

A Robust Deception-Free Coalition Formation Model

Maria-Victoria Belmonte	Ricardo Conejo	J.L. Pérez-de-la-Cruz	Francisco Triguero
Dpto. Lenguajes y C.Comp.	Dpto. Lenguajes y C.Comp.	Dpto. Lenguajes y C.Comp.	Dpto. Lenguajes y C. Comp.
Universidad de Málaga	Universidad de Málaga	Universidad de Málaga	Universidad de Málaga
Campus de Teatinos, 29080	Campus de Teatinos, 29080	Campus de Teatinos, 29080	Campus de Teatinos, 29080
Málaga, SPAIN	Málaga, SPAIN	Málaga, SPAIN	Málaga, SPAIN
+34 52 132809	+34 52 132812	+34 52 132801	+3452 131438
mavi@lcc.uma.es	conejo@lcc.uma.es	perez@lcc.uma.es	triguero@ctima.uma.es

ABSTRACT

We study two properties of coalition formation algorithms, very important for their application in real-life scenarios: robustness and tolerance to some agent misbehaviors. The study is performed for a previously proposed coalition formation model –based upon game theory for a class of task-oriented problems that guarantees an optimum task allocation and a stable (fair) profit division. The results show acceptable behavior and performance.

Categories and Subject Descriptors

[I.2.11] *Distributed Artificial Intelligence: Coherence and coordination; Intelligent agents.*

General Terms

Experimentation, Algorithms.

Keywords

Multi-Agent Systems, Coalitions, Game Theory, Task-oriented domains.

1. INTRODUCTION

Coalition formation is an important mechanism for cooperation in Multi-Agent Systems. Autonomous agents forming coalitions may improve their profits and abilities to satisfy their goals, sharing their resources and distributing their tasks. The desired goals of a coalition formation process include: i) maximize the agents' profit or utility, ii) divide the total utility among agents in a fair and stable way, such that the agents in the coalition are not motivated to abandon it –stable payment configuration–, iii) do it within a reasonable time and with a reasonable amount of computational efforts. In [1] it was presented a coalition formation model that allows cooperation among autonomous, rational and self-interested agents in a class of superadditive task oriented domains [2]. In these domains agents are in charge of the realization of certain task, but they do it at different cost depending on the agent and the type of task. However, it is possible to form a coalition among agents with a new re-distribution of the task that may allow them to obtain benefits. That model [1] maximizes the benefits and abilities of agents to satisfy their goals, guarantees an optimum task allocation and a stable (fair) profit distribution among the coalition members, and has a polynomial complexity in

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SAC '04, March 14-17, 2004, Nicosia, Cyprus

Copyright 2004 ACM 1-58113-812-1/03/2004...\$5.00.

terms of number of agents.

In this paper¹ we present two empirical studies on the coalition formation model. The first analyses the robustness of the model in the face of failure of one or more system agents. Any service provided by a MAS must be maintained, even with a worst quality, before the failure of processors, communication networks or agents. The second studies the effects of deception and manipulation on the model. Since the agents have some kind of execution autonomy and are self-interested, they can deceive or mislead each other when they reveal their information if they believe that they will obtain more profits doing so. The results will show an acceptable performance of the model even with different failure patterns in the first case and the stability of the coalition formation mechanism in the second one.

The rest of the paper is organized as follows: in the next section we show briefly the proposed coalition formation model. In the third section we study the model behavior in the face of failure of one or more system agents. In section four, we describe one of the possible manipulations that the system agents can carry out when they reveal their private information, and test the effects of it upon the stability of the model. We will finish, in fifth section, with some conclusions and future works.

