{n}c(T_j)+c\left(T_1^*\cup\bigcup_{j=2}^{n}T_j\right)\right]\times c\left(\bigcup_{j=1}^{n}T_j\right)+\left[\sum_{j=2}^{n}c(T_j)-(n-1)\times c(T_1^*)+(n-1)\times c\left(T_1^*\cup\bigcup_{j=2}^{n}T_j\right)\right]\times c(L)\geq$$
$$\left[(n-1)\times c(T_1)-\sum_{j=2}^{n}c(T_j)+c\left(\bigcup_{j=1}^{n}T_j\right)\right]\times c\left(T_1^*\cup\bigcup_{j=2}^{n}T_j\right)$$

$$\left[(n-1)\times c(T_1^*)-\sum_{j=2}^{n}c(T_j)\right]\times c\left(\bigcup_{j=1}^{n}T_j\right)+\left[\sum_{j=2}^{n}c(T_j)-(n-1)\times c(T_1^*)+(n-1)\times c\left(T_1^*\cup\bigcup_{j=2}^{n}T_j\right)\right]\times c(L)\geq.$$
$$\left[(n-1)\times c(T_1)-\sum_{j=2}^{n}c(T_j)\right]\times c\left(T_1^*\cup\bigcup_{j=2}^{n}T_j\right)$$

There are two cases:

1. $c\left(\bigcup_{i=1}^{n}T_i\right)+c(T_1^*)\geq c\left(T_1^*\cup\bigcup_{i=2}^{n}T_i\right)+c(T_1)$

2. $c\left(\bigcup_{i=1}^{n}T_i\right)+c(T_1^*)< c\left(T_1^*\cup\bigcup_{i=2}^{n}T_i\right)+c(T_1)$

Case 1:

We know that $c\left(\bigcup_{i=1}^{n}T_i\right)\leq c\left(T_1^*\cup\bigcup_{i=2}^{n}T_i\right)+c(L)$ (subadditivity) which means that $c\left(\bigcup_{i=1}^{n}T_i\right)-c\left(T_1^*\cup\bigcup_{i=2}^{n}T_i\right)\leq c(L)$. It will be sufficient to show that:

$$\left[(n-1)\times c(T_1^*)-\sum_{j=2}^{n}c(T_j)\right]\times c\left(\bigcup_{j=1}^{n}T_j\right)+\left[\sum_{j=2}^{n}c(T_j)-(n-1)\times c(T_1^*)+(n-1)\times c\left(T_1^*\cup\bigcup_{j=2}^{n}T_j\right)\right]\times\left[c\left(\bigcup_{j=1}^{n}T_j\right)-c\left(T_1^*\cup\bigcup_{j=2}^{n}T_j\right)\right]\geq$$
$$\left[(n-1)\times c(T_1)-\sum_{j=2}^{n}c(T_j)\right]\times c\left(T_1^*\cup\bigcup_{j=2}^{n}T_j\right)$$

$$\left[(n-1)\times c(T_1^*)-\sum_{j=2}^{n}c(T_j)+\sum_{j=2}^{n}c(T_j)-(n-1)\times c(T_1^*)+(n-1)\times c\left(T_1^*\cup\bigcup_{j=2}^{n}T_j\right)\right]\times c\left(\bigcup_{j=1}^{n}T_j\right)-$$
$$\left[\sum_{j=2}^{n}c(T_j)-(n-1)\times c(T_1^*)+(n-1)\times c\left(T_1^*\cup\bigcup_{j=2}^{n}T_j\right)\right]\times c\left(T_1^*\cup\bigcup_{j=2}^{n}T_j\right)\geq$$
$$\left[(n-1)\times c(T_1)-\sum_{j=2}^{n}c(T_j)\right]\times c\left(T_1^*\cup\bigcup_{j=2}^{n}T_j\right)$$

$$(n-1)\times c\left(T_1^*\cup\bigcup_{j=2}^{n}T_j\right)\times c\left(\bigcup_{j=1}^{n}T_j\right)-\left[\sum_{j=2}^{n}c(T_j)-(n-1)\times c(T_1^*)+(n-1)\times c\left(T_1^*\cup\bigcup_{j=2}^{n}T_j\right)\right]\times c\left(T_1^*\cup\bigcup_{j=2}^{n}T_j\right)\geq$$
$$\left[(n-1)\times c(T_1)-\sum_{j=2}^{n}c(T_j)\right]\times c\left(T_1^*\cup\bigcup_{j=2}^{n}T_j\right)$$

$$(n-1)\times c\left(\bigcup_{j=1}^{n}T_j\right)-\left[\sum_{j=2}^{n}c(T_j)-(n-1)\times c(T_1^*)+(n-1)\times c\left(T_1^*\cup\bigcup_{j=2}^{n}T_j\right)\right]\geq(n-1)\times c(T_1)-\sum_{j=2}^{n}c(T_j)$$

$$(n-1)\times c\left(\bigcup_{j=1}^{n}T_j\right)-\sum_{j=2}^{n}c(T_j)+(n-1)\times c(T_1^*)-(n-1)\times c\left(T_1^*\cup\bigcup_{j=2}^{n}T_j\right)\geq(n-1)\times c(T_1)-\sum_{j=2}^{n}c(T_j)$$

$$(n-1)\times c\left(\bigcup_{j=1}^{n}T_j\right)+(n-1)\times c(T_1^*)-(n-1)\times c\left(T_1^*\cup\bigcup_{j=2}^{n}T_j\right)\geq(n-1)\times c(T_1)$$

$$c\left(\bigcup_{j=1}^{n}T_j\right)+c(T_1^*)-c\left(T_1^*\cup\bigcup_{j=2}^{n}T_j\right)\geq c(T_1)$$

$$c\left(\bigcup_{j=1}^{n}T_j\right)+c(T_1^*)\geq c(T_1)+c\left(T_1^*\cup\bigcup_{j=2}^{n}T_j\right)$$

Case 2:
$c\left(\bigcup_{i=1}^{n}T_i\right)+c(T_1^*)<c\left(T_1^*\cup\bigcup_{i=2}^{n}T_i\right)+c(T_1)$ means that $c\left(\bigcup_{i=1}^{n}T_i\right)+c(T_1^*)-c(T_1)<c\left(T_1^*\cup\bigcup_{i=2}^{n}T_i\right)$.

It will be sufficient to show that:

$$\left[(n-1)\times c(T_1^*)-\sum_{j=2}^{n}c(T_j)\right]\times c\left(\bigcup_{j=1}^{n}T_j\right)+\left[\sum_{j=2}^{n}c(T_j)+(n-1)\times\left\{c\left(\bigcup_{j=1}^{n}T_j\right)-c(T_1)\right\}\right]\times c(L)\geq$$
$$\left[(n-1)\times c(T_1)-\sum_{j=2}^{n}c(T_j)\right]\times c\left(T_1^*\cup\bigcup_{j=2}^{n}T_j\right)$$

$$\left[(n-1)\times c(T_1^*)-\sum_{j=2}^{n}c(T_j)\right]\times c\left(\bigcup_{j=1}^{n}T_j\right)+\left[\sum_{j=2}^{n}c(T_j)-(n-1)\times c(T_1^*)+(n-1)\times\left\{c\left(\bigcup_{j=1}^{n}T_j\right)+c(T_1^*)-c(T_1)\right\}\right]\times c(L)\geq$$
$$\left[(n-1)\times c(T_1)-\sum_{j=2}^{n}c(T_j)\right]\times c\left(T_1^*\cup\bigcup_{j=2}^{n}T_j\right)$$

$$\left[(n-1)\times c(T_1^*)-\sum_{j=2}^{n}c(T_j)\right]\times c\left(\bigcup_{j=1}^{n}T_j\right)+c(L)\times(n-1)\times c\left(\bigcup_{j=1}^{n}T_j\right)+c(L)\times\left[\sum_{j=2}^{n}c(T_j)-(n-1)\times c(T_1)\right]\geq$$
$$\left[(n-1)\times c(T_1)-\sum_{j=2}^{n}c(T_j)\right]\times c\left(T_1^*\cup\bigcup_{j=2}^{n}T_j\right)$$

$$\left[(n-1)\times c(T_1^*)-\sum_{j=2}^{n}c(T_j)+c(L)\times(n-1)\right]\times c\left(\bigcup_{j=1}^{n}T_j\right)+c(L)\times\left[\sum_{j=2}^{n}c(T_j)-(n-1)\times c(T_1)\right]\geq$$
$$\left[(n-1)\times c(T_1)-\sum_{j=2}^{n}c(T_j)\right]\times c\left(T_1^*\cup\bigcup_{j=2}^{n}T_j\right)$$

$$\left[(n-1)\times c(T_1^*)-\sum_{j=2}^{n}c(T_j)+c(L)\times(n-1)\right]\times c\left(\bigcup_{j=1}^{n}T_j\right)+\left[c(L)+c\left(T_1^*\cup\bigcup_{j=2}^{n}T_j\right)\right]\times\left[\sum_{j=2}^{n}c(T_j)-(n-1)\times c(T_1)\right]\geq 0$$

We know that $c\left(\bigcup_{i=1}^{n} T_i\right) \le c\left(T_1^* \cup \bigcup_{i=2}^{n} T_i\right) + c(L)$. It will be sufficient to show that:

$$\left[(n-1)\times c(T_1^*) - \sum_{j=2}^{n} c(T_j) + c(L)\times(n-1)\right]\times c\left(\bigcup_{j=1}^{n} T_j\right) + c\left(\bigcup_{j=1}^{n} T_j\right)\times\left[\sum_{j=2}^{n} c(T_j) - (n-1)\times c(T_1)\right] \ge 0$$

$$(n-1)\times c(T_1^*) - \sum_{j=2}^{n} c(T_j) + c(L)\times(n-1) + \sum_{j=2}^{n} c(T_j) - (n-1)\times c(T_1) \ge 0$$

$$(n-1)\times c(T_1^*) + c(L)\times(n-1) - (n-1)\times c(T_1) \ge 0$$

$$c(T_1^*) + c(L) - c(T_1) \ge 0$$

$$c(T_1^*) + c(L) \ge c(T_1)$$

Which is true.

<u>Theorem 13</u>: [equivalent all-or-nothing]: For any encounter in a multi-agent subadditive TOD, there is always an all-or-nothing deal in NS maximizing the product of the utilities.

<u>Proof</u>:

Assume that $\delta = (D_1, D_2, \ldots, D_n): (p_1, p_2, \ldots, p_n)$ is in the negotiation set. If we use the same all-or-nothing deal $\delta^* = (D_1 \cup D_2 \cup \ldots \cup D_n, \phi, \phi, \ldots, \phi): (q_1, q_2, \ldots, q_n)$ that we use in the proof of theorem 7, we can show that $\delta \ge \delta'$ using the same steps as in the proof of theorem 7, by changing the "<" symbol to "≤". Because δ is in the negotiation set, it cannot be dominated, and therefore $\delta \equiv \delta'$.

Since an all-or-nothing deal is always a candidate agreement, the negotiation mechanism might arbitrarily choose it. This is the reason that the L in [Subadditive, Mixed, D] is implied by the L in [Subadditive, A/N, D]. If there exists an encounter in which one of the agents has an incentive to lie when the negotiation is over all-or-nothing deals, then in the same encounter the same agents have the same incentive to lie when negotiating over general mixed deals: the agreement can always be an all-or-nothing deal.

Another consequence is the following theorem:

<u>Theorem 14</u>:[Fixed point 2] For any encounter in a multi-agent subadditive TOD, and any PMM over mixed deals, every "phantom" lie has a positive probability of being discovered. Therefore, with sufficiently severe penalty mechanism, telling the truth is the optimal strategy.

<u>Proof:</u>

It follows from theorem 13 that the agents can always agree on some all-or-nothing deal. Therefore, there is a positive probability that one of the non-lying agents will be assigned to execute all tasks (including the phantom tasks). In this case the phantom lie would be revealed.

<u>Theorem 15</u>: [Fixed point 3]: For any encounter in a multi-agent concave TOD, and any PMM over mixed deals, every "decoy" lie is not beneficial.

<u>Proof</u>:

Let $(T_1, T_2, \ldots, T_n)$ be the true encounter. From theorems 8 and 11 follows that, negotiation using the true information and using any PMM over mixed deals will lead to an agreement on a mixed deal δ that satisfies

$$\forall i\, Utility_i(\delta) = \tfrac{1}{n} \times \left[\sum_{j=1}^{n} c(T_j) - c\left(\bigcup_{j=1}^{n} T_j \right) \right].$$

Assume that agent A_1 lies and declares that his initial set of tasks is $T_1^* = T_1 \cup L$, where L is a set of decoy tasks. The negotiation that assumes this information is true will lead to an agreement on a mixed deal δ^* that satisfies

$$\forall i\, Utility_i(\delta^*) = \frac{1}{n} \times \left[c(T_1^*) + \sum_{j=2}^{n} c(T_j) - c\left(T_1^* \cup \bigcup_{j=2}^{n} T_j \right) \right].$$

Because $T_1^* \subseteq T_1^* \cup \bigcup_{i=2}^{n} T_i$, we can use the definition of concavity and conclude that

$$c\left(T_1^* \cup \bigcup_{i=2}^{n} T_i \right) - c\left(\bigcup_{i=1}^{n} T_i \right) \le c(T_1^*) - c(T_1).$$

This also means that $c(T_1) - c\left(\bigcup_{i=1}^{n} T_i \right) \le c(T_1^*) - c\left(T_1^* \cup \bigcup_{i=2}^{n} T_i \right)$. We can thus conclude that

$$\forall i \ne 1\; Utility_i(\delta^*) = \frac{1}{n} \times \left[c(T_1^*) + \sum_{j=2}^{n} c(T_j) - c\left(T_1^* \cup \bigcup_{j=2}^{n} T_j \right) \right] \ge \frac{1}{n} \times \left[\sum_{j=1}^{n} c(T_j) - c\left(\bigcup_{j=1}^{n} T_j \right) \right] = Utility_i(\delta).$$

In other words, even though A_i ($i \ne 1$) might have to perform some of the decoy tasks, his utility can only be larger than if agent A_1 would have told the truth.

The agent will agree on a mixed deal $\delta^* = (D_1^*, \ldots, D_n^*) : (p_1^*, \ldots, p_n^*)$ such that $\bigcup_{i=1}^{n} D_i^* = T_1^* \cup \bigcup_{i=2}^{n} T_i$.

Let $D_i = D_i^* \setminus L$ for all i.

Agent A_1 does *not* have to perform his decoy tasks, and therefore his actual utility from δ^* (which we denote by $\overline{Utility_1(\delta^*)}$) is $c(T_1) - \sum_{j=1}^{n} p_j^* \times c\left(D_{(1-j) \bmod n+1}\right)$.

On the other hand, the other agents *do* have to perform the decoy tasks. Therefore $\forall i \ne 1\, Utility_i(\delta^*) = c(T_i) - \sum_{j=1}^{n} p_j^* \times c\left(D^*_{(i-j) \bmod n+1}\right)$. We can sum those utilities.

$$\overline{Utility_1(\delta^*)}+\sum_{i=2}^{n}Utility_i(\delta^*)=c(T_1)-\sum_{j=1}^{n}p_j^*\times c\left(D_{(1-j)\bmod n+1}\right)+\sum_{i=2}^{n}\left[c(T_i)-\sum_{j=1}^{n}p_j^*\times c\left(D^*_{(i-j)\bmod n+1}\right)\right]=$$
$$\sum_{i=1}^{n}c(T_i)-\sum_{j=1}^{n}p_j^*\times\left[c\left(D_{(1-j)\bmod n+1}\right)+\sum_{i=2}^{n}c\left(D^*_{(i-j)\bmod n+1}\right)\right]=\sum_{i=1}^{n}c(T_i)-\sum_{k=1}^{n}\left[p_k^*\times c(D_k)+\sum_{l\neq k}p_l^*\times c\left(D_k^*\right)\right]\leq$$
$$\sum_{i=1}^{n}c(T_i)-\sum_{k=1}^{n}\sum_{l=1}^{n}p_l^*\times c(D_k)=\sum_{i=1}^{n}c(T_i)-\sum_{k=1}^{n}c(D_k)=\sum_{i=1}^{n}c(T_i)-c\left(\bigcup_{i=1}^{n}D_i\right)=\sum_{i=1}^{n}c(T_i)-c\left(\bigcup_{i=1}^{n}T_i\right)=$$
$$\sum_{i=1}^{n}Utility_i(\delta)$$

We have shown that $\overline{Utility_1(\delta^*)}+\sum_{i=2}^{n}Utility_i(\delta^*)\leq\sum_{i=1}^{n}Utility_i(\delta)$. However, we have already shown that $\forall i\neq 1\ Utility(\delta^*)\geq Utility(\delta)$. Thus, the conclusion is that agent A_1's actual utility cannot be bigger than his unvarnished utility. Therefore, the lie is not beneficial to agent A_1.