2. COALITION FORMATION MODEL

Let us consider a set of n agents $N=\{a_1, a_2, \dots, a_n\}$ that can communicate, and such that each agent a_i must perform a certain initial amount t_i^0 of task units (assumed infinitely divisible). Agent a_i can perform a maximum of k_i task units ($t_i^0 \leq k_i$) and the unitary cost is c_i . We will define the surplus capacity h_i of a_i by $h_i = k_i - t_i^0$ ($0 \leq h_i$). The agents can change the amounts of tasks initially assigned to each one, in order to achieve a better global efficiency; let $t_{i,j}$ be the number of transferred task units from a_i to a_j . However, to allow this new deal, they must incur a transfer cost; namely, we will assume that the cost of transferring a task unit from a_i to a_j is $c_{i,j}$. In this way, the unitary profit, $b_{i,j}$, attached to the transfer of a task unit between a_i and a_j is $b_{i,j} = (c_i - c_j) - c_{i,j}$ and $B_{i,j} = b_{i,j} * t_{i,j}$ is the transfer profit. On the other hand, we will assume that coalition formation costs (communication and coordination) are negligible compared to actual task transfer costs. This assumption leads to a superadditive problem.

The characteristic function or coalitional value, $v(S)$, of a game with a set of players N , is the total utility or profit that the members of S can reach by coordinating and acting together. In

¹ This work has been partially supported by the Spanish Government under grant CICYT TIC 2000-1370-C04-02

our domain the coalitional value of any coalition will be stated as: $v(S) = \sum_{i,j \in S / i \neq j} B_{i,j} = \sum_{i,j \in S / i \neq j} b_{i,j} * t_{i,j}^*, \forall i, j \in S / i \neq j$ (1)

where $t_{i,j}^* \forall i,j \in S / i \neq j$, are the task transferences obtained from an optimum task allocation T^* . In order to establish T^* , we state the task allocation as a linear programming problem, in which the costs are minimized subject to the capacity constraints of each agent [1]. If we impose the following constraint: $c_{i,j} + c_{j,k} \geq c_{i,k} \forall i,j,k \in N$, there is always a solution in which any agent is a producer or a consumer but not both (even if this constraint were not fulfilled, it would be possible to apply this algorithm [1]). Producer agents are those that only transfer tasks and consumer agents are those that only receive tasks. Depending on the situation, there could be a fringe agent, a_f , i.e., a producer that does not transfer its entire task, or a consumer that receive task but does not consume its entire surplus.

Payoff division is the process of dividing the utility or benefits of a coalition among its members. The problem is to distribute the utility of the coalition in a fair and stable way, so the agents in the coalition are not motivated to abandon it. Game theory [3] provides different concepts (core, kernel, Shapley value, etc.) for the stability of coalitions. Our proposed payoff division scheme is calculated by means of the concept of marginal profit. Thus, the payment to each agent will be given by the marginal profit with respect to the resource supplied by the agent, and multiplied by the quantity of this resource. Since the concept of marginal profit is really that of partial derivative, the payment vector x , according to the formulae (1), will be computed as follows:

$$x_i = t_i \frac{\partial v}{\partial t_i} + h_i \frac{\partial v}{\partial h_i}, \forall i \in N \quad (2)$$

In [1] we proof that this payment vector computes in one step a stable payoff division in the sense of the core, and that it has a polynomial complexity in terms of n .

3. EXPERIMENTS ABOUT ROBUSTNESS

Any service provided by a MAS must be maintained even with a degraded quality in the face of failures in the processors, in the communication networks or even in the agents. Concretely, agents' failures must be taking into account in coalition formation models. In our case an agent's failure means that the agent stops working without sending any other message (silent failure). The resultant profits should suffer the minimum loss or damage in the case of the failure of one or several agents taking part in the coalition formation process. An alternative could be to calculate the optimal transferences among the survival agents again, and to distribute the profit according to the above section algorithm. However, it may have a high cost. For that reason, we propose an incremental alternative payoff division model with a lower computational cost, and we experimentally check that the degradation with respect to the optimal solution is not really significant.

3.1 Experimental Design Description

A simulation tool has been developed to experiment with our model. This tool calculates the optimum task allocation and the stable payoff division according to our coalition formation model. Experiments were carried out for a number of agents varying from 10 to 50, and with a failure probability ranged from 0.1 to 0.5.