Theorem 16: For any encounter in a multi-agent concave TOD, and any PMM over mixed deals, every lie (including combinations of hidden, phantom and decoys) is not beneficial.

Proof:

Let (T_1, T_2,…, T_n) be the true encounter, and let the agreement that would be reached using a PMM with common knowledge of the true information be the mixed deal δ.

Assume that agent A_1 declares that his set of tasks is $T_1^*=\{T_1-T_h\cup T_d\}$ where T_h is the set of all his hidden tasks, and T_d is the set of all his decoy tasks. Let the agreement that the agents will reach when they assume this information is true be δ^*.

Consider the negotiation where agent A_1 declares his set of tasks to be $T_1^{**}=\{T_1-T_h\}$. The negotiation that assumes this information is true will lead to an agreement on a mixed deal δ^{**} that satisfies:

$$\forall i\, Utility_i(\delta^{**})=\frac{1}{n}\times\left[c(T_1^{**})+\sum_{j=2}^{n}c(T_j)-c\left(T_1^{**}\cup\bigcup_{j=2}^{n}T_j\right)\right].$$

Because $T_1^{**}\subseteq T_1^{**}\cup\bigcup_{i=2}^{n}T_i$, we can use the definition of concavity and conclude that

$$c\left(\bigcup_{i=1}^{n}T_i\right)-c\left(T_1^{**}\cup\bigcup_{i=2}^{n}T_i\right)\leq c(T_1)-c(T_1^{**}).$$

This also means that $c(T_1^{**})-c\left(T_1^{**}\cup\bigcup_{i=2}^{n}T_i\right)\leq c(T_1)-c\left(\bigcup_{i=1}^{n}T_i\right)$.

So we can conclude that

$$Utility_1(\delta^{**})=\frac{1}{n}\times\left[c(T_1^{**})+\sum_{j=2}^{n}c(T_j)-c\left(T_1^{**}\cup\bigcup_{j=2}^{n}T_j\right)\right]\leq\frac{1}{n}\times\left[\sum_{j=1}^{n}c(T_j)-c\left(\bigcup_{j=1}^{n}T_j\right)\right]=Utility_1(\delta).$$

In other words, the apparent utility of agent A_1 is less than is unvarnished utility without hidden tasks.

Let us now consider the situation where agent A_1 declares $T_1^* = \{T_1^{**} \cup T_d\}$. From theorem 15 we can see that $\overline{Utility_1(\delta^*)} \le Utility_1(\delta^{**})$, and the final conclusion will be that $\overline{Utility_1(\delta^*)} \le Utility_1(\delta)$. The liar's actual utility will be less than his unvarnished utility.

Because of the theorem above, it is clear that for concave domains, agents cannot benefit by lying when all-or-nothing deals are in use – i.e., any PMM over all-or-nothing deals is incentive compatible. This is also true in modular domains (a subcase). This can be seen in the table, where the entire all-or-nothing row is marked T for modular and concave domains.

Additionally, in a subadditive domain where decoy lies cannot be generated, a PMM over all-or-nothing deals with penalty for discovered lies is incentive compatible. This can be seen in the table, where the all-or-nothing row is marked T and T/P (excluding the decoy column).

<u>Theorem 17</u>: [Fixed point 4]: For any encounter in a multi-agent modular TOD, and any PMM over pure deals, every "decoy" lie is not beneficial.

<u>Proof</u>:

Let $(T_1, T_2, \ldots, T_n)$ be an encounter, and let the agreement that would be reached using a PMM with common knowledge of the true information be the mixed deal δ.

$\sum_{i=1}^{n} Utility_i(\delta) = \sum_{i=1}^{n} [c(T_i) - c(D_i)] = \sum_{i=1}^{n} c(T_i) - \sum_{i=1}^{n} c(D_i)$. Since the TOD is modular and $\delta \in NS$, we know from theorem 4 that $\sum_{i=1}^{n} c(D_i) = c\left(\bigcup_{i=1}^{n} D_i\right) = c\left(\bigcup_{i=1}^{n} T_i\right)$. This means that $\sum_{i=1}^{n} Utility_i(\delta) = \sum_{i=1}^{n} c(T_i) - c\left(\bigcup_{i=1}^{n} T_i\right)$.

The agents will agree on a deal that maximizes the product of the utilities (from the definition of a PMM). Since all pure deals have a constant sum of utility (which is $\Sigma c(T_i) - c(\cup T_i)$), the deals that maximize the product are those that have the smallest difference between the agents utilities (i.e., deals in which the maximal difference between two agents utilities is the smallest). Therefore, there is either one pure deal, or n! pure deals (which are symmetric in the division of the utility), that maximize the product. In the former case, the unique pure deal gives all agents equal utility. In the latter case, one of the n! pure deals will be chosen arbitrarily (each with probability 1/n!). All agents will have the same expected utility ($[\Sigma c(T_i) - c(\cup T_i)]/n$).

Assume that agent A_1 lies and creates a decoy task $d \in T$. His apparent initial set of tasks is $T_1^* = T_1 \cup \{d\}$. Since $d \notin T_1$ (otherwise it was not a decoy task) and the TOD is modular, $c(T_1^*) = c(T_1 \cup \{d\}) = c(T_1) + c(\{d\}) - c(T_1 \cap \{d\}) = c(T_1) + c(\{d\})$.

For any pure deal $\delta = (D_1, D_2, \ldots, D_n)$ the apparent utility for agent A_1 from the deal is: $c(T_1^*) - c(D_1)$. The expected value of the apparent utility for agent A_1 is $\frac{1}{n} \times \left\langle c(T_1^*) + \sum_{i=2}^{n} c(T_i) - c\left(T_1^* \cup \bigcup_{i=2}^{n} T_i\right)\right\rangle$.

If agent A_1 was honest is expected utility was $\frac{1}{n} \times \left\langle \sum_{i=1}^{n} c(T_1) - c\left(\bigcup_{i=1}^{n} T_i\right)\right\rangle$.

There are two cases:

1) $d \in \bigcup_{i=2}^{n} T_i$

2) $d \notin \bigcup_{i=2}^{n} T_i$

<u>Case 1</u> ($d \in \bigcup_{i=2}^{n} T_i$)<u>:</u>

In this case,

$$c(T_1^*)+\sum_{i=2}^{n}c(T_i)-c\left(T_1^* \cup \bigcup_{i=2}^{n} T_i\right)=c(T_1)+c(\{d\})+\sum_{i=2}^{n}c(T_i)-c\left(T_1^* \cup \bigcup_{i=2}^{n} T_i\right)=c(\{d\})+\sum_{i=1}^{n}c(T_i)-c\left(T_1^* \cup \bigcup_{i=2}^{n} T_i\right)=$$
$$c(\{d\})+\sum_{i=1}^{n}c(T_i)-c\left(T_1 \cup \{d\} \cup \bigcup_{i=2}^{n} T_i\right)=c(\{d\})+\sum_{i=1}^{n}c(T_i)-c\left(T_1 \cup \bigcup_{i=2}^{n} T_i\right)=c(\{d\})+\sum_{i=1}^{n}c(T_i)-c\left(\bigcup_{i=1}^{n} T_i\right)$$

Therefore, the apparent utility of lying agent A_1 is higher than the utility he would have if he were telling the truth by c({d})/n.

<u>Case 2</u> ($d \notin \bigcup_{i=2}^{n} T_i$)<u>:</u>

In this case,

$$c(T_1^*)+\sum_{i=2}^{n}c(T_i)-c\left(T_1^* \cup \bigcup_{i=2}^{n} T_i\right)=c(T_1)+c(\{d\})+\sum_{i=2}^{n}c(T_i)-c\left(T_1^* \cup \bigcup_{i=2}^{n} T_i\right)=c(\{d\})+\sum_{i=1}^{n}c(T_i)-c\left(T_1^* \cup \bigcup_{i=2}^{n} T_i\right)=$$
$$c(\{d\})+\sum_{i=1}^{n}c(T_i)-c\left(T_1 \cup \{d\} \cup \bigcup_{i=2}^{n} T_i\right)=c(\{d\})+\sum_{i=1}^{n}c(T_i)-c\left(\{d\} \cup \bigcup_{i=1}^{n} T_i\right)=c(\{d\})+\sum_{i=1}^{n}c(T_i)-\left[c\{d\}+c\left(\bigcup_{i=1}^{n} T_i\right)\right]=$$
$$\sum_{i=1}^{n}c(T_i)-c\left(\bigcup_{i=1}^{n} T_i\right)$$

Therefore, the apparent utility of lying agent A_1 is equal to the utility he would have if he were telling the truth.

The actual utility that agent A_1 gets from δ depends on whether the decoy task d is in D_1 or not.

1) $d \in D_1$

If $d \in \bigcup_{i=2}^{n} T_i$ (case 1) then A_1's actual utility from δ is:

$$c(T_1)-c(D_1)=c(T_1)+c(\{d\})-c(D_1)-c(\{d\})=c(T_1^*)-c(D_1)-c(\{d\})$$

I.e., A_1's actual utility from δ is lower than its apparent utility by c({d}). Therefore, agent A_1's expected actual utility is $\frac{1}{n}\times\left\langle\sum_{i=1}^{n}c(T_1)-c\left(\bigcup_{i=1}^{n}T_i\right)-(n-1)\times c\{d\}\right\rangle$. He would have gotten more declaring his true set of tasks.

If $d \notin \bigcup_{i=2}^{n} T_i$ (case 2) then A_1's actual utility from δ is

$$c(T_1)-[c(D_1)-c(\{d\})]=c(T_1)-c(D_1)+c(\{d\})=c(T_1^*)-c(\{d\})-c(D_1)+c(\{d\})=c(T_1^*)-c(D_1)$$

I.e., A_1's actual utility from δ is equal to its apparent utility from δ. Therefore, the expected value of A1's actual utility is same as the expected utility he would have if he were telling the truth.

2) $d \notin D_1$

In this case A_1's actual utility from δ is:

$$c(T_1)-c(D_1)=c(T_1)+c(\{d\})-c(D_1)-c(\{d\})=c(T_1^*)-c(D_1)-c(\{d\}).$$

I.e., A_1's actual utility from δ is lower than its apparent utility by c({d}). Therefore, the expected value of A_1's actual utility not higher than $\frac{1}{n}\times\left\langle\sum_{i=1}^{n}c(T_1)-c\left(\bigcup_{i=1}^{n}T_i\right)-(n-1)\times c\{d\}\right\rangle$. He would have gotten more declaring his true set of tasks.

$\Rightarrow$ The expected actual utility of agent A_1 is not higher than the expected utility he would have if he were telling the truth. Therefore, lying is not beneficial for A_1.

5.3.2. Non-Incentive Compatible Fixed Points

<u>Fixed point 5</u>: For an example of a beneficial phantom lie in a concave domain using a PMM over pure deals, consider the following example in the Postman Domain restricted to graphs that have a tree topology (which is concave):

Consider the graph given in figure 9. The length, and thus the cost, of each edge is written next to it. The post office is at the root of the tree. There are three agents. All of them need to deliver letters to nodes A, B and C.

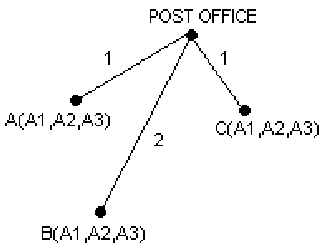

Figure 9

If the agents are using some PMM over pure deals, the protocol chooses with equal probability between the product maximizing deals.

Each agent has 0.33 change of delivering the letters to A (Utility = 6), 0.33 change of delivering the letters to B (Utility = 4) and 0.33 of delivering the letters to C (Utility = 6). The expected utility for each agent is 5.33.

What happens if agent 1 declares a phantom letter, and tells the other agents that he has another letter to deliver to node D? See figure 10.

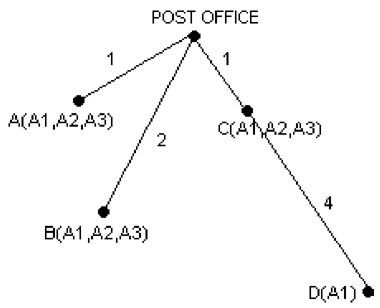

Figure 10

The cost for A_1 of delivering his letters plus the phantom letter is now 16. It would not be rational for agent 2 or agent 3 to visit D. Agent 1 will thus have to visit D, and he could deliver A_2 and A_3 to C on his way. So they will agree on a deal where agent 1 delivers the letters to C and D (with apparent utility of 6, and actual utility of 6). Thus, agent 1's utility has risen from 5.33 to 6 by creating this phantom letter. This lie is also a "safe" lie, since A_2 and A_3 cannot verify whether the phantom letter was actually delivered.

Fixed point 6: For an example for a beneficial decoy lie in a subadditive domain (e.g., the Postmen Domain with an unrestricted graph topology) using a PMM over all-or-nothing deals, consider the following example:

Let the graph be as in Figure 11. Every edge between nodes has cost 1.

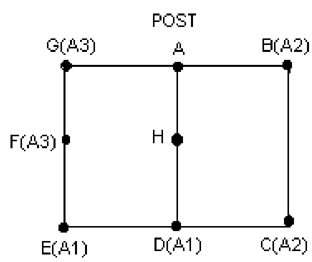

Figure 11

The post office is at node A. agent 1 must deliver letters to nodes D and E, agent 2 must deliver letters to nodes B and C, and agent 3 must deliver letters to nodes F and G.

The cost for agent 1of delivering his letters alone is 6 and the cost for agent 2 and 3 is 4 (each). One agent can deliver the letters of agents 1 and 2 in the cost of 7. The same cost is required when one agent delivers the letters of agents 2 and 3. The letters of agents 1 and 3 can be delivered at cost of 6 and all the letters (of all three agents) can be delivered at a total cost of 7.

If agent A_1 tells the truth and the agents are using some PMM, they will agree on the all-or-nothing deal in which A_1 has 11/21 change of delivering all the letters, and each of the other agents (A_2 and A_3) has 5/21 change of delivering all the letters. The expected utility for each agent is 2.33.

If A_1 lies with a producible decoy letter to node H, his apparent cost is still 6. Delivery of the entire set of letters, however, now (apparently) costs 9. If the agents use some PMM over all-or-nothing deals, they will agree on deal that gives all agents apparent utility of 1.66. Agents A_2 and A_3 will do the entire delivery, including the decoy letter, with probability 7/27, in fact getting utility of 1.66. Agent A_1, however, will simply do the cycle (i.e., without the decoy letter), with probability 13/27, getting an actual utility of 2.62. Lying is thus beneficial to A_1: telling the truth gives him a real utility of 2.33, while lying gives him a real utility of 2.62.

Fixed Point 7: For an example for a beneficial hidden lie in a modular using a PMM over pure deals, consider the following example:

The example is from the Postman Domain restricted to graphs that have a star topology. Consider the graph given in the figure 12. The length, and thus the cost, of each edge is written next to it. The post office is at the root of the tree. There are three agents. All of them need to deliver letters to nodes A, B and C.

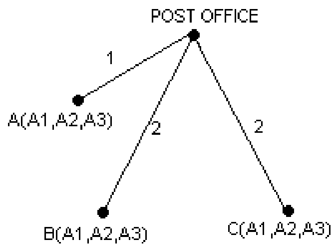

Figure 12

Note that $c(T_1) = c(T_2) = c(T_3) = c(T_1, T_2) = c(T_1, T_3) = c(T_2, T_3) = c(T_1, T_2, T_3) = 10$.

If the agents are using some PMM over pure deals, the protocol chooses with equal probability between the product maximizing deals. Each agent has 0.33 change of delivering the letters to A (Utility = 8), 0.33 change of delivering the letters to B (Utility = 6) and 0.33 of delivering the letters to C

(Utility = 6). The expected utility for each agent is 6.66.

What happens if agent 1 "hides" the letter he has to take to A, and tells the other agents that he only has to take letters to B and C?

Then the only pure deals that maximize the product of the agents' utilities are those which send agent A_1 to node A. The agreed-upon deal gives agent A_2 and A_3 a real utility of 6. It gives A_1 an apparent utility of 6, but an actual utility of 8; while visiting node A ahent1 can, at no extra cost, deliver his hidden letter.

Therefore, agent 1, by hiding one of his letters, has lowered the apparent cost of his goal. This ends up benefiting the lying agent.

<u>Fixed Point 8</u>: For an example for a beneficial hidden lie in a modular domain (e.g., the Postmen Domain restricted to graphs that have a star topology) using a PMM over mixed deals, consider again the encounter presented in figure 12.