The input data to the problem have been generated randomly. Namely, the capacity k_i was a uniform random value in the interval [MINCAP..MAXCAP] (data here shown correspond to [1..100]); the production cost c_i was a uniform random value in [MINCOST..MAXCOST] (in our case, [5..25]); the assigned task t_i was stated as $\beta * k_i$, where β is a uniform random value in [0..1]; and, finally, to compute transfer costs, c_{ij} , all the agents were placed inside a rectangle with coordinates $(0,0,X_{max},Y_{max})$ and costs were assumed proportional to the Euclidean distance.

The experimental design compares the optimum model results (which produce a stable payoff division in the sense of the core) with the results of an *alternative payoff division scheme*. This schema (algorithm 1) makes a division of the loss utility -due to the failure of the agents- proportional to the initial utility obtained by them in the optimum allocation scheme prior to the falling. In this way, a new payment vector x' is computed for the $n-j$ participant agents after the failure of j of them, but without applying the optimum model of task allocation and payment division again. This alternative schema does not guarantee that x' lies inside in the core but obtains (with a lower computational complexity) a payment vector near to the optimum.

Algorithm 1 Alternative Payoff Division Model

```

1. Goal: computation of a new payment vector,  $x'=(x'_1, x'_2, ..., x'_n)$  after the failure of one of agent called  $a_{falling}$ 
2. Input: optimum payment vector, before the agent failure,  $x=(x_1, x_2, ..., x_n)$ , optimum task transfers for all the system agents,  $t_{h,j} \forall h, j \in N / h \neq j$ , and the attached profit to each task transfer for all the system agents,  $B_{h,j} \forall h, j \in N / h \neq j$ 
3. Output: alternative payment vector  $x'=(x'_1, x'_2, ..., x'_n)$ 

1. Determine the coalition  $S$  in which  $a_{falling}$  participates
2. Settle the  $t_{h,j} / h = falling \vee j = falling$  in which  $a_{falling}$  participates like producer or consumer agent
if  $t_{h,j} = \{\emptyset\}$  then
{
   $x' \leftarrow x$ 
}
else
{
  if  $\sum_{i \in S} x_i = \sum_{\forall h, j \in S / h = falling \vee j = falling} B_{h,j}$  then
/*if the addition of the coalition profits in which  $a_{falling}$  participates is equal to the addition of the transfer profits in which  $a_{falling}$  participates*/
{
     $x'_i \leftarrow 0, \forall i \in S$ 
}
  else
  {
     $trans\_addition \leftarrow \sum_{\forall h, j \in S / h = falling \vee j = falling} B_{h,j}$ 
     $loss\_util \leftarrow trans\_addition - x_{falling}$ 
    if  $loss\_util = 0$  then
    {
       $x'_{falling} \leftarrow 0$ 
    }
    else
    {
      /* $loss\_util$  must be divided among the remaining members of  $S$  that have a utility value distinct to 0. The division is made in a proportional way to the initial utility of these agents*/
       $total \leftarrow \sum_{i \in S} x_i - x_{falling}$ 

```

```

/* p_util is a vector that contains the percentage of the total
utility (total) that obtains each agent into S*/
p_util = (p_util1, p_util2, ..., p_util|S|), where p_utili ←
(100*x_i) / total ∀ i ∈ S / i ≠ falling, otherwise p_utili ← 0
/* d_util is a vector that contains the quantity of utility that will
be discount to each agent into S*/
d_util = (d_util1, d_util2, ..., d_util|S|), where d_utili
← (loss_util * p_utili) / 100 ∀ i ∈ S / i ≠ falling,
otherwise d_utili ← 0
x'_falling ← 0, and ∀ i ∈ S / i ≠ falling, x'_i ← x_i - d_utili
}
}
}

```

3.2 Report of Results

We have performed several hundreds of runs of the simulation. Mean values are shown here. Figure 1 shows a graphic that compares the addition of the utilities of the system agents after the failure of one or several of them. The abscise axis represents the number of agents, n , and the ordinates axis the addition of the utilities of the system agents after the failure. The graphic shows two lines for each failure probability of the agents (f) (value ranged from 0.1 to 0.3): *util-opt*, the utility addition of the system agents after applying our coalition formation model after the failure; and *sub-opt*, the utility addition after the failure, but applying, in this case, the alternative payoff division model. From the above results (figure 1) we can conclude that: (a) *util-opt* is always better than *sub-opt*; (b) *util-opt* and *util-sub* grows as a function of n ; (c) when n is increased the difference between *util-opt* and *util-sub* is also increased. In figure 1 we can appreciate that *util-opt* and *util-sub* lines shape a triangle that grows quickly as a function of n .