If all the agents reveal their true sets of tasks and use some PMM over mixed deals, they will agree on a mixed deal that maximizes the product of their utilities. According to theorems 8 and 11, such a deal will give each agent utility of:

$$\tfrac{1}{3}\times\left[c(T_1)+c(T_2)+c(T_3)-c(T_1\cup T_2\cup T_3)\right]=\tfrac{1}{3}\times\left[10+10+10-10\right]=\tfrac{20}{3}=6\tfrac{2}{3}$$

What happens if agent 1 "hides" the letter he has to take to A, and tells the other agents that he only has to take letters to B and C?

If the agents use some PMM over mixed deals, there are six deals that maximize the product of the (apparent) utility. Therefore, there are only six deals they can agree on, and one of those deals will be chosen with equal probability.

One of the following six possibilities will be chosen.

1) The mixed deal ({A},{B},{C}):(1,0,0). The apparent utility for agent 1 is 6, which is also the real utility for agents 2 and 3. However, A_1's real utility is 10-2=8, which is greater than 6.66, the utility he would have if he had told the truth. Therefore, this deal is beneficial to agent 1.

2) The mixed deal ({A},{C},{B}):(1,0,0). This deal is symmetric to the previous one. Therefore, A_1's real utility is 10-2=8, which is greater than 6.66, the utility he would have if he had told the truth. Therefore, this deal is beneficial to agent 1.

3) The mixed deal ({A,B,C},ϕ,ϕ):(0.2,0.4,0.4). This deal gives agent 1 apparent expected utility of 6 and agents 2 and 3 (real) expected utility of 6. This deal however gives agent 1 a real utility of

   10-0.2*10-0.6*2=6.4<6.66. When agent 1 wins the coin toss (with probability 0.8), he still has to deliver his hidden letter. Therefore, this deal is not beneficial to agent 1.

4) The mixed deal ({A,B},{C},ϕ):(1/7,2/7,4/7). This deal gives agent 1 apparent expected utility of 6 and agents 2 and 3 (real) expected utility of 6. This deal, however gives agent 1 a real utility of 10-1/7*6-2/7*4-6/7*2=6.285<6.66. Therefore, this deal is not beneficial to agent 1.

5) The mixed deal ({A,C},{B},ϕ):(1/7,2/7,4/7). This deal is symmetric to the previous one. Therefore, A_1's real utility is 6.285. . Therefore, this deal is not beneficial to agent 1.

6) The mixed deal ({A},{B,C},ϕ):(5/13,2/13,6/13). This deal gives agent 1 apparent expected utility of 6 and agents 2 and 3 (real) expected utility of 6. This deal, however gives agent 1 a real utility of 10-5/13*2-2/13*8-8/13*2=6.79. There, this deal is beneficial to agent 1.

Since one of the above deals will be chosen with equal probability, the expected payoff for the lying agent is 1/6*8+1/6*8+1/6*6.4+1/6*6.285+1/6*6.285+1/6*6.79=6.959. Therefore, even when the negotiation mechanism sometimes chooses deals which are not beneficial to agent 1, the lying agent still has an expected benefit, having gotten expected utility of 6.959 instead of the 6.66 he would have gotten.

4 Shapley Value-based Mechanism

Various *negotiation protocols* are possible in encounters among agents. In the previous section weintroduced the Product Maximizing Mechanisms (PMMs). In this section we introduce another mechanism, which is called the Shapley Value-based Mechanism. This mechanism was already introduced by Zlotkin and Rosenschein [32] for the case of full information. Here, we extend their work for cases where agents have only partial information about the goals of other group members and can deceive one another about this private information.

4.1 Definition of the Mechanism

The Shapley Value [13,15] for agent i is a weighted average of all the utilities that i contributes to all possible coalitions. The weight of each coalition is the probability that this coalition will be formed in a random process that starts with the one-agent coalition, and in which the coalition grows by one agent at a time such that each agent that joins the coalition is credited with his contribution to the coalition. The Shapley value is actually the expected utility that the agent will have from such a random coalition (assuming any coalition and permutation is equally likely).

In definition 15, we give a formula for the Shapley value that uses the notation of TODs.

Definition 15: The Shapley Value is: $u_i = c(T_i) - \sum_{S \subset N, i \notin S} \frac{(n-|S|-1)!|S|!}{n!}\left[c(S \cup \{i\}) - c(S)\right]$.

Following [32], we can define a Shapley Value-based mechanism for subadditive TODs that forms the full coalition and divides the value of the full coalition using the Shapley value. The mechanism simply chooses an all-or-nothing deal ($p_1,\ldots,p_n$), such that:

$$p_i = \frac{\sum_{S \subset N, i \notin S} \frac{(n-|S|-1)!|S|!}{n!}\left[c(S \cup \{i\}) - c(S)\right]}{c(N)}$$

Theorem 18: The above all-or-nothing deal is well defined, (i.e., $\forall i \in N$: $0 \le p_i \le 1$; $\Sigma p_i = 1$) and gives each agent i an accepted utility that is exactly the Shapley Value u_i.

Proof:

- Because of the monotonicity of c, $\forall S \forall i$ $c(S) \le c(S \cup \{i\})$. This also means that $0 \le c(S \cup \{i\}) - c(S)$. Thus, $\forall i$ $0 \le p_i$.
- From the subadditivity of c, $\forall S \forall i$ $c(S \cup \{i\}) \le c(S) + c(\{i\})$. This also means that $c(S \cup \{i\}) - c(S) \le c(\{i\})$. Therefore,

$$\forall i\ p_i = \frac{\sum_{S \subset N, i \notin S} \frac{(n-|S|-1)!|S|!}{n!}\left[c(S \cup \{i\}) - c(S)\right]}{c(N)} \le \frac{\sum_{S \subset N, i \notin S} \frac{(n-|S|-1)!|S|!}{n!} \times c(\{i\})}{c(N)}$$

$$= \frac{c(\{i\})}{c(N)} \times \sum_{S \subset N, i \notin S} \frac{(n-|S|-1)!|S|!}{n!} = \frac{c(\{i\})}{c(N)} \le 1$$

- Moreover,

$$\sum_{i=1}^{n} p_i = \sum_{i=1}^{n} \frac{\sum_{S\subset N, i\notin S} \frac{(n-|S|-1)!|S|!}{n!}\left[c(S\cup\{i\})-c(S)\right]}{c(N)} =$$

$$\frac{1}{n!c(N)} \times \sum_{i=1}^{n} \sum_{S\subset N, i\notin S} \frac{(n-|S|-1)!|S|!}{n!}\left[c(S\cup\{i\})-c(S)\right] =$$

$$\frac{1}{n!c(N)} \times \left[\sum_{i=1}^{n} \sum_{S\subset N, i\notin S} \frac{(n-|S|-1)!|S|!}{n!} \times c(S\cup\{i\}) - \sum_{i=1}^{n} \sum_{S\subset N, i\notin S} \frac{(n-|S|-1)!|S|!}{n!} \times c(S)\right] =$$

$$\frac{1}{n!c(N)} \times \left[\sum_{i=1}^{n} \sum_{S\subset N, i\in S} \frac{(n-|S|)!|S-1|!}{n!} \times c(S) - \sum_{i=1}^{n} \sum_{S\subset N, i\notin S} \frac{(n-|S|-1)!|S|!}{n!} \times c(S)\right] =$$

$$\frac{1}{n!c(N)} \times \left[\sum_{S\subset N} \sum_{i\in S} \left[\frac{(n-|S|)!|S-1|!}{n!} \times c(S)\right] - \sum_{S\subset N} \sum_{i\notin S} \left[\frac{(n-|S|-1)!|S|!}{n!} \times c(S)\right]\right] =$$

$$\frac{1}{n!c(N)} \times \left[n!c(N) + \sum_{S\subset N, S\neq N} \sum_{i\in S} \left[\frac{(n-|S|)!|S-1|!}{n!} \times c(S)\right] - \sum_{S\subset N} \sum_{i\notin S} \left[\frac{(n-|S|-1)!|S|!}{n!} \times c(S)\right]\right] =$$

$$\frac{1}{n!c(N)} \times \left[n!c(N) + \sum_{S\subset N, S\neq N} \left[\sum_{i\in s} \frac{(n-|S|)!|S-1|!}{n!} - \sum_{i\notin s} \frac{(n-|S|-1)!|S|!}{n!}\right] \times c(S)\right] =$$

$$\frac{1}{n!c(N)} \times \left[n!c(N) + \sum_{S\subset N, S\neq N} \left[|S| \times \frac{(n-|S|)!|S-1|!}{n!} - (n-|S|) \times \frac{(n-|S|-1)!|S|!}{n!}\right] \times c(S)\right] =$$

$$\frac{1}{n!c(N)} \times \left[n!c(N) + \sum_{S\subset N, S\neq N} \left[\frac{(n-|S|)!|S|!}{n!} - \frac{(n-|S|)!|S|!}{n!}\right] \times c(S)\right] =$$

$$\frac{1}{n!c(N)} \times \left[n!c(N) + \sum_{S\subset N, S\neq N} 0 \times c(S)\right] = 1$$

Thus, the above all-or-nothing deal is well defined.

- $Utility_i(p_1,\dots,p_n) = c(T_i) - p_i \times c(N) = c(T_i) - \frac{\sum_{S\subset N, i\notin S} \frac{(n-|S|-1)!|S|!}{n!}\left[c(S\cup\{i\})-c(S)\right]}{c(N)} \times c(N) =$

$$c(T_i) - \sum_{S\subset N, i\notin S} \frac{(n-|S|-1)!|S|!}{n!}\left[c(S\cup\{i\})-c(S)\right] = u_i$$

This means that each agent gets an expected utility that is exactly its Shapley Value.

4.2 Is the Shapley Value-based Mechanism a PMM?

In this section we consider the question of whether the Shapley Value-based mechanism is a member of the class of Product Maximizing Mechanisms.

4.2.1 Two-agent Domains

In two-agent subadditive TODs, the answer to the question is affirmative.

<u>Theorem 19</u>: For any encounter in a two-agent subadditive TOD, the Shapley Value based Mechanism over all-or-nothing deals is a PMM.

<u>Proof</u>:

Let (T_1,T_2) be the encounter. From theorems 9 and 11 follows that using any PMM over all-or-nothing deals will lead to an agreement on an all-or-nothing deal δ that satisfies:
$Utility_1(\delta) = Utility_2(\delta) = ½ * [C(T_1)+C(T_2)-C(T_1,T_2)]$.

Therefore, $\prod(\delta) = Utility_1(\delta)*Utility_2(\delta) = ¼ * [C(T_1)+ C(T_2)- C(T_1,T_2)]^2$.

From the definition of a PMM we know that δ will maximize the product of the agent utilities. Thus, the maximal value of the product of the agent utilities in this encounter is
$¼ * [C(T_1)+C(T_2)-C(T_1,T_2)]^2$.

Let δ' be the all-or-nothing deal that the agents would reach when using the Shapley Value-based Mechanism. According to the definition of the mechanism $\delta'=(p_1, p_2)$ where
$p_1 = \frac{C(T_1)-C(T_2)+C(T_1\cup T_2)}{2C(T_1\cup T_2)}$ and $p_2 = \frac{C(T_2)-C(T_1)+C(T_1\cup T_2)}{2C(T_1\cup T_2)}$.

Thus,

$$Cost_1(\delta')= p_1 \times C(T_1\cup T_2)= \frac{C(T_1)-C(T_2)+C(T_1\cup T_2)}{2}$$

$$Cost_2(\delta')= p_2 \times C(T_1\cup T_2)= \frac{C(T_2)-C(T_1)+C(T_1\cup T_2)}{2}$$

Therefore,

$$Utility_1(\delta')=C(T_1)-Cost_1(\delta')=\frac{C(T_1)+C(T_2)-C(T_1\cup T_2)}{2}$$

$$Utility_2(\delta')=C(T_2)-Cost_2(\delta')=\frac{C(T_1)+C(T_2)-C(T_1\cup T_2)}{2}$$

This means that $\prod(\delta')= ¼ * [C(T_1)+C(T_2)-C(T_1,T_2)]^2$. Thus, δ' maximize the product of the agent utilities and therefore δ' is a PMM.

4.2.2 More-than-two-agent Domains

The situation is different in more-than-two-agent subadditive TODs. Here, in most of the cases the answer to the question is negative, i.e., the Shapley Value-based Mechanism is not a PMM.

For an example of an encounter in a multi-agent subadditive TOD in which the agreement that the agents would reach using the Shapley Value-based Mechanism is not a PMM, consider the following example in the Postmen Domain.

Example: Let the graph be as in the figure 13. The length (and thus the cost) of each edge in the figure is 1. The post office is at node A.

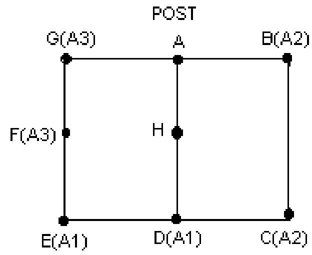

Figure 13

Agent 1 must deliver letters to nodes D and E, agent 2 must deliver letters to nodes B and C, and agent 3 must deliver letters to nodes F and G.

The cost for agent 1of delivering his letters alone is 6 and the cost for agent 2 and 3 is 4 (each). One agent can deliver the letters of agents 1 and 2 in the cost of 7.The same cost is required when one agent deliver the letters of agents 2 and 3. The letters of agents 1 and 3 can be delivered at cost of 6 and all the letters (of all three agents) can be delivered at a total cost of 7.

Let δ be the deal that the agents agree on when they are using the Shapley Value-based Mechanism over all-or-nothing deals. Thus, δ=(p_1, p_2, p_3) such that:

$$p_i = \frac{\sum_{S\subset\{1,2,3\}, i\notin S} \frac{(3-|s|-1)!|s|!}{3!}[C(S\cup\{i\})-C(S)]}{C(T_1\cup T_2\cup T_3)}$$

According to the formula,

$p_1 = \frac{17}{42}$, $p_2 = \frac{14}{42}$ and $p_3 = \frac{11}{42}$.

Therefore,

$$Utility_1 = c(T_1) - p_1 \times c(T_1\cup T_2\cup T_3) = 6 - \tfrac{17}{42}\times 7 = 3\tfrac{1}{6}$$
$$Utility_2 = c(T_2) - p_2 \times c(T_1\cup T_2\cup T_3) = 4 - \tfrac{14}{42}\times 7 = 1\tfrac{2}{3}$$
$$Utility_3 = c(T_3) - p_3 \times c(T_1\cup T_2\cup T_3) = 4 - \tfrac{11}{42}\times 7 = 2\tfrac{1}{6}$$

Thus, the utility is not divided equally between the agents and therefore according to theorem 11 the mechanism we used is not a PMM.

This is exactly what happens in the general case. Let (T_1, T_2,…,T_n) be an encounter in a multi-agent subadditive TOD, and let δ be the deal that the agents will reach using the Shapley Value-based Mechanism over all-or-nothing deals. Thus,δ = (p_1,…, p_n),such that

$$p_i = \frac{\sum_{S\subset N, i\notin S} \frac{(n-|s|-1)!|s|!}{n!}[C(S\cup\{i\})-C(S)]}{C(N)}.$$

Therefore,

$$Utility_i(\delta) = C(T_i) - p_i \times c(N) = C(T_i) - \sum_{S \subset N, i \notin S} \frac{(n-|s|-1)!|s|!}{n!}[C(S \cup \{i\}) - C(S)] =$$
$$\frac{[n! - (n-1)!]}{n!} \times C(T_i) + \sum_{S \neq \phi, S \subset N, i \notin S} \frac{(n-|s|-1)!|s|!}{n!}[C(S) - C(S \cup \{i\})]$$

Thus, for n>2 there is no reason that all the $Utility_i(\delta)$'s will be equal. However, a PMM divides the utility equally between the agents. Therefore δ is not necessarily a PMM.

4.3 Evaluation of the Mechanism

Since the Shapley Value-based Mechanism is not a PMM, it may have different attributes than the PMMs. In this section, we consider its attributes with regard to its efficiency, stability, simplicity, distribution, and symmetry.

4.3.1 Efficiency of the Mechanism

Theorem 20: The Shapley Value-based Mechanism is efficient.

Proof:

Let $(T_1, T_2,\ldots,T_n)$ be an encounter in a multi-agent subadditive TOD, and let $\delta = (p_1,\ldots,p_n)$ be the deal that the agents will reach using the Shapley Value-based Mechanism over all-or-nothing deals. In order to show that δ is efficient, it is sufficient to show that δ is pareto optimal. Using proof by contradiction, let's assume that there exists another deal $\delta^* = (p_1^*, p_2^*,\ldots, p_n^*)$ such that $\delta^* \succ \delta$. This means that ∀i 1≤i≤n $Utility_i(\delta^*) \geq Utility_i(\delta)$ and that there exists j 1≤j≤n such that $Utility_j(\delta^*) > Utility_j(\delta)$. This also means that ∀i 1≤i≤n $c(T_i) - p_i^* \times c(N) \geq c(T_i) - p_i \times c(N)$ and ∃j 1≤j≤n such that $c(T_j) - p_j^* \times c(N) > c(T_j) - p_j \times c(N)$. Thus, ∀i 1≤i≤n $p_i \geq p_i^*$ and ∃j 1≤j≤n such that $p_j > p_j^*$. Therefore $\sum_{i=1}^{n} p_i^* < \sum_{i=1}^{n} p_i = 1$. This contradicts the fact that δ^* is a deal. Thus, δ is pareto optimal.

4.3.2 Stability of the Mechanism

- **Individual Rationality**

The Mechanism satisfies this criterion as well.

Theorem 21: The Shapley Value-based Mechanism satisfies the individual rationality condition.

Proof:

Let $(T_1, T_2,\ldots,T_n)$ be an encounter in a multi-agent subadditive TOD, and let δ be the deal that the agents will reach using the Shapley Value-based Mechanism over all-or-nothing deals. Thus, $\delta = (p_1,\ldots, p_n)$, such that

$$p_i = \frac{\sum_{S \subset N, i \notin S} \frac{(n-|s|-1)!|s|!}{n!}[C(S \cup \{i\}) - C(S)]}{C(N)}.$$

Therefore,

$$Utility_i(\delta) = C(T_i) - p_i \times c(N) = C(T_i) - \sum_{S \subset N, i \notin S} \frac{(n-|s|-1)!|s|!}{n!}[C(S \cup \{i\}) - C(S)]$$

Because of the subadditivity of the domain $C(S \cup \{i\}) \le C(S) + C(T_i)$. This also means that, $-[C(S \cup \{i\}) - C(S)] \ge -C(T_i)$.

Therefore,

$$Utility_i(\delta) \ge C(T_i) - \sum_{S \subset N, i \notin S} \frac{(n-|s|-1)!|s|!}{n!} \times C(T_i) = C(T_i) \times \left[1 - \sum_{S \subset N, i \notin S} \frac{(n-|s|-1)!|s|!}{n!}\right] = 0$$

Thus, δ is individual rational.

- **Group Rationality**

This criterion is equivalent to efficiency, and thus according to 4.3.1, the Shapley Value-based Mechanism satisfies this condition.

- **Coalition Rationality**

Since we assume that the full coalition will be formed, the coalition rationality means that no group of agents should have any incentive to deviate from the full coalition and creates a sub-coalition. This condition is very strong and cannot always be satisfied.

Consider the encounter from a three-agent Postmen Domain that can be seen in Figure 14.

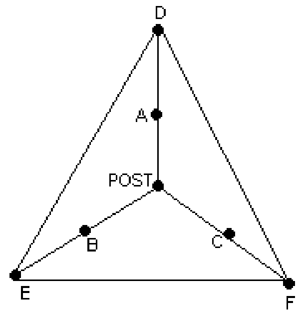

Figure 14

The Post Office is in the center. The length of each arch is 1. The encounter is (T_1 = {A,D}, T_2 = {B,E}, T_3 = {C,F}). Each agent can deliver its letters with a cost of 4. The cost of delivering the letters of any two agents is 5. Therefore, the joint utility of any two agents' coalition is (according to theorem 9) is (2*4)-5=3. The cost of delivering all the letters is 8. Therefore, the joint utility of the full coalition (according to the same theorem) is (3*4)-8=4.

We would like to find a deal δ such that:

$\forall i \neq j \in \{1,2,3\}$ $Utility_i(\delta) + Utility_j(\delta) \ge 3$ (i.e., no pair of agents would have any incentive to deviate from the full coalition and creates a sub-coalition).

If we add up all the inequalities, we will have $Utility_1(\delta) + Utility_2(\delta) + Utility_3(\delta) \ge 4½$.

However, we know from theorem 9 that $Utility_1(\delta) + Utility_2(\delta) + Utility_3(\delta) = 4$. Therefore, such a δ doesn't exist. This means that in any division of the utility of the full coalition among the agents there will be at least two agents that will prefer to opt out of the coalition and form a sub-coalition!

However, in *concave* TODs, the Shapley Value-based Mechanism satisfies coalition rationality. The theorem below (22) is based on Shapley [21].

Theorem 22: In concave TODs, the Shapley Value-based Mechanism always satisfies the criterion of coalition rationality.

Proof:

Let $(T_1, T_2,\ldots,T_n)$ be an encounter in a multi-agent concave TOD, and let δ be the deal that the agents will reach using the Shapley Value-based Mechanism over all-or-nothing deals. Thus, $\delta = (p_1,\ldots, p_n)$, such that

$$p_i = \frac{\sum_{S\subset N, i\notin S} \frac{(n-|s|-1)!|s|!}{n!}[C(S\cup\{i\})-C(S)]}{C(N)}.$$

In order to prove that δ is coalition rational, it is sufficient to show that for every group of agents $G \subseteq N$, $\sum_{i\in G} Utility_i(\delta) \geq \sum_{i\in G} c(T_i) - c(G)$.

From the definition of δ, we know that

$$Utility_i(\delta) = C(T_i) - p_i \times c(N) = C(T_i) - \sum_{S\subset N, i\notin S} \frac{(n-|s|-1)!|s|!}{n!}[C(S\cup\{i\})-C(S)].$$

Thus, it is sufficient to show that

$$\sum_{i\in G}\left\{C(T_i) - \sum_{S\subset N, i\notin S} \frac{(n-|s|-1)!|s|!}{n!}[C(S\cup\{i\})-C(S)]\right\} \geq \sum_{i\in G} C(T_i) - C(G)$$

$$C(G) \geq \sum_{i\in G}\sum_{S\subset N, i\notin S} \frac{(n-|s|-1)!|s|!}{n!}[C(S\cup\{i\})-C(S)]$$

We denote the last inequality by (*).

Let ω be any permutation of the agents. Let ω(i) be the place of agent A_i in permutation ω, and let $S_{\omega,i} = \{j\in N \mid \omega(j)\leq\omega(i)\}$, i.e., all the predecessors of i according to ω.

Let $G\subseteq N$ (a group of agents) and let $k = \arg\min\{\omega(j) \mid j\in N-G\}$. The definition of k means that all the predecessors of k, according to ω, are in G, but k itself isn't.

We have:

1. $S_{\omega,k-1} \subseteq S_{\omega,k}$
2. $G \cup S_{\omega,k} = G \cup \{k\}$
3. $G \cup S_{\omega,k-1} = G$

Hence, by the concavity of the domain:

$c(G \cup \{k\}) - S_{\omega,k} \leq c(G) - S_{\omega,k-1}$.

This also means, $c(G \cup \{k\}) - c(G) \leq S_{\omega,k} - S_{\omega,k-1}$.

Repeating this argument n-|G|-1 times (each time adding the "smallest" element according to ω to the "augmented" G), and summing all the inequalities, we obtain

$$c(N)-c(G)\leq \sum_{k\notin G}\left[c\left(S_{\omega,k}\right)-c\left(S_{\omega,k-1}\right)\right].$$

Since $c(N)=\sum_{1\leq k\leq n}\left[c\left(S_{\omega,k}\right)-c\left(S_{\omega,k-1}\right)\right]$, we have $c(G)\geq \sum_{k\in G}\left[c\left(S_{\omega,k}\right)-c\left(S_{\omega,k-1}\right)\right]$.

Since ω is arbitrary, this is true for every permutation of the agents. Therefore,

$$c(G)\geq \frac{1}{n!}\times \sum_{\omega\in s(N)}\sum_{k\in G}\left[c\left(S_{\omega,k}\right)-c\left(S_{\omega,k-1}\right)\right].$$

Thus,

$$c(G)\geq \frac{1}{n!}\times \sum_{k\in G}\sum_{\omega\in s(N)}\left[c\left(S_{\omega,k}\right)-c\left(S_{\omega,k-1}\right)\right]=$$

$$\frac{1}{n!}\times \sum_{k\in G}\sum\nolimits_{S\subset N,k\notin S}\left(n-|S|-1\right)!|S|!\times\left[c\left(S\cup\{i\}\right)-c\left(S\right)\right]$$

But, this is exactly inequality (*). Therefore, inequality (*) holds and thus δ is coalition rational.

4.3.3 Distribution of the Mechanism

There is no central decision-maker in this mechanism. For every encounter, there is only *one* deal that the agents can agree on when using the Shapley Value-based Mechanism and having all the information each agent can calculate it.

4.3.4 Symmetry of the Mechanism

Theorem 23: The Shapley Value-based Mechanism satisfies the symmetry condition.

Proof:

Let $(T_1, T_2,\ldots,T_n)$ be an encounter in a multi-agent subadditive TOD, and let δ be the deal that the agents will reach using the Shapley Value-based Mechanism over all-or-nothing deals. Thus, $\delta = (p_1,\ldots, p_n)$, such that

$$p_i=\frac{\sum_{S\subset N,i\notin S}\frac{(n-|s|-1)!|s|!}{n!}[C(S\cup\{i\})-C(S)]}{C(N)}.$$

Let A_i and A_j be two symmetric agents. This means that $\forall S\subseteq N$ such $i\notin S$ and $j\notin S$ holds $c(S\cup\{i\}) = c(S\cup\{j\})$. Therefore,

1) $\forall S\subseteq N$ such $i\notin S$ and $j\notin S$ holds $c(S\cup\{i\}) - c(S) = c(S\cup\{j\}) - c(S)$.

2) $\forall S\subseteq N$ such $i\notin S$ and $j\notin S$ holds $c(S\cup\{i\}\cup\{j\}) - c(S\cup\{j\}) = c(S\cup\{j\}\cup\{i\}) - c(S\cup\{i\})$.

Thus,

$$\frac{\sum_{S\subset N,i\notin S}\frac{(n-|s|-1)!|s|!}{n!}[C(S\cup\{i\})-C(S)]}{C(N)}=\frac{\sum_{S\subset N,j\notin S}\frac{(n-|s|-1)!|s|!}{n!}[C(S\cup\{j\})-C(S)]}{C(N)}.$$

This means that $p_i = p_j$. This also means that:

$Utility_i(\delta) = c(T_i) - p_i * c(N) = c(T_j) - p_j * c(N) = Utility_j(\delta)$.

Thus, A_i and A_j are assigned the same utility by the mechanism. Hence, the mechanism is symmetric.

4.3.5 Simplicity of the Mechanism

- **Computational Complexity**

The naive method for calculating the Shapley Value has an exponential complexity in the number of the agents. In order to overcome this problem Zlotkin and Rosenschein [32] presented a simple mechanism, which they called *Random Permutation Mechanism.* In this mechanism, the agents choose a random permutation and form the full coalition, according to the chosen permutation. Each agent gets utility that is equal to its contribution to the coalition, at the time he joined it. They showed that if each permutation has an equal chance of being chosen, the expected utility that each agent gets from this mechanism is equal to its Shapley Value. In order to choose this random permutation in a non-manipulable way, they suggested a cryptographic mechanism. Each agent chooses a random permutation and a key, encrypts the permutation using the key, and broadcasts the encrypted message to all other agents. After he has received all the encrypted messages, the agent broadcasts the key. Each agent unecrypts all messages using the associated keys. The consensus permutation is the combination of all permutations.

- **Communication Complexity**

In the basic Shapley Value-based Mechanism there is no communication at all. In the *Random Permutation Mechanism,* each agent sends two messages (the encrypted permutation and the key) to all other agents. Thus, there are $O(n^2)$ messages.

4.4 Incentive Compatible Mechanisms

The Shapley Value-based Mechanism can be straightforwardly implemented only if all the agents have full information about each other's goals. In many situations this is not the case.

There are several frameworks for dealing with incomplete information, but the framework we explore here is that of a "-1 negotiation phase" in which agents simultaneously declare private information before the beginning of the negotiation (we used this framework above as well for the case of PMMs). The negotiation then proceeds as if the information were true.

In this section, we provide a characterization of the relationship between kind of lies and domain attributes. There are three kinds of lies in TODs (hidden tasks, phantom tasks, decoy tasks) and we considered three domain attributes (subadditivity, concavity, modularity). The resulting table is represented in Figure 15. Its notation is described below.

	Hidden	Phantom	Decoy
Subadditive	$T_1\downarrow$	T/P_2	L_4
Concave	$T\downarrow$	$T\downarrow$	$\Leftarrow T_3\downarrow$
Modular	T	T	$\Leftarrow T$

Figure 15

The entries in the table marked T signify that honesty is the best policy (lies are never beneficial). The entry L means that there exists an encounter such that at least one of the agents has an incentive to lie. The entry marked T/P refers to lies that are not beneficial because they may be discovered; if the agent tells the truth, it is because he is afraid of the penalty that he will be levied if his lie is discovered. Thus, T/P can be transformed to T if the mechanism is enhanced to include sufficiently high penalty for discovered lies.

The arrows ↓ signify implication between sells in the table. For example, the ↓ connecting the [Subadditive, Hidden] entry with the [Concave, Hidden] entry means that if decoy lie is never beneficial in subadditive domains, it is certainly never beneficial in concave domains (a sub-class). Similarly, the arrow ⇦ connecting the [Concave, Decoy] entry with the [Concave, Phantom] entry means that if there doesn't exist a decoy beneficial lie in a given encounter, there certainly won't exist a phantom beneficial lie in that encounter.

To fill out the table, therefore, we need only demonstrate a small number of "fixed points", which in turn imply all the other table entries. The fixed points that need to be demonstrated are numbered in the table from 1 to 4.

4.4.1 Incentive Compatible Fixed Points

The three incentive compatible fixed points are determined by the theorems below.

<u>Theorem 24</u>: [Fixed point 1]: For any encounter in a multi-agent subadditive TOD, using the Shapley Value-based mechanism over all-or-nothing deals, every "hiding" lie is not beneficial.

<u>Proof</u>:

According to the Shapely Value-based Mechanism for subadditive TODs, the agents will agree on an all-or-nothing deal, $\delta = (p_1,\dots, p_n)$, such that:

$$p_i = \frac{\sum_{S\subset N, i\notin S} \frac{(n-|s|-1)!|s|!}{n!}[C(S\cup\{i\}) - C(S)]}{C(N)}$$

Therefore:

$$Cost_i(\delta) = C(N)\times p_i = \sum_{S\subset N, i\notin S} \frac{(n-|s|-1)!|s|!}{n!}[C(S\cup\{i\}) - C(S)]\,.$$

Assume that Agent A_1 lies and hides some tasks $L \subseteq T_1$ and broadcasts $T_1^* = T_1 - L$.

If agents negotiate over $T_1^*, T_2,\dots, T_n$ they will agree on $(p_1^*,\dots, p_n^*)$, such that:

$$p_i^* = \frac{\sum_{S\subset N, i\notin S} \frac{(n-|s|-1)!|s|!}{n!}[C((S\cup\{i\})^*) - C(S^*)]}{C(N^*)}$$

When $\forall U\subseteq N$, $U^* = U - L$ if $1\in U$ and $U^* = U$ otherwise.

Agent A_1 will perform N with probability p_1^* or perform only L with probability $1-p_1^*$.