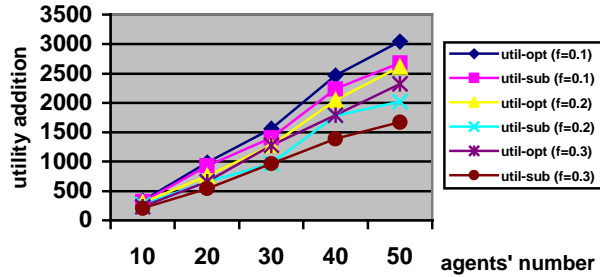


Figure 1. Utility addition of the system agents

Figure 2 shows the same parameters that the above one, but changing the performance. In this case the abscise axis represent f (value ranged from 0.1 to 0.5). The ordinates axis represents again the utilities addition after the failure of one or more agents. In this case the figure compares *util-opt* and *util-sub* for different number of agents (from 10 to 50). In this case, we can observe that: (a) if f is increased, *util-opt* and *util-sub* are decreased, falling to the half in the most cases. This result is logic since if n is lower, the number of tasks is also lower and so, the obtained profit will be smaller; (b) if we modify f keeping constant n , the difference between *util-opt* and *util-sub* grows more slowly –becoming nearly parallel- than in the above figure.

Figure 3 represents the difference between *util-opt* and *util-sub*. The graphic shows this parameter for different f values (value ranged from 0.1 to 0.5). In accordance with the above results this

difference is: i) always positive, ii) an increasing monotonic function of n , and iii) grows as a function of f .

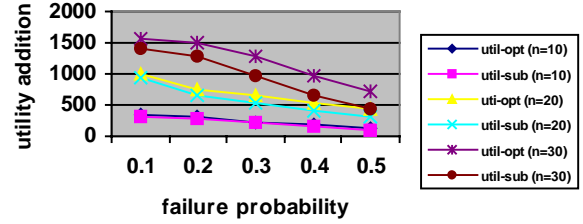


Figure 2. Utility addition of the system agents

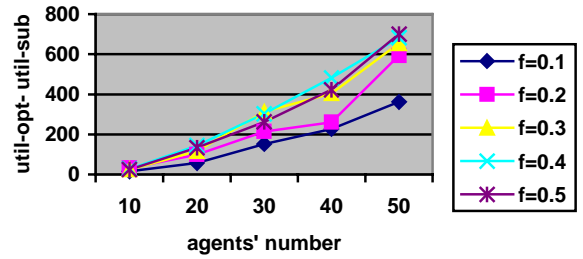


Figure 3. Difference between opt. and sub-opt. utility addition

In figure 4 we study the same parameters –*util-opt* – *util-sub*– but in this case f is modified and n is kept constant. In this figure we can observe that if f is increased also the difference is. However, this difference does not grow indefinitely. For example, with 10 agents the function maximum is found between the values 0.1 and 0.2 of the f . For 30 agents the function maximum is moved and is placed between 0.3 and 0.4. With 40 agents the maximum is moved to 0.5. This is because the c_i and k_i of the agents are limited to a concrete intervals. So, although n is increased, it would not be possible to obtain a greater benefit. Due to graphic shape we can expect that beyond 50 agents the difference will become steady independently of an increase of n .