Therefore,

$$Cost_1^*(\delta)=C(N)\times p_1^*+C(L)\times(1-p_1^*)=$$

$$C(N)\times\frac{\sum_{S\subset N,1\notin S}\frac{(n-|s|-1)!|s|!}{n!}[C((S\cup\{1\})^*)-C(S^*)]}{C(N^*)}+C(L)\times(1-\frac{\sum_{S\subset N,1\notin S}\frac{(n-|s|-1)!|s|!}{n!}[C((S\cup\{1\})^*)-C(S^*)]}{C(N^*)})=$$

$$C(N)\times\frac{\sum_{S\subset N,1\notin S}\frac{(n-|s|-1)!|s|!}{n!}[C((S\cup\{1\})^*)-C(S)]}{C(N^*)}+C(L)\times(1-\frac{\sum_{S\subset N,1\notin S}\frac{(n-|s|-1)!|s|!}{n!}[C((S\cup\{1\})^*)-C(S)]}{C(N^*)})$$

In order to prove that it is not beneficial to A_1 to hide tasks, it is sufficient to show that $Cost_1 \leq Cost_1^*$.

$$\sum_{S\subset N,1\notin S}\frac{(n-|s|-1)!|s|!}{n!}[C(S\cup\{1\})-C(S)]\leq$$

$$C(N)\times\frac{\sum_{S\subset N,1\notin S}\frac{(n-|s|-1)!|s|!}{n!}[C((S\cup\{1\})^*)-C(S)]}{C(N^*)}+C(L)\times(1-\frac{\sum_{S\subset N,1\notin S}\frac{(n-|s|-1)!|s|!}{n!}[C((S\cup\{1\})^*)-C(S)]}{C(N^*)})$$

$$\sum_{S\subset N,1\notin S}(n-|s|-1)!|s|![C(S\cup\{1\})-C(S)]\leq$$

$$C(N)\times\frac{\sum_{S\subset N,1\notin S}(n-|s|-1)!|s|![C((S\cup\{1\})^*)-C(S)]}{C(N^*)}+C(L)\times(n!-\frac{\sum_{S\subset N,1\notin S}(n-|s|-1)!|s|![C((S\cup\{1\})^*)-C(S)]}{C(N^*)})$$

$$C(N^*)\times\sum_{S\subset N,1\notin S}(n-|s|-1)!|s|![C(S\cup\{1\})-C(S)]\leq$$

$$C(N)\times\sum_{S\subset N,1\notin S}(n-|s|-1)!|s|![C((S\cup\{1\})^*)-C(S)]+C(L)\times(n!\times C(N^*)-\sum_{S\subset N,1\notin S}(n-|s|-1)!|s|![C((S\cup\{1\})^*)-C(S)])$$

There are two cases:

1) $n!C(N^*)-\sum_{S\subset N,1\notin S}(n-|s|-1)!|s|!C((S\cup\{1\})^*)\leq n!C(N)-\sum_{S\subset N,1\notin S}(n-|s|-1)!|s|!C(S\cup\{1\}))$

2) $n!C(N^*)-\sum_{S\subset N,1\notin S}(n-|s|-1)!|s|!C((S\cup\{1\})^*)< n!C(N)-\sum_{S\subset N,1\notin S}(n-|s|-1)!|s|!C(S\cup\{1\}))$

Case 1:

Because of the subadditivity of the domain $C(N)\leq C(N^*)+C(L)$. This means that $C(N)-C(N^*)\leq C(L)$. It will be sufficient to show that:

$$C(N^*)\times\sum_{S\subset N,1\notin S}(n-|s|-1)!|s|![C(S\cup\{1\})-C(S)]\leq$$

$$C(N)\times\sum_{S\subset N,1\notin S}(n-|s|-1)!|s|![C((S\cup\{1\})^*)-C(S)]+$$

$$[C(N)-C(N^*)]\times\{n!\times C(N^*)-\sum_{S\subset N,1\notin S}(n-|s|-1)!|s|![C((S\cup\{1\})^*)-C(S)]\}$$

$$C(N^*)\times\sum_{S\subset N,1\notin S}(n-|s|-1)!|s|![C(S\cup\{1\})-C(S)]\leq$$

$$n!\times C(N)\times C(N^*)-C(N^*)\times\{n!\times C(N^*)-\sum_{S\subset N,1\notin S}(n-|s|-1)!|s|![C((S\cup\{1\})^*)-C(S)]\}$$

$$\sum_{S\subset N,1\notin S}(n-|s|-1)!|s|![C(S\cup\{1\})-C(S)]\le$$
$$n!\times C(N)-n!\times C(N^*)+\sum_{S\subset N,1\notin S}(n-|s|-1)!|s|![C((S\cup\{1\})^*)-C(S)]$$

$$\sum_{S\subset N,1\notin S}(n-|s|-1)!|s|!C(S\cup\{1\})\le n!\times C(N)-n!\times C(N^*)+\sum_{S\subset N,1\notin S}(n-|s|-1)!|s|!C((S\cup\{1\})^*)$$

$$n!\times C(N^*)-\sum_{S\subset N,1\notin S}(n-|s|-1)!|s|!C((S\cup\{1\})^*)\le n!\times C(N)-\sum_{S\subset N,1\notin S}(n-|s|-1)!|s|!C(S\cup\{1\})$$

Case 2:

It will be sufficient to show that:

$$C(N^*)\times\sum_{S\subset N,1\notin S}(n-|s|-1)!|s|![C(S\cup\{1\})-C(S)]\le$$
$$C(N)\times\sum_{S\subset N,1\notin S}(n-|s|-1)!|s|![C((S\cup\{1\})^*)-C(S)]+C(L)\times(n!\times C(N)-\sum_{S\subset N,1\notin S}(n-|s|-1)!|s|![C(S\cup\{1\})-C(S)])$$

$$C(N^*)\times\sum_{S\subset N,1\notin S}(n-|s|-1)!|s|![C(S\cup\{1\})-C(S)]\le$$
$$C(N)\times\sum_{S\subset N,1\notin S}(n-|s|-1)!|s|![C((S\cup\{1\})^*)-C(S)]+$$
$$C(L)\times n!\times C(N)-C(L)\times\sum_{S\subset N,1\notin S}(n-|s|-1)!|s|![C(S\cup\{1\})-C(S)]$$

$$0\le C(N)\times\left\langle C(L)\times n!+\sum_{S\subset N,1\notin S}(n-|s|-1)!|s|![C((S\cup\{1\})^*)-C(S)]\right\rangle$$
$$-\left\langle C(N^*)+C(L)\right\rangle\times\sum_{S\subset N,1\notin S}(n-|s|-1)!|s|![C(S\cup\{1\})-C(S)]$$

Because of the subadditivity of the domain $C(N)\le C(N^*)+C(L)$. It will be sufficient to show that:

$$0\le C(N)\times\left\langle C(L)\times n!+\sum_{S\subset N,1\notin S}(n-|s|-1)!|s|![C((S\cup\{1\})^*)-C(S)]\right\rangle-C(N)\times\sum_{S\subset N,1\notin S}(n-|s|-1)!|s|![C(S\cup\{1\})-C(S)]$$

$$0\le C(L)\times n!+\sum_{S\subset N,1\notin S}(n-|s|-1)!|s|![C((S\cup\{1\})^*)-C(S)]-\sum_{S\subset N,1\notin S}(n-|s|-1)!|s|![C(S\cup\{1\})-C(S)]$$

$$0\le C(L)\times n!+\sum_{S\subset N,1\notin S}(n-|s|-1)!|s|![C((S\cup\{1\})^*)-C(S\cup\{1\})]$$

We know (from subadditivity) that: $-C(L)\le C((S\cup\{i\})^*)-C(S\cup\{i\})$. It will be sufficient to show that:

$$0\le C(L)\times n!+\sum_{S\subset N,1\notin S}(n-|s|-1)!|s|!(-C(L))$$

$$0\le n!-\sum_{S\subset N,1\notin S}(n-|s|-1)!|s|!$$

Which is true.

<u>Theorem 25</u>:[Fixed point 2] For any encounter in a multi-agent subadditive TOD, using the Shapley Value-based mechanism over all-or-nothing deals, every "phantom" lie has a positive probability of being discovered. Therefore, with sufficiently severe penalty mechanism, telling the truth is the optimal strategy.

<u>Proof</u>:

Phantom lies are not safe when using all-or-nothing deals, because there is a positive probability that one of the non-lying agents will be assigned to execute all tasks (including the phantom tasks). In this case the phantom lie would be revealed.

<u>Theorem 26</u>:[Fixed point 3]: For any encounter in a multi-agent concave TOD, using the Shapley Value-based mechanism over all-or-nothing deals, every "decoy" lie is not beneficial.

<u>Proof</u>:

Let (T_1, T_2,…,T_n) be the true encounter. According to the Shapely Value-based mechanism for subadditive TODs, the agents will agree on the following all-or-nothing deal, (p_1,…, p_n), such that:

$$p_i = \frac{\sum_{S\subset N, i\notin S} \frac{(n-|s|-1)!|s|!}{n!}[C(S\cup\{i\})-C(S)]}{C(N)}$$

Therefore:

$$Cost_i = C(N)\times p_i = \sum_{S\subset N, i\notin S} \frac{(n-|s|-1)!|s|!}{n!}[C(S\cup\{i\})-C(S)]$$

Assume that agent A_1 lies and declares that his initial set of tasks is $T_1^* = T_1 \cup L$, where L is a set of decoy tasks. The negotiation that assumes this information is true will lead to an agreement on an all-or-nothing deal (p_1^*,…, p_n^*), such that:

$$p_i^* = \frac{\sum_{S\subset N, i\notin S} \frac{(n-|s|-1)!|s|!}{n!}[C((S\cup\{i\})^*)-C(S^*)]}{C(N^*)}$$

When $\forall U \subseteq N$, $U^* = U - L$ if $1 \in U$ and $U^* = U$ otherwise.

For every A_i, $i \neq 1$:

$$Cost_i^* = C(N^*)\times p_i^* = \sum_{S\subset N, i\notin s} \frac{(n-|s|-1)!|s|!}{n!}[C((S\cup\{i\})^*)-C(S^*)]$$

We will show that $\forall i\neq 1$ $Cost_i^* \leq Cost_i$.

$$\sum_{S\subset N, i\notin s} \frac{(n-|s|-1)!|s|!}{n!}[C((S\cup\{i\})^*)-C(S^*)] \leq \sum_{S\subset N, i\notin s} \frac{(n-|s|-1)!|s|!}{n!}[C(S\cup\{i\})-C(S)]$$

$$\sum_{S\subset N, i\notin s} (n-|s|-1)!|s|![C((S\cup\{i\})^*)-C(S^*)] \leq \sum_{S\subset N, i\notin s} (n-|s|-1)!|s|![C(S\cup\{i\})-C(S)]$$

$$\sum_{S\subset N, i\notin s, 1\in S} (n-|s|-1)!|s|![C((S\cup\{i\})^*)-C(S^*)] + \sum_{S\subset N, i\notin s, 1\notin S} (n-|s|-1)!|s|![C((S\cup\{i\})^*)-C(S^*)] \leq$$
$$\sum_{S\subset N, i\notin s, 1\in S} (n-|s|-1)!|s|![C(S\cup\{i\})-C(S)] + \sum_{S\subset N, i\notin s, 1\notin S} (n-|s|-1)!|s|![C(S\cup\{i\})-C(S)]$$

$$\sum_{S\subset N, i\notin s, 1\in S} (n-|s|-1)!|s|![C(S\cup\{i\})-C(S)] + \sum_{S\subset N, i\notin s, 1\notin S} (n-|s|-1)!|s|![C((S\cup\{i\})^*)-C(S^*)] \leq$$
$$\sum_{S\subset N, i\notin s, 1\in S} (n-|s|-1)!|s|![C(S\cup\{i\})-C(S)] + \sum_{S\subset N, i\notin s, 1\notin S} (n-|s|-1)!|s|![C(S\cup\{i\})-C(S)]$$

$$\sum_{S\subset N, i\notin s, 1\in S} (n-|s|-1)!|s|![C((S\cup\{i\})^*)-C(S^*)] \leq \sum_{S\subset N, i\notin s, 1\in S} (n-|s|-1)!|s|![C(S\cup\{i\})-C(S)]$$

$$\sum_{T\subset N, i\notin T, 1\notin T} (n-|T|-2)!|T+1|![C((T\cup T_1^*\cup\{i\})-C(T\cup T_1^*)] \leq$$
$$\sum_{T\subset N, i\notin T, 1\notin T} (n-|T|-2)!|T+1|![C(T\cup T_1\cup\{i\})-C(T\cup T_1)]$$

From the definition of concavity we can conclude that:

$C(T_1 \cup T \cup L) - C(T_1 \cup T) \geq C(T_1 \cup T \cup \{i\} \cup L) - C(T_1 \cup T \cup \{i\})$.

Thus, $C(T_1^* \cup T) - C(T_1 \cup T) \geq C(T_1^* \cup T \cup \{i\}) - C(T_1 \cup T \cup \{i\})$.

This also means that $C(T_1^* \cup T \cup \{i\}) - C(T_1^* \cup T) \leq C(T_1 \cup T \cup \{i\}) - C(T_1 \cup T)$

We can thus conclude that

$$\sum_{T \subset N, i \notin T, 1 \notin T} (n - |T| - 2)!|T+1|![C((T \cup T_1^* \cup \{i\}) - C(T \cup T_1^*)] \leq$$
$$\sum_{T \subset N, i \notin T, 1 \notin T} (n - |T| - 2)!|T+1|![C(T \cup T_1 \cup \{i\}) - C(T \cup T_1)]$$

We have shown that $\forall i \neq 1$ $Cost_i^* \leq Cost_i$. In other words, even though agent A_i might have to perform all the decoy tasks, his cost can only be smaller than if agent A_1 would have told the truth.

We can sum the costs of all the agents:

$$\sum_{i=1}^{n} \cos t_i^* = p_1^* \times C(N) + \sum_{1=2}^{n} p_i^* \times C(N^*) = p_1^* \times C(N) + (1 - p_1^*) \times C(N^*) \geq C(N) = \sum_{i=1}^{n} p_i \times C(N) = \sum_{i=1}^{n} \cos t_i$$

We have shown that the sum of agents' costs when agent A_1 lies is greater or equal to the some of utilities when agent A_1 doesn't lie. However, we have already shown that for all $i \neq 1$, A_i's cost when agent A_1 lies is less than or equal to A_i's cost when agent A_1 doesn't lie. Thus, the conclusion is that A_1's actual cost cannot be smaller than his unvarnished cost. The lie is not beneficial for A_1.

Theorem 27: For any encounter in a multi-agent concave TOD, using the Shapley Value-based mechanism over all-or-nothing deals, every lie (including combinations of hidden, phantom and decoys) is not beneficial.

Proof:

Let (T_1, T_2,…, T_n) be the true encounter, and let the agreement that would be reached using the Shapley Value-based Mechanism with common knowledge of the true information be the all-or-nothing deal δ.

Assume that agent A_1 declares that his set of tasks is $T_1^* = \{T_1 - T_h \cup T_d\}$ where T_h is the set of all his hidden tasks, and T_d is the set of all his decoy tasks. Let the agreement that the agents will reach when they assume this information is true be δ^*.

Consider the negotiation where agent A_1 declares his set of tasks to be $T_1^{**} = \{T_1 \cup T_d\}$. The negotiation that assumes this information is true will lead to an agreement on an all-or-nothing deal δ^{**}.

From theorem 26 we now that declaring T_1^{**} instead of T_1 is not beneficial for A_1. This means that the *actual* cost of A_1's role in δ^{**} (which we denote by $\overline{Cost_1(\delta^{**})}$) is greater or equal to the cost of A_1's role in δ. I.e., $Cost_1(\delta) \leq \overline{Cost_1(\delta^{**})}$.

The apparent cost of A_1's role in δ^{**} is even higher, since apparently A_1 has to perform the decoy tasks T_d (and actually he doesn't). Thus, $\overline{Cost_1(\delta^{**})} \le Cost_1(\delta^{**})$.

Let us now consider the situation that A_1 declares $T_1^* = \{T_1^{**} - T_h\}$. From theorem 24 we can see that $Cost_1(\delta^{**}) \le \overline{Cost_1(\delta^*)}$, and the final conclusion is that $Cost_1(\delta) \le \overline{Cost_1(\delta^*)}$. Thus the lie is not beneficial to A_1.

Because of the theorem above, it is clear that for concave domains, agents cannot benefit by lying when using the Shapley Value-based Mechanism. This is also true in modular domains (a subcase). This can be seen in the table, where the entire row is marked true for modular and concave domains.

Additionally, in subadditive domain where decoy lies cannot be generated, the Shapley Value-based Mechanism with penalty for discovered lies is incentive compatible. This can be seen in the table, where the subadditive row is marked T and T/P (excluding the decoy column).

4.4.2 Non-Incentive Compatible Fixed Points

Fixed point 4: For an example for a beneficial decoy lie in a subadditive domain (e.g., the Postmen Domain with an unrestricted graph topology) using a the Shapley Value-based Mechanism, consider the following example:

Let the graph be as in Figure 16. The length (and thus the cost) of each edge in the figure is 1. The post office is at node A.

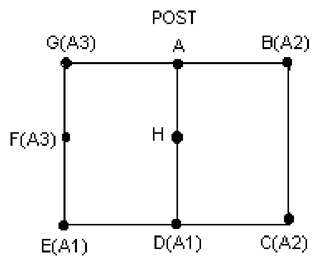

Figure 16

There are three agents. Agent 1 must deliver letters to nodes D and E, agent 2 must deliver letters to nodes B and C, and agent 3 must deliver letters to nodes F and G.

The cost for agent 1of delivering his letters alone is 6 and the cost for agent 2 and 3 is 4 (each). One agent can deliver the letters of agents 1 and 2 in the cost of 7.The same cost is required when one agent deliver the letters of agents 2 and 3. The letters of agents 1 and 3 can be delivered at cost of 6 and all the letters (of all three agents) can be delivered at a total cost of 7.