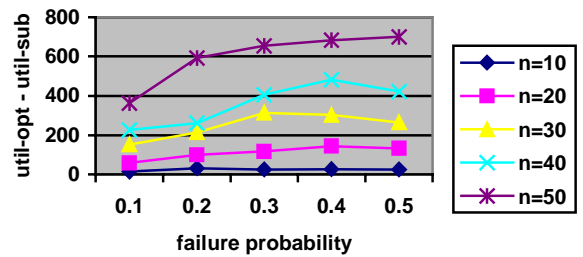


Figure 4. Difference between opt. and sub-opt. utility addition

The main conclusions could be summarized as follows: (a) As expected, the addition of agent utilities obtained by applying our coalition formation model after the failure of one or more agents (*util-opt*), is always better than the utilities addition obtained applying the alternative payoff division model (*util-sub*); (b) the difference between *util-opt* and *util-sub* grows as a function of n ,

depends on n more than on f , and becomes steady in spite of the increment of n , because of c_i and k_i are limited; (c) If n is small (≤ 20), it can be more efficient, from a computational point of view, to apply the alternative payoff division model after the agents' failure. On the other hand, if n is larger, the loss of utility after applying the alternative scheme is more important. In this case it may be convenient to apply the optimum model.

4. A STUDY ON INSINCERE AGENTS

An important goal into the development of negotiation mechanisms consists of designing protocols in which the effects of deception and manipulation can be constrained [2]. Because of the execution autonomy and self-interest of the agents, they can deceive or mislead each other if they believe that they will get more profits. "Stable" protocols are such that sincere behaviour will be self-enforced. This will be possible when the agents have no economic reasons to deviate from the negotiation protocol. Next, we will study what are the effects of deception and manipulation in our coalition formation model. We will suppose that the agents make manipulations or lie when they reveal their private information.

4.1 Experimental Design Description

Let us suppose that any agent lies to the remaining ones about its real value of c_i . The agent could think that a cost increment or decrement could increase its profit. With the goal of testing the effects of this manipulation, we have designed another simulation tool. The simulation was tested for a number of agents varying from 10 to 50, and with a failure probability ranged from 0.1 to 0.5. The input data to the problem are similar to the above study. In relation to the manipulation of c_i the tool allows to increase or decrease the cost of any agent. The manipulated cost will be generated in a random way as a percentage (10%, 20%,...) over (larger or smaller) the original cost. This quantity will be limited by the parameters MAXCOST and MINCOST. In addition, the simulation tool computes the lying bonus obtained by the insincere agent when it deceived during the revelation of c_i . The lying bonus, P_M , is stated in the following way: Let c_v be the original cost of the insincere agent, and t_v and x_v be respectively the quantity of task units and the utility assigned to the agent after the execution of the tasks allocation and payoff division algorithm with the original cost, c_v . In addition, let c_m be the manipulated or lying cost, and t_m and x_m be respectively the quantity of task units and the utility assigned to the agent after the execution of the model with the lying cost, c_m . P_M will be calculated as the difference between the total profit obtained by the agent revealing its original cost, true balance, B_v , and the total profit obtained by the agent revealing the manipulated cost, lying balance, B_m . If the insincere agent is a consumer or fringe consumer agent, B_v and B_m are calculated in the following way:

$$B_v = -t_v c_v + t_v c_v + x_v = x_v$$

$$B_m = -t_m c_v + t_m c_m + x_m = t_m (c_m - c_v) + x_m$$

In both cases, the benefit obtained by the agent is stated as the addition of its benefit (money) less the work that it has to carry out. B_v is calculated as the addition of the utility obtained by the agent, x_v , plus the quantity of money paid by the producer agent to the lying agent for its work, $t_v c_v$ (due to the lying agent is a consumer it carries out the task transferred by a producer agent), less the work that the lying agent has to carry out, $t_v c_v$. For B_m the agent's profit will be the addition of its utility, x_m , plus the

quantity of money paid by the producer agent, $t_m c_m$; and its work will be $t_m c_v$. So, P_M is:

$$P_M = B_m - B_v = t_m (c_m - c_v) + x_m - x_v$$

If the lying agent is a producer or fringe producer agent, B_v and B_m are stated as follows:

$$B_v = -t_v c_v + t_v c_v + x_v = x_v$$

$$B_m = t_m c_v - t_m c_m + x_m = t_m (c_v - c_m) + x_m$$

In both cases, the total benefit got by the agent is calculated as the addition of its profit (money) less the work that it has to carry out, or, in this case, the money that it must send to the consumer agent to execute its task, because of the lying agent is a producer agent. The reasoning is analogous to the above one, and P_M will be:

$$P_M = B_m - B_v = t_m (c_v - c_m) + x_m - x_v$$

4.2 Report of Results

We have performed several hundreds of simulations. Table 1 summarizes the results obtained when the lying agent manipulates, increasing, its c_i . The table shows the effect of this kind of manipulation on the different types of system agents (consumer, fringe consumer, producer, fringe producer). The possible results for this manipulation in any type of agent may be a positive, a negative or a null lying bonus. If in the table the corresponding square is empty, it means that for that agent type and with the actual manipulation –increase of the cost– a lying bonus of that type will never be obtained. Otherwise, if the square is not empty, it means that it is possible to obtain bonus of that type, and, in addition, inside the square the new possible status of the lying agent after the manipulation is showed. Table 2 is the analogue 1 for a lying agent that decreases c_i .

Table 1. The lying agent increases its c_i

LYING A. P_M	POSITIVE	NEGATIVE	NULL
Consumer	Fringe Cons.	Producer Fringe Producer Non Participant Fringe Cons.	Consumer
Fringe Consumer	Fringe Cons.	Producer Fringe Producer Non Participant Consumer	Consumer
Producer			Producer
Fringe Producer		Producer Fringe Producer	

Table 2. The lying agent decreases its c_i

LYING A. P_M	POSITIVE	NEGATIVE	NULL
Consumer			Consumer
Fringe Consumer		Consumer Fringe Cons.	
Producer	Fringe Prod.	Consumer Fringe Cons. Non Participant Fringe Producer	Producer
Fringe Producer	Fringe Prod.	Consumer Fringe Cons. Non Participant Producer	Producer

From the above results we can conclude that a positive lying bonus is only obtained in two situations: (i) When the lying agent is a consumer or fringe consumer agent and the manipulation is an increase of c_i ; (ii) When the lying agent is a producer or fringe producer agent and the manipulation is a decrease of c_i . Even in these situations a positive lying bonus is not obtained in all cases. Only when the consumer/producer agent or the fringe consumer/fringe producer agent is transformed after the manipulation in fringe consumer/fringe producer, the (lying) agent will obtain a positive bonus. We will discuss now each of the possible cases, i.e. each of the rows of tables 1 and 2.

Case 1 (Rw 1-T.2/Rw 3-T.1); Case 2 (Rw 2-T.2/Rw 4-T.1)

In this cases, and with the goal of making easy the discussion of the results, we will suppose into the general framework of the coalition formation model the following constrains: a) null transfer costs ($c_{ij}=0, \forall i,j \in N$), b) the agents are ordered by cost ($c_1 > c_2 > \dots > c_n$). The concluded justifications for this simplified model will be valid for the general framework. In this situation, and according to (1), we can establish the coalitional value $v(N)$, as (f is the index of the fringe agent):

$$v(N) = \sum_{i=1/f < f}^n t_i(c_i - c_f) + \sum_{i=1/f > f}^n h_i(c_f - c_i) \quad (3)$$

And x_i , following (2), will be given by:

$$x_i = t_i(c_i - c_f), \text{ if } a_i \in \text{producers} \quad (4)$$

$$x_i = h_i(c_f - c_i), \text{ if } a_i \in \text{consumers} \quad (5)$$

$$x_i = 0, \text{ if } a_i = a_f$$

Let us suppose (case 1) that the consumer agent (lying agent) declares a lying cost smaller than its original cost, $c_m < c_v$. The profit of the lying agent is calculated as in (5).