According to the Shapely Value-based mechanism for subadditive TODs, the agents will agree on the following all-or-nothing deal, (p_1, p_2, p_3), such that:

$$p_i = \frac{\sum_{S\subset\{1,2,3\}, i\notin S} \frac{(3-|s|-1)!|s|!}{3!}[C(S\cup\{i\})-C(S)]}{C(T_1,T_2,T_3)}$$

Thus,

$$p_1 = \frac{2C(T_1)+C(T_1,T_2)-C(T_2)+C(T_1,T_3)-C(T_3)+2C(T_1,T_2,T_3)-2C(T_2,T_3)}{6C(T_1,T_2,T_3)} = \frac{2\times6+7-4+6-4+2\times7-2\times7}{6\times7} = \frac{17}{42} = 0.4047$$

$$\cos t_1 = p_1 \times C(T_1,T_2,T_3) = \frac{17}{42}\times 7 = 2.833$$

After A_1 lies with producible decoy letter to node H, his apparent cost is still 6. Delivery of both his and agent 2's letters still cost 7, and the delivery of his and agent 3's letters still cost 6. However, delivery of the entire set of letters, now (apparently) costs 9.

They will agree on the following deal:

$$p_1^* = \frac{2C(T_1^*)+C(T_1^*,T_2)-C(T_2)+C(T_1^*,T_3)-C(T_3)+2C(T_1^*,T_2,T_3)-2C(T_2,T_3)}{6C(T_1^*,T_2,T_3)} = \frac{2\times6+7-4+6-4+2\times9-2\times7}{6\times9} = \frac{21}{54} = 0.3889$$

Agent A_1 will simply do the cycle (i.e., without the decoy letter). Thus,

$$\cos t_1^* = p_1^* \times C(T_1,T_2,T_3) = \frac{21}{54}\times 7 = 2.722$$

Thus, lying was beneficial to agent 1.

5 Default Decoy Lies in Subadditive Domains

In the previous chapters, we have seen that lying over all-or-nothing deals is almost always non-beneficial *no matter which mechanism we use*, PMM or the Shapley Value-based Mechanism. In both cases, agents cannot benefit by lying in concave and modular domains. Moreover, in subadditive domains where decoy lies cannot be generated, both mechanisms are incentive compatible if we give a penalty for discovered lies. Thus, in both cases, the only lie that *may* be beneficial is a decoy lie in a subadditive domain.

The main questions of this section are how much information does the liar need in order to make such a beneficial lie, and is it possible to make a default decoy lie that may benefit the liar but can never harm him.

5.1 Reliably Generating Decoy lies using PMM

Using a PMM, things are quite simple for the potential liar. It turns out that if an agent *knows* the relationship between his and the other agents' total costs, it is possible for him to generate a beneficial "default" lie that is made up of decoy tasks. In a subadditive domain, like the Postmen Domain, he can generate extra decoy tasks "along the way" (i.e., they don't increase his stand-alone cost), but which improve his position in the final agreement. This can be done by agent A_1 when the cost of T_1 (his own task set) is greater than the than the average of the costs of $T_2,\ldots,T_n$ (his opponents' tasks sets).

Theorem 28: For any encounter in a multi-agent subadditive TOD, and any PMM over all-or-nothing deals, if agent A_1 knows that the cost of T_1 is greater than the average of the costs of $T_2,\dots,T_n$, he can generate a default decoy lie that may benefit him, and will never harm him.

Proof:

Using theorems 9 and 11, we know that the agents will agree on a deal δ that satisfies

$$\forall i\; Utility_i(\delta) = \frac{1}{n} \times \left[\sum_{j=1}^{n} c(T_j) - c\left(\bigcup_{j=1}^{n} T_j \right) \right].$$

Negotiating over all-or-nothing deals means that they will agree on probabilities $(p_1,\dots, p_n)$ that satisfy the following relationship:

$$\forall i\; Utility_i(\delta) = c(T_i) - p_i \times c\left(\bigcup_{j=1}^{n} T_j \right)$$

Thus,

$$\forall i\; c(T_i) - p_i \times c\left(\bigcup_{j=1}^{n} T_j \right) = \frac{1}{n} \times \left[\sum_{j=1}^{n} c(T_j) - c\left(\bigcup_{j=1}^{n} T_j \right) \right].$$

Particularly,

$$c(T_1) - p_1 \times c\left(\bigcup_{j=1}^{n} T_j \right) = \frac{1}{n} \times \left[\sum_{j=1}^{n} c(T_j) - c\left(\bigcup_{j=1}^{n} T_j \right) \right].$$

Thus,

$$p_1 = \frac{(n-1) \times c(T_1) - \sum_{j=2}^{n} c(T_j) + c\left(\bigcup_{j=1}^{n} T_j \right)}{n \times c\left(\bigcup_{j=1}^{n} T_j \right)}.$$

If agent A_1 declares T_1^* which is a superset of T_1 (i.e., T_1 with extra decoy tasks), then they will agree on:

$$p_1^* = \frac{(n-1) \times c(T_1^*) - \sum_{j=2}^{n} c(T_j) + c\left(T_1^* \cup \bigcup_{j=2}^{n} T_j \right)}{n \times c\left(T_1^* \cup \bigcup_{j=2}^{n} T_j \right)}.$$

We have $c\left(T_1^* \cup \bigcup_{j=2}^{n} T_j \right) \ge c\left(\bigcup_{j=1}^{n} T_j \right)$ because of the monotonicity of c. If agent A_1 has chosen his decoy tasks such $c(T_1^*)=c(T_1)$, then we have that agent A_1's apparent utility is $c(T_1^*) - p_1^* \times c\left(T_1^* \cup \bigcup_{j=2}^{n} T_j \right)$, while his actual utility is $c(T_1) - p_1^* \times c\left(\bigcup_{j=1}^{n} T_j\right)$ (since he won't have to carry out his decoy tasks).

Thus, if $p_1^* < p_1$, agent A_1 benefits from this lie.

If $c(T_1) \geq \frac{\sum_{i=2}^{n} c(T_i)}{n-1}$, then $p_1^* \leq p_1$, from the definition of p_1^* and p_1, the equivalence of c(T_1^*) and c(T_1), and the inequality $c\left(T_1^* \cup \bigcup_{j=2}^{n} T_j\right) \geq c\left(\bigcup_{j=1}^{n} T_j\right)$.

The important result here is that the lie can be generated reliably without having full information about the other agents' sets of tasks.

5.2 Decoy Lies using the Shapley Value-based Mechanism

The situation is much more complicated when the agents use the Shapley Value-based Mechanism. In two-agent subadditive TODs, the situation is still simple. In such domains the Shapley Value-based Mechanism is a PMM, and therefore the result of theorem 28 is true when using the Shapley Value-based Mechanism as well. That is, if agent A_1 knows that the cost of T_1 is greater than the cost of T_2, he can generate a default decoy lie that may benefit him and will never hurt him.

Theorem 29: For any encounter in a two-agent subadditive TOD, using the Shapley Value-based mechanism over all-or-nothing deals, if agent A_1 knows that the cost of T_1 is greater than the cost of T_2, he can generate a default decoy lie that may benefit him, and will never harm him.

Proof:

The proof follows immediately from theorems 19 and 28.

In more than two-agent subadditive TODs, things are really complicated when using the Shapley Value-based mechanism. As in the previous cases (PMMs, two-agent domains) there are many examples in which a decoy lie is beneficial and there are many examples in which a decoy lie is harmful. However, it is not so simple to characterize the cases in which a decoy lie can only benefit the liar (and never harm him), nor the cases in which a decoy lie can only harm the liar (and never benefit him).

5.2.1 Examples

First, let have a look in a few examples.

Example 1: Consider again at the example presented above in chapter 6 at fixed point 4, shown again in figure 17.

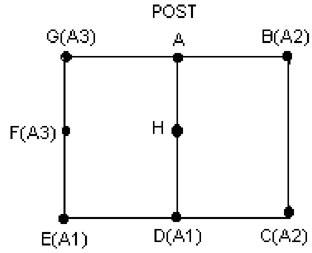

Figure 17

The length (and thus the cost) of each edge in the figure is 1. The post office is at node A. There are three agents. Agent 1 must deliver letters to nodes D and E, agent 2 must deliver letters to nodes B and C, and agent 3 must deliver letters to nodes F and G.

Thus, $c(T_1) = 6$, $c(T_2) = c(T_3) = 4$, $c(T_1 \cup T_2) = c(T_2 \cup T_3) = 7$, $c(T_1 \cup T_3) = 6$, $c(T_1 \cup T_2 \cup T_3) = 7$. According to the Shapely Value-based mechanism for subadditive TODs, the agents will agree on the following all-or-nothing deal, (p_1, p_2, p_3), such that:

$$p_i = \frac{\sum_{S \subset \{1,2,3\}, i \notin S} \frac{(3-|s|-1)!|s|!}{3!}[C(S \cup \{i\}) - C(S)]}{C(T_1, T_2, T_3)}$$

Thus,

$$p_1 = \frac{2C(T_1) + C(T_1,T_2) - C(T_2) + C(T_1,T_3) - C(T_3) + 2C(T_1,T_2,T_3) - 2C(T_2,T_3)}{6C(T_1,T_2,T_3)} = \frac{2\times 6 + 7 - 4 + 6 - 4 + 2\times 7 - 2\times 7}{6\times 7} = \frac{17}{42} = 0.4047$$

$$\cos t_1 = p_1 \times C(T_1,T_2,T_3) = \frac{17}{42}\times 7 = 2.833$$

After A_1 lies with producible decoy letter to node H, his apparent cost is still 6. Delivery of both his and agent 2's letters still cost 7, and the delivery of his and agent 3's letters still cost 6. However, delivery of the entire set of letters, now (apparently) costs 9.

They will agree on the following deal:

$$p_1^* = \frac{2C(T_1^*) + C(T_1^*,T_2) - C(T_2) + C(T_1^*,T_3) - C(T_3) + 2C(T_1^*,T_2,T_3) - 2C(T_2,T_3)}{6C(T_1^*,T_2,T_3)} = \frac{2\times 6 + 7 - 4 + 6 - 4 + 2\times 9 - 2\times 7}{6\times 9} = \frac{21}{54} = 0.3889$$

Agent A_1 will simply do the cycle (i.e., without the decoy letter). Thus,

$$\cos t_1^* = p_1^* \times C(T_1,T_2,T_3) = \frac{21}{54}\times 7 = 2.722$$

Thus, lying was beneficial to agent 1.

<u>Example 2</u>: Assume that after observing how successful his first lie was, agent 1 declares that he "needs" to deliver another (decoy) letter to node F. This lie changes only the cost of delivering both agents 1 and 2. The cost is now (apparently) 9.

The agents will agree on the following deal:

$$p_1^{**} = \frac{2C(T_1^{**}) + C(T_1^{**},T_2) - C(T_2) + C(T_1^{**},T_3) - C(T_3) + 2C(T_1^{**},T_2,T_3) - 2C(T_2,T_3)}{6C(T_1^{**},T_2,T_3)} = \frac{2\times 6 + 9 - 4 + 6 - 4 + 2\times 9 - 2\times 7}{6\times 9} = \frac{23}{54} = 0.425$$

Agent A_1 will simply do the cycle (i.e., without the decoy letter). Thus,

$$\cos t_1^{**} = p_1^{**} \times C(T_1,T_2,T_3) = \frac{23}{54}\times 7 = 2.98$$

Thus, lying is very bad for agent 1. It is much better for him to use only his first lie, or even to tell the truth…

<u>Example 3</u>: Consider another example from the Postman Domain given in Figure 18. The length (and thus the cost) of each edge is written next to it. The post office is at node A.

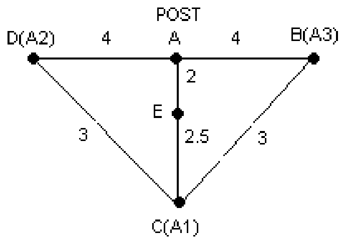

Figure 18

Agent 1 needs to deliver a letter to node C, with cost of 9. Agent 2 needs to deliver a letter to D with cost of 8 and agent 3 needs to deliver a letter to D with cost 8. If agent 1 cooperate with agent 2 or agent 3, he can deliver his and the other agent's letters at cost of 11½. Delivery of agent 2and agent 3's letters cost 14, and this is also the cost of delivery of all the letters.

According to the Shapely Value-based mechanism for subadditive TODs, the agents will agree on the following all-or-nothing deal, (p_1, p_2, p_3), such that:

$$p_1 = \frac{2C(T_1)+C(T_1,T_2)-C(T_2)+C(T_1,T_3)-C(T_3)+2C(T_1,T_2,T_3)-2C(T_2,T_3)}{6C(T_1,T_2,T_3)} = \frac{2\times9+11.5-8+11.5-8+2\times14-2\times14}{6\times14} = \frac{25}{84} = 0.297$$

$$\cos t_1 = p_1 \times C(T_1,T_2,T_3) = \frac{25}{84}\times 14 = 4.1666$$

After agent 1 lie with producible decoy letter to node E, the only cost that (apparently) changes is the cost of delivery of the entire set of letters, which is now 18.

They will agree on the following deal:

$$p_1^* = \frac{2C(T_1^*)+C(T_1^*,T_2)-C(T_2)+C(T_1^*,T_3)-C(T_3)+2C(T_1^*,T_2,T_3)-2C(T_2,T_3)}{6C(T_1^*,T_2,T_3)} = \frac{2\times9+11.5-8+11.5-8+2\times18-2\times14}{6\times18} = \frac{33}{108} = 0.3055$$

Agent A_1 will simply do the cycle (i.e., without the decoy letter). Thus,

$$\cos t_1^* = p_1^* \times C(T_1,T_2,T_3) = \frac{33}{108}\times 14 = 4.27$$

Thus, lying was not beneficial to agent 1.

<u>Example 4</u>: Consider the following example from the Postman Domain. Let the graph be as in the figure 19.

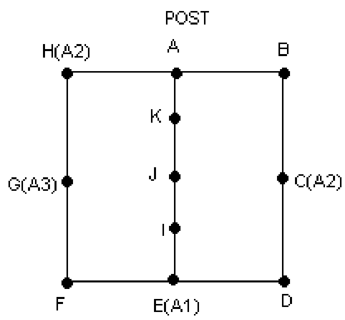

Figure 19

The length (and thus the cost) of each edge in the figure is 1. The post office is at node A. agent 1 must deliver letters to node E, agent 2 must deliver letters to nodes C and H, and agent 3 must deliver letters to node G.

$c(T_1) = 8$, $c(T_2) = 6$, $c(T_3) = 4$, $c(T_1 \cup T_2) = c(T_1 \cup T_3) = c(T_2 \cup T_3) = c(T_1 \cup T_2 \cup T_3) = 8$.

According to the Shapely Value-based mechanism for subadditive TODs, the agents will agree on the following all-or-nothing deal, (p_1, p_2, p_3), such that:

$$p_1 = \frac{2C(T_1) + C(T_1, T_2) - C(T_2) + C(T_1, T_3) - C(T_3) + 2C(T_1, T_2, T_3) - 2C(T_2, T_3)}{6C(T_1, T_2, T_3)} = \frac{2\times 8 + 8 - 6 + 8 - 4 + 2\times 8 - 2\times 8}{6\times 8} = \frac{22}{48} = 0.4583$$

$$\cos t_1 = p_1 \times C(T_1, T_2, T_3) = \frac{22}{48} \times 8 = 3.6666$$

After agent 1 lie with producible decoy letter to node J, $c(T_1^*) = 8$, $c(T_2) = 6$, $c(T_3) = 4$,

$c(T_1^*, T_2) = 10$, $c(T_1^*, T_3) = c(T_2, T_3) = 8$ and $c(T_1^*, T_2, T_3) = 12$.

They will agree on the following deal:

$$p_1^* = \frac{2C(T_1^*) + C(T_1^*, T_2) - C(T_2) + C(T_1^*, T_3) - C(T_3) + 2C(T_1^*, T_2, T_3) - 2C(T_2, T_3)}{6C(T_1^*, T_2, T_3)} = \frac{2\times 8 + 10 - 6 + 8 - 4 + 2\times 12 - 2\times 8}{6\times 12} = \frac{32}{72} = 0.4444$$

Agent A_1 will simply do the cycle (i.e., without the decoy letter). Thus,

$$\cos t_1^* = p_1^* \times C(T_1, T_2, T_3) = \frac{32}{72} \times 8 = 3.55$$

Thus, lying was beneficial to agent 1.

Example 5: Consider again the last encounter from the Postman Domain. However, this time agent 3 needs to deliver an additional letter to node B.

This time $c(T_1) = 8$, $c(T_2) = c(T_3) = 6$, $c(T_1 \cup T_2) = c(T_1 \cup T_3) = c(T_2 \cup T_3) = c(T_1 \cup T_2 \cup T_3) = 8$.

The agents will agree on the following deal:

$$p_1 = \frac{2\times 8 + 8 - 6 + 8 - 6 + 2\times 8 - 2\times 8}{6\times 8} = \frac{20}{48} = 0.416$$

$$\cos t_1 = p_1 \times C(T_1, T_2, T_3) = \frac{20}{48} \times 8 = 3.333$$

Again agent 1 lies with a decoy letter to node J. $c(T_1^*) = 8$, $c(T_2) = c(T_3) = 6$,

$c(T_1^* \cup T_2) = c(T_1^* \cup T_3) = 10$, $c(T_2 \cup T_3) = 8$ and $c(T_1^* \cup T_2 \cup T_3) = 12$.

They will agree on the following deal:

$$p_1^* = \frac{2\times 8 + 10 - 6 + 10 - 6 + 2\times 12 - 2\times 8}{6\times 12} = \frac{32}{72} = 0.4444$$

Agent A_1 will simply do the cycle (i.e., without the decoy letter). Thus,

$$\cos t_1^* = p_1^* \times C(T_1, T_2, T_3) = \frac{32}{72} \times 8 = 3.55$$

Thus, this time the lie is not beneficial to agent 1.

Example 6: Consider another example from the Postman Domain given in the following figure. The length (and thus the cost) of each edge is written next to it. The post office is at node A.

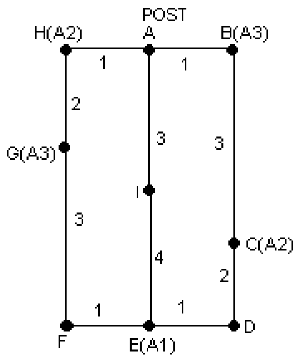

Figure 20

A_1 needs to deliver a letter to node E, A_2 needs to deliver letters to nodes C and H and agent A_3 needs to deliver letters to nodes B and G.

$c(T_1) = 14$, $c(T_2) = 10$, $c(T_3) = 8$, $c(T_1 \cup T_2) = c(T_1 \cup T_3) = c(T_2 \cup T_3) = c(T_1 \cup T_2 \cup T_3) = 14$.

The agents will agree upon a deal in which:

$$p_1 = \frac{2C(T_1) + C(T_1, T_2) - C(T_2) + C(T_1, T_3) - C(T_3) + 2C(T_1, T_2, T_3) - 2C(T_2, T_3)}{6C(T_1, T_2, T_3)} = \frac{38}{84} = 0.452$$

$$\cos t_1 = p_1 \times C(T_1, T_2, T_3) = \frac{38}{84} \times 14 = 6.333$$

As usual agent 1 lies. This time he claims that he has a letter to I.

$c(T_1^*) = 14$, $c(T_2) = 10$, $c(T_3) = 8$, $c(T_1^* \cup T_2) = c(T_1^* \cup T_3) = 16$, $(T_2 \cup T_3) = 14$ and $c(T_1^* \cup T_2 \cup T_3) = 20$.

They will agree on the following deal:

$$p_1^* = \frac{2C(T_1^*) + C(T_1^*, T_2) - C(T_2) + C(T_1^*, T_3) - C(T_3) + 2C(T_1^*, T_2, T_3) - 2C(T_2, T_3)}{6C(T_1^*, T_2, T_3)} = \frac{54}{120} = 0.45$$

Agent A_1 will simply do the cycle (i.e., without the decoy letter). Thus,

$$\cos t_1^* = p_1^* \times C(T_1, T_2, T_3) = \frac{54}{120} \times 14 = 6.3$$

Thus, this time the lie is beneficial to agent 1.

Example 7: Look at the following example, which seems similar to the previous one.

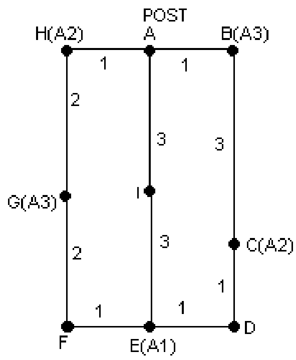

Figure 21

$c(T_1) = 12$, $c(T_2) = 10$, $c(T_3) = 8$, $c(T_1 \cup T_2) = c(T_1 \cup T_3) = c(T_2, \cup T_3) = c(T_1 \cup T_2 \cup T_3) = 12$.

They will agree on a deal in which:

$$p_1 = \frac{2\times12+12-10+12-8+2\times12-2\times12}{6\times12} = \frac{30}{72} = 0.416$$

$$\cos t_1 = p_1 \times C(T_1, T_2, T_3) = \frac{30}{72} \times 12 = 5$$

After agent 1 declares a decoy letter to I.

$c(T_1^*) = 12$, $c(T_2) = 10$, $c(T_3) = 8$, $c(T_1^* \cup T_2) = c(T_1^* \cup T_3) = 14$, $c(T_2 \cup T_3) = 13$ and $c(T_1^* \cup T_2 \cup T_3) = 28$

The agents will agree on the following all-or-nothing deal:

$$p_1^* = \frac{2\times12+14-10+14-8+2\times18-2\times12}{6\times18} = \frac{46}{108} = 0.425$$

$$\cos t_1^* = p_1^* \times C(T_1, T_2, T_3) = \frac{46}{108} \times 12 = 5.111$$

$\Rightarrow$ The lie is not beneficial to agent 1 this time.

5.2.2 Analysis

The examples we have just seen differ from one another in subtle ways. But, in spite of the similarity, some of them allow beneficial deception and some do not.

Example 1 and example 3 have many things in common. In both cases, the cost of T_1 is greater than the cost of T_2 and T_3 (the opponents), and thus the cost of T_1 is greater than the average of the costs of the opponents. Moreover, in both cases the lie causes a change only in the cost of delivering the entire set of letters, and not in any partial set of tasks. And in spite of all the above, the lie is beneficial in example 1, and harmful in example 3. Why?

Well, there *is* a difference between the examples. It stems from the expression $\{2c(T_1)+c(T_1 \cup T_2)+c(T_1 \cup T_3)-c(T_2)-c(T_3)-2c(T_2 \cup T_3)\}$. This expression gets a positive value in example 1 and a negative value in example 3. It seems like a minor difference, but it makes the whole difference…

Theorem 30: For any encounter in a multi-agent subadditive TOD, using the Shapley Value-based mechanism over all-or-nothing deals, if agent A_k generates a decoy lie L such that

$$[C(N^*)-C(N)]\times\left\langle\sum_{S\subset N,S\neq N,k\in S}(n-|s|)!|s-1|!C(S)-\sum_{S\subset N,k\notin S}(n-|s|-1)!|s|!C(S)\right\rangle\geq$$
$$C(N)\times\left[\sum_{S\subset N,S\neq N,k\in S}(n-|s|)!|s-1|!\left\{C(S^*)-C(S)\right\}\right]$$

then L may benefit agent A_k and cannot harm him.

Proof:

According to the Shapley Value-based mechanism, the agent will agree on an all-or-nothing deal,

$(p_1, \ldots, p_n)$, such that:

$$p_i=\frac{\sum_{S\subset N,i\notin S}\frac{(n-|s|-1)!|s|!}{n!}[C(S\cup\{i\})-C(S)]}{C(N)}$$

Thus,

$$p_k=\frac{\sum_{S\subset N,k\notin S}(n-|s|-1)!|s|![C(S\cup\{k\})-C(S)]}{n!C(N)}=$$
$$\frac{(n-1)!C(N)+\sum_{S\subset N,S\neq N,k\in S}(n-|s|)!|s-1|!C(S)-\sum_{S\subset N,k\notin S}(n-|s|-1)!|s|C(S)}{n!C(N)}=$$
$$\frac{1}{n}+\frac{\sum_{S\subset N,S\neq N,k\in S}(n-|s|)!|s-1|!C(S)-\sum_{S\subset N,k\notin S}(n-|s|-1)!|s|C(S)}{n!C(N)}$$

and $\cos t_k = p_k \times C(N)$.

If agent A_k declares T_k^* which is a superset of T_k (i.e., T_k with extra decoy tasks), then they will agree on p^* that satisfies

$$p_k^*=\frac{1}{n}+\frac{\sum_{S\subset N,S\neq N,k\in S}(n-|s|)!|s-1|!C(S^*)-\sum_{S\subset N,k\notin S}(n-|s|-1)!|s|C(S^*)}{n!C(N^*)}$$

Since A_k won't have to carry out the decoy tasks, $\cos t_k^* = p_k^* \times C(N)$.

In order to show that A_k may only benefit from L, it will be sufficient to show that $\cos t_k \geq \cos t_k^*$.

$$\cos t_k \geq \cos t_k^*$$

$$p_k \geq p_k^*$$

$$\frac{1}{n}+\frac{\sum_{S\subset N,S\neq N,k\in S}(n-|s|)!|s-1|!C(S)-\sum_{S\subset N,k\notin S}(n-|s|-1)!|s|!C(S)}{n!C(N)}\geq$$
$$\frac{1}{n}+\frac{\sum_{S\subset N,S\neq N,k\in S}(n-|s|)!|s-1|!C(S^*)-\sum_{S\subset N,k\notin S}(n-|s|-1)!|s|!C(S^*)}{n!C(N^*)}$$

$$\frac{\sum_{S\subset N,S\neq N,k\in S}(n-|s|)!|s-1|!C(S)-\sum_{S\subset N,k\notin S}(n-|s|-1)!|s|!C(S)}{C(N)}\geq$$
$$\frac{\sum_{S\subset N,S\neq N,k\in S}(n-|s|)!|s-1|!C(S^*)-\sum_{S\subset N,k\notin S}(n-|s|-1)!|s|!C(S^*)}{C(N^*)}$$

$$\frac{\sum_{S\subset N,S\neq N,k\in S}(n-|s|)!|s-1|!C(S)-\sum_{S\subset N,k\notin S}(n-|s|-1)!|s|!C(S)}{C(N)}\geq$$
$$\frac{\sum_{S\subset N,S\neq N,k\in S}(n-|s|)!|s-1|!C(S^*)-\sum_{S\subset N,k\notin S}(n-|s|-1)!|s|!C(S)}{C(N^*)}$$

$$\left[\sum_{S\subset N,S\neq N,k\in S}(n-|s|)!|s-1|!C(S)-\sum_{S\subset N,k\notin S}(n-|s|-1)!|s|!C(S)\right]\times C(N^*)\geq$$
$$\left[\sum_{S\subset N,S\neq N,k\in S}(n-|s|)!|s-1|!C(S^*)-\sum_{S\subset N,k\notin S}(n-|s|-1)!|s|!C(S)\right]\times C(N)$$

$$\left[\sum_{S\subset N,S\neq N,k\in S}(n-|s|)!|s-1|!C(S)-\sum_{S\subset N,k\notin S}(n-|s|-1)!|s|!C(S)\right]\times\left(C(N)+\left(C(N^*)-C(N)\right)\right)\geq$$
$$\left[\sum_{S\subset N,S\neq N,k\in S}(n-|s|)!|s-1|!C(S^*)-\sum_{S\subset N,k\notin S}(n-|s|-1)!|s|!C(S)\right]\times C(N)$$

$$\left[\sum_{S\subset N,S\neq N,k\in S}(n-|s|)!|s-1|!C(S)-\sum_{S\subset N,k\notin S}(n-|s|-1)!|s|!C(S)\right]\times C(N)+$$
$$\left[\sum_{S\subset N,S\neq N,k\in S}(n-|s|)!|s-1|!C(S)-\sum_{S\subset N,k\notin S}(n-|s|-1)!|s|!C(S)\right]\times\left\{C(N^*)-C(N)\right\}\geq$$
$$\left[\sum_{S\subset N,S\neq N,k\in S}(n-|s|)!|s-1|!C(S^*)-\sum_{S\subset N,k\notin S}(n-|s|-1)!|s|!C(S)\right]\times C(N)$$

$$\left[\sum_{S\subset N,S\neq N,k\in S}(n-|s|)!|s-1|!C(S)-\sum_{S\subset N,k\notin S}(n-|s|-1)!|s|!C(S)\right]\times\left\{C(N^*)-C(N)\right\}+$$
$$\left[\sum_{S\subset N,S\neq N,k\in S}(n-|s|)!|s-1|!C(S)\right]\times C(N)\geq\left[\sum_{S\subset N,S\neq N,k\in S}(n-|s|)!|s-1|!C(S^*)\right]\times C(N)$$

$$\left[\sum_{S\subset N,S\neq N,k\in S}(n-|s|)!|s-1|!C(S)-\sum_{S\subset N,k\notin S}(n-|s|-1)!|s|!C(S)\right]\times\left\{C(N^*)-C(N)\right\}\geq$$
$$\left[\sum_{S\subset N,S\neq N,k\in S}(n-|s|)!|s-1|!\left\{C(S^*)-C(S)\right\}\right]\times C(N)$$

which is true.

This theorem characterizes the relationship among agents' costs that allow for beneficial deception. The problem with this characterization is that it requires from the potential liar a lot of computational effort and a lot of information about the other agents' set of tasks. Thus, it may be computationally difficult to find beneficial lies and even impossible given symmetric incomplete information between the agents.

In general, it is very difficult to find a non-computationally complex characterization of decoy lies that are guaranteed to be beneficial. Therefore, we explore sufficient conditions that *prevent* an agent from having a beneficial decoy lie.

First, we present a mathematical characterization of the relationship among costs of tasks that prevent an agent from having a beneficial decoy lie.

Theorem 31: For any encounter in a multi-agent subadditive TOD, using the Shapley Value-based mechanism over all-or-nothing deals, if agent A_k generates a decoy lie L such that

$$[C(N^*)-C(N)]\times\left\langle\sum_{S\subset N,S\neq N,k\in S}(n-|s|)!|s-1|!C(S)-\sum_{S\subset N,k\notin S}(n-|s|-1)!|s|!C(S)\right\rangle\leq$$
$$C(N)\times\left[\sum_{S\subset N,S\neq N,k\in S}(n-|s|)!|s-1|!\left\{C(S^*)-C(S)\right\}\right]$$

then L may harm agent A_k and cannot benefit him.

Proof:

The proof of this theorem is very similar to the proof of theorem 30. We can use the same steps, and the only difference will be that we will change the "≥" symbol to "≤".

From theorem 26 we know that "decoy" lie are never beneficial in concave domains. Thus, concavity is a sufficient condition that prevents an agent from having a beneficial decoy lie. Here, we refine the concept of concavity to speak of "a concave encounter" and of "an encounter which is concave at agent k".

<u>Definition 16</u>: An encounter ($T_1, T_2, \ldots, T_n$) within a TOD $<T, A, C>$ is called *concave* if for all $S \subseteq N$ and for all finite set of tasks $X \subseteq T$, we have $C(N \cup X)-C(N) \leq C(S \cup X)-C(S)$.

<u>Theorem 32</u>: For any concave encounter in a multi-agent subadditive TOD, using the Shapley Value-based mechanism over all-or-nothing deals, every "decoy" lie is not beneficial.

<u>Proof</u>:

Let ($T_1, T_2, \ldots, T_n$) be the true encounter. Assume that agent A_k lies and declares that his initial set of tasks is $T_k^* = T_k \cup L$, where L is a set of decoy tasks. We know from theorem 31 that in order to show that A_k cannot benefit from L, it will be sufficient to show that:

$$[C(N^*)-C(N)]\times\left\langle\sum_{S\subset N,S\neq N,k\in S}(n-|s|)!|s-1|!C(S)-\sum_{S\subset N,k\notin S}(n-|s|-1)!|s|!C(S)\right\rangle \leq$$
$$C(N)\times\left[\sum_{S\subset N,S\neq N,k\in S}(n-|s|)!|s-1|!\{C(S^*)-C(S)\}\right]$$

Which is equivalent to,

$$[C(N\cup L)-C(N)]\times\left\langle\sum_{S\subset N,S\neq N,k\in S}(n-|s|)!|s-1|!C(S)-\sum_{S\subset N,k\notin S}(n-|s|-1)!|s|!C(S)\right\rangle \leq$$
$$C(N)\times\left[\sum_{S\subset N,S\neq N,k\in S}(n-|s|)!|s-1|!\{C(S\cup L)-C(S)\}\right]$$

Since the encounter is concave $\forall S \subseteq N$, $C(N\cup L)-C(N) \leq C(S\cup L)-C(S)$.