In order to calculate P_M we make the following reasoning: since $c_m < c_v$, the situation of the agent is kept similar after the manipulation (consumer agent), and its utility x_i will be larger because of the difference between c_f and c_m will be greater than between c_f and c_v . However, this utility increase will be compensated due to the lying agent suffers a loss of utility revealing a c_m lower than c_v . Since the producer agent, which transfers task to the consumer agent for its execution, pays c_m to the consumer agent instead of c_v , and $c_m < c_v$. So, in this case the lying bonus will be null. This intuitive reasoning may be easily proved by means of elementary manipulations: $P_M = t_m(c_m - c_v) + x_m - x_v$, where $x_v = h_v(c_f - c_v)$, and $x_m = h_m(c_f - c_m)$, so $P_M = t_m c_m - t_m c_v + h_m c_f - h_m c_m - h_v c_f + h_v c_v = 0$.

In the case of the lying agent is a producer or fringe agent the reasoning is similar and also for case 2.

Case 3 (Rw 3-T.2/Rw 1-T.1); Case 4 (Rw 4-T.2/Rw 2-T.1)

In this cases we will make an empiric justification of the results, because after the manipulation the lying agent may change its situation to many different ones (fringe consumer, consumer, fringe producer, producer or non participant). Since we are trying to prove the "stability" of the coalition formation model facing to a manipulation, our main goal would be to know the situations in which this manipulation will get profits.

Table 3. Obtained percentages for consumers/producers lying agents after an increase/decrease of their costs

LYING AGENT P_M	POSITIVE	NEGATIVE	NULL
Consumer	5.81%	62.7%	31.39%
Producer	2.43%	78%	17.07%

Table 4. Obtained percentages for fringe consumers/producers lying agents after an increase/decrease of their costs

LYING AGENT P_M	POSITIVE	NEGATIVE	NULL
Fringe Consumer	18.18%	81.81%	0%
Fringe Producer	35%	60%	5%

Tables 3-4 show the obtained rates from the simulations (these were carried out for a number of agents ranged from 5 to 20). In the rows we point out the situation of the lying agent before the manipulation (producer or consumer), and in the columns the possible results of the manipulation or lying bonus (positive, negative or null), after an increase of c_i when the lying agent is a consumer or a decrease when is a producer. The obtained ratio for each kind of agent and for each possible result of the lying bonus is showed in each table cell. The results in table 3 show that for producer and consumer agents, the rate of positive lying bonus is lower than the negative or null one (for consumer agents a 5.81% against a 94.09%, and for producers a 2.43% against a 95.07%).

For fringe agents (table 4), the percentage of positive lying bonus is also lower than the negative or null ones. However, if we compare with table 3, in 4 the ratio of positive lying bonus is larger than in 3. The reason is that in case 3 the lying agents, in order to obtain a positive P_M , have to modify their situations after an increase/decrease of their c_i (changing their situations to fringe agents, producer or consumers). But, in case 4 the lying agents do not have to modify their situations after an increase/decrease of their c_i with the goal of obtaining a positive P_M .

So, we can conclude that our coalition formation model is "stable" in the face of an agent's deception in the revelation of its c_i . With this type of manipulation, the rational behaviour for the agents is to be honest with respect to its c_i .

5. CONCLUSIONS

In this paper we have presented two empirical studies about the robustness of the coalition formation model in the face of the failure of one or more system agents, and about the effects of deception and manipulation on the model. The results have showed an acceptable performance of the model with different failure patterns, and the stability of the coalition formation model in the face of an agent's deception in the revelation of its unitary execution cost. Future works include an extension of the study about insincere agents, by considering other types of manipulations.

6. REFERENCES

- [1]Belmonte, M.V., Conejo, R., Pérez-de-la-Cruz, J.L. & Triguero, F. A Stable and Feasible Payoff Division for Coalition Formation in a Class of Task Oriented Domains. In Meyer, J.J. and Tambe, M.E. (eds.): LNAI 2333, Intelligent Agents VIII: Agents Theories, Architectures and Languages. 324-334, 2002.
- [2]Rosenschein, J.S., and Zlotkin, G. Rules of Encounter. MIT Press, 1994.
- [3]Kahan, J., and Rapaport, A. Theories of Coalition Formation. Lawrence Erlbaum Associates Publishers, 1984.