Therefore, it will be sufficient to show that:

$$[C(N\cup L)-C(N)]\times\left\langle\sum_{S\subset N,S\neq N,k\in S}(n-|s|)!|s-1|!C(S)-\sum_{S\subset N,k\notin S}(n-|s|-1)!|s|!C(S)\right\rangle \leq$$
$$C(N)\times\left[\sum_{S\subset N,S\neq N,k\in S}(n-|s|)!|s-1|!\{C(N\cup L)-C(N)\}\right]$$

$$\sum_{S\subset N,S\neq N,k\in S}(n-|s|)!|s-1|!C(S)-\sum_{S\subset N,k\notin S}(n-|s|-1)!|s|!C(S) \leq$$
$$C(N)\times\left[\sum_{S\subset N,S\neq N,k\in S}(n-|s|)!|s-1|!\right]$$

It will be sufficient to show that:

$$\sum_{S\subset N,S\neq N,k\in S}(n-|s|)!|s-1|!C(S) \leq C(N)\times\left[\sum_{S\subset N,S\neq N,k\in S}(n-|s|)!|s-1|!\right]$$

From the monotonicity of function c: $\forall S \subseteq N$, $C(S) \leq C(N)$.

Thus, it will be sufficient to show that

$$\sum_{S\subset N,S\neq N,k\in S}(n-|s|)!|s-1|!C(N) \leq C(N)\times\left[\sum_{S\subset N,S\neq N,k\in S}(n-|s|)!|s-1|!\right]$$

$$\sum_{S\subset N,S\neq N,k\in S}(n-|s|)!|s-1|! \leq \sum_{S\subset N,S\neq N,k\in S}(n-|s|)!|s-1|!$$

Which is (of course) true.

<u>Definition 17</u>: An encounter $(T_1, T_2, \ldots, T_n)$ within a TOD $<T, A, C>$ is called *concave at agent k*, if for all $S \subseteq N$ such that $k \in S$ and for all finite set of tasks $X \subseteq T$, we have $C(N \cup X) - C(N) \leq C(S \cup X) - C(S)$.

<u>Theorem 33</u>: For any encounter in a multi-agent subadditive TOD, using the Shapley Value-based mechanism over all-or-nothing deals, which is concave at agent k, every "decoy" lie is not beneficial to agent k.

<u>Proof</u>:

The proof is almost identical to the proof of theorem 32. The only difference is that we use the "concavity of the encounter at agent k" instead of the "concavity of the encounter".

6 Bibliography

[1] A. H. Bond and L. Gasser, editors. *Readings in Distributed Artificial Intelligence.* Morgan Kaufmann Publishers, Inc., San Mateo, California. 1988.

[2] N. Camiel, S. London, N. Nisan and O. Regev. The POPCORN project – An interim report. Distributed computation over the internet in Java. *Sixth International World Wide Web Conference.* Santa –Clara. 1997.

[3] K. S. Decker and V. R. Lesser. Designing a family of coordination algorithms. In Michael N. Huhns and Munindar P. Singh, editors, *Readings in agents, pages 450-457.* Morgan Kaufmann Publishers, Inc., San Francisco, California. 1997.

[4] E. H. Durfee and V. R. Lesser. Using partial global plans to coordinate distributed solvers. In Alan H. Bond and Les Gasser, editors, *Readings in Distributed Artificial Intelligence, pages 285-293.* Morgan Kaufmann Publishers, Inc., San Mateo, California. 1988.

[5] E. Ephrati and J. S. Rosenschein. The Clarke tax as consensus mechanism among automated agents. In proceedings of the Ninth National Conference on Artificial Intelligence. Boston. 1991.

[6] E. Ephrati and J. S. Rosenschein. Multi-Agent planning as a dynamic search for social consensus. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence, pages 423-429.* 1993.

[7] P. J. Gymtrasiewics and E. H. Durfee. A rigorous, operational formalization of recursive modeling. In *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS), pages 125-132*. 1995.

[8] B. Grosz and S. Kraus. Collaborative Plans for Complex Group Action. *Artificial Intelligence, 86(2), pages 269-357.* 1996.

[9] S. Ketchpel. Coalition formation among autonomous agents. In *Pre-Proceedings of the fifth European Workshop on "Modeling Autonomous Agents in Multi-Agent World" Neuchatel, Switzerland. 1993.*

[10] S. Ketchpel. Forming coalitions in the face of uncertain rewards. In *Proceedings of the Twelfth National Conference on Artificial Intelligence, pages 414-419.* 1994.

[11] S. Kraus, J. Wilkenfeld and G. Zlotkin. Multiagent Negotiation under time constraints. *Artificial Intelligence, 75(2), pages 297-345.* 1995.

[12] S. Kraus. An overview of Incentive Contracting. *Artificial Intelligence Journal 83(2), pages 297-346.* 1996.

[13] R. D. Luce and H. Raiffa. *Games and Decisions.* John Wiley and sons, Inc., New York. 1957.

[14] T. W. Malone, R. E. Fikes, K. R. Grant and M. T. Howard. Enterprise: A market-like task scheduler for distributed computing environments. In B. A. Huberman, editor, *The Ecology of Computation.* North-Holland Publishing Company, Amsterdam. 1988.

[15] M. Maschler. *Game Theory.* Lectures' Summary. 1970. (Hebrew).

[16] Y. Moses and M. Tennenholtz. Artificial social systems part one: basic principles. *Technical Report CS90-12, Weizmann Institute.* 1990.

[17] J. S. Rosenschein and G. Zlotkin. *Rules of encounter.* MIT press. 1994.

[18] T. W. Sandholm and V. R. Lesser. Coalitions between computationally bounded agents. *Artificial Intelligence 94, pages 99-137.* 1997.

[19] T. W. Sandholm. *Introduction to distributed artificial intelligence.* MIT press. (Draft). 1998.

[20] L. S. Shapley. A value for n-person games. In H. W. Kunn and A. W. Tucker, editors, *Contributions to the theory of games 2, Annals of mathematics studies 28, (Princeton University Press, Princeton N.J.), pages 307 –317.* 1953.

[21] L. S. Shapley. Cores of convex games. In *International Journal of Game Theory 1, pages 11-26.* 1971.

[22] O. Shechory and S. Kraus. Task allocation via coalition formation among autonomous agents. In Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence, pages 655-661. 1995.

[23] O. Shechory and S. Kraus. Formation of overlapping coalitions for precedence-ordered task-execution among autonomous agents. . In *Proceedings of the Second International Conference on Multi-Agent Systems, pages 330-337.* 1996.

[24] O. Shechory and S. Kraus. Emergent cooperative goal-satisfaction in large-scale automated agent systems. In *proceedings of European Conference on Artificial Intelligence, pages 544-548.* 1996.

[25] O. Shechory and S. Kraus. A kernel-oriented model for coalition formation in general environments: Implementation and results.). In *proceedings of the National Conference on Artificial Intelligence, pages 134-140.* 1996.

[26] Y. Shoham and M. Tennenholtz. On the synthesis of useful social laws for artificial agent societies (preliminary report). In *proceedings of the National Conference on Artificial Intelligence,* Sun Jose, California. 1992.

[27] Y. Shoham and M. Tennenholtz. On Social Laws for Artificial Agent Societies: Off-Line Design. *Artificial Intelligence Journal 73.* 1993.

[28] R. G. Smith. The Contract Net Protocol: high level communication and control in a distributed problem solver. In Alan H. Bond and Les Gasser, editors, *Readings in Artificial Intelligence.* Morgan Kaufmann Publishers, Inc., San Mateo, California. 1988.

[29] K. P. Sycara. Resolving goal conflicts via negotiation. In *proceedings of the Seventh National Conference on Artificial Intelligence, pages 245-250.* 1988.

[30] M. P. Wellman. A market-oriented programming environment and its application to distributed multicommodity flow problems. *Journal Artificial Intelligence Research 1(1) pages 1-23.* 1993.

[31] M. P. Wellman. Market-oriented programming: Some early lessons, In S. Clearwater, editor, Market-Based Control: A paradigm for distributed resource allocation. *World Scientific.* 1996.

[32] 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, pages 432-437.* 1994.

[33] Greg Elofson and William N. Robinson. Creating a Custom Mass-Production Channel on the Internet, *Communications of the ACM,* March 1998, pp. 56-62.

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

[35] T. Sandholm and V. Lesser. Coalition Formation among Bounded Rational Agents. 14th International Joint Conference on Artificial Intelligence (IJCAI-95), Montreal, Canada, pp. 662-669. 1995.

[36] Onn Shehory and Sarit Kraus. Coalition formation among autonomous agents: Strategies and complexity. *Pre-Proceedings of the Fifth European Workshop on Modeling Autonomous Agents in a Multi-Agent World*. 1993.

[37] Onn Shehory and Sarit Kraus. Methods for Task Allocation via Agent Coalition Formation. *Artificial Intelligence*. Volume 101, Numbers 1-2, May 1998, pp. 165-200.

[38] H. P. Young. Individual contribution and just compensation. In A. E. Roth, ed., *The Shapley Value*. Cambridge: Cambridge University Press, Chapter 17, pp. 267-278. 1988.

[39] Gilad Zlotkin and Jeffrey S. Rosenschein. Mechanism Design for Automated Negotiation, and its Application to Task Oriented Domains, *Journal of Artificial Intelligence*. Volume 86, Number 2, October 1996, pages 195--244.

7 Annotated Bibliography

[1] N. Camiel, S. London, N. Nisan and O. Regev. The POPCORN project – An interim report. Distributed computation over the internet in Java. Sixth International World Wide Web Conference. Santa –Clara. 1997.

The POPCORN project provides an infrastructure for globally distributed computation over the entire Internet. It provides any programmer connected to the Internet with a single huge virtual parallel computer composed of all processors on the Internet which care to participate at any given moment. A market-based mechanism of trade of CPU time underlies the system, so as to motivate processors to provide their CPU cycles for other people's computations.

In our work we have been examining the problem of coalition formation. The POPCORN project is an example of a possible use of this theory.

[2] E. Ephrati and J. S. Rosenschein. The Clarke tax as consensus mechanism among automated. In proceedings of the Ninth National Conference on Artificial Intelligence. Boston. 1991.

Ephrati and Rosenschein presented a voting mechanism called the Clarke Task mechanism. In a voting setting, each agent expresses its preferences, and a group choice mechanism is used to select the result. This result is enforced so that all agents have to abide by the solution prescribed by the mechanism. Ephrati and Rosenschein have also considered the case where the preferences of the agents are not known. In this case the agents have to declare their preferences. However, the preferences declared might be different from the true preferences. The Clarke tax procedure assures that every agent should always tell his true preferences values.

Ephrati and Rosenschein have used a variety of different mechanisms to reach consensus among agents. Using these mechanisms, they considered the case of lack of information and they have chosen a protocol that motivates the agents to tell the truth, as we try to do with coalition formation.

[3] P. J. Gymtrasiewics and E. H. Durfee. A rigorous, operational formalization of recursive modeling. In *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS), pages 125-132*. 1995.

Another method to coordinate the activities of autonomous agents is the *Recursive Modeling Method (RMM)*. RMM views a multi-agent situation from the perspective of an agent that is individually trying to decide what actions it should do right now. To make decisions, RMM uses a decision-theoretic paradigm of rationality, where an agent attempts to maximize its expected utility given its beliefs. The basic building block of RMM's representation is a payoff matrix that expresses the agent's beliefs about its environment. In order to solve its own decision-making situation, the agent needs an idea of what the other agents are likely to do. This can be done by representing what it knows about the other agents' decision-making situations, thus modeling them in terms of their payoff matrices. The fact that other agents could also be modeling others, including the original agent, leads to a recursive model.

Thus, the RMM is another coordination model based on game theory which assumes that different agents may have different knowledge about the world.

[4] S. Ketchpel. Forming coalitions in the face of uncertain rewards. In *Proceedings of the Twelfth National Conference on Artificial Intelligence, pages 414-419.* 1994.

The work of Ketchpel also addresses the problem of forming coalitions between self-interested agents in super-additive environments. Ketchpel noted that the coalition formation problem is related to the stable marriage problem and makes use of algorithms for the latter problem. Ketchpel points out that in reality the value that a coalition obtains may be unknown. In fact, each agent might have a different expectation for the value of the collaboration, and both values may be different from the true utility. He suggests a modification to the algorithm to solve this problem.

The problem that Ketchpel refers to is derived from the fact that the agents don't have full information; this is also what makes lying possible, and this is exactly the problem we address in our work.

[5] J. S. Rosenschein and G. Zlotkin. *Rules of encounter.* MIT press. 1994.

Rosenschein and Zlotkin use tools from game theory for analyzing different negotiation mechanisms among *pairs* of agents. Rosenschein and Zlotkin developed a domain theory for negotiation. The aim of this theory is to classify domains where agents can operate in a way that an appropriate negotiation mechanism could be chosen. They categorized classes of domains into a three-level hierarchy, where each level is increasingly more general. For each domain, they designed a negotiation protocol (a negotiation protocol means the public rules of the negotiation). A protocol specifies the kind of deals the agent can make, as well as the sequence of offers and counter-offers that they are allowed.

They also observed that common knowledge is somewhat unrealistic. In order to get around the lack of information, Rosenschein and Zlotkin suggest that the agents will explicitly declare their tasks and resources. In this case the agents may lie in order to gain in the subsequent negotiation. Rosenschein and Zlotkin have analyzed when rational agents are motivated to declare truthfully.

In our work, we extend their work from the two-agent case to the multiple agent case.

[6] T. W. Sandholm and V. R. Lesser. Coalitions between computationally bounded agents.

Sandholm and Lesser's research also discusses coalition. What makes their work unique is the fact that they relax the assumption of perfect rationality. In their setting, the agents lack full rationality. Thus, the deduction process itself is costly: either an agent has to pay for computational resources or the computation with his own limited resources takes time. Based on the agents' problem-solving algorithms, Sandholm and Lesser state which agents should form coalitions and which coalition structures are stable. These prescriptions differ significantly from those for fully rational agents.

[7] L. S. Shapley. A value for n-person games. In H. W. Kunn and A. W. Tucker, editors, *Contributions to the theory of games 2, Annals of mathematics studies 28, (Princeton University Press, Princeton N.J.), pages 307-317. 1953.*

Coalition formation has been widely studied in game theory. Many of the solution concepts for coalition formation are static. They address the question of how to divide the payoffs among the agents. But being static in nature, they do not usually address the dynamics of the coalition formation process. Nevertheless, models of game theory can be used as a basis for the agents' interaction protocols. Automated agents in DAI can be modeled as players of game-theoretic models. Analyzing the coordination between agents as a game gives a stronger theoretical basis to the research. In this work we have also used tools from game theory.

[8] O. Shechory and S. Kraus. Task allocation via coalition formation among autonomous agents. In Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence, pages 661-665. 1995.

Shechory and Kraus present an algorithm for coalition structure generation among cooperative – social welfare maximizing, i.e., not self-interested – agents. They consider situations where each task should be attached to a group of agents, which will perform the task. Since task allocation among agents may be approached as assigning groups of agents to tasks, the partition of agents into subgroups becomes the main issue. Therefore, the task allocation problem becomes similar to the *set-partitioning problem* (SPP). The SPP is finding a partition of a set into subsets, so that the partition has minimal cost. The SPP is an NP-hard problem. This implies a computational complexity which is too high. However, a variety of heuristic algorithms for solving this problem have been suggested. Among them, is the algorithm of Chvatal. Based on this algorithm, Shechory and Kraus present a greedy distributed set-partitioning algorithm for autonomous agents that work as DPS system.

[9] O. Shechory and S. Kraus. A kernel-oriented model for coalition formation in general environments: Implementation and results. In *proceedings of the National Conference on Artificial Intelligence, pages 134-140.* 1996.

Shechory and Kraus analyze coalition formation among rational, self-interested agents in environments, which are not necessarily superadditive.

They present a distributed, negotiation-based, polynomial, kernel-oriented algorithm. The algorithm consists of steps in which coalitions transmit, accept, and reject proposals for creating new coalitions. The algorithm starts with all the agents in single-membered coalitions. At each step, at least one coalition will attempt to improve the payoffs of its members by making a coalition formation proposal to another coalition. The acceptance of such a proposal will improve the situation of the agents involved. The algorithm may continue until all the proposals of all the coalitions are rejected or until a steady state has been reached.

The work of Shechory and Kraus considers a more general environment than ours, but it assumes full information, which is generally not realistic.

[10] 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, pages 432-437.* 1994.

Zlotkin and Rosenschein analyze coalitions between rational, self-interested agents. They discuss the special case of subadditive task-oriented domains, which are a strict subset of task-oriented domains. In such domains, agents will always form the *grand coalition* (i.e., one coalition in which all the agents are members), since this coalition structure maximizes the utility that the agents can derive. The main problem in such domains is how to divide the joint utility between the agents. Zlotkin and Rosenschein suggest a mechanism in which each agent will handle the tasks of all agents with a certain probability. This probability is set according to the Shapley value, in a way that guarantees each agent an expected utility that equals its Shapley value.

A naive method that guarantees each agent an expected utility that equals each Shapley value has exponential complexity in the number of agents, but Zlotkin and Rosenschein present a novel cryptographic method for achieving this with linear complexity in the number of agents.

In this work we consider their mechanism when agents don’t have full information about each other’s goals.