# A DECENTRALIZED COALITION FORMATION ALGORITHM AMONG HOMOGENEOUS AGENTS

[1] LEILA KHALOUZADEH, [2] DR.NASER NEMATBAKHSH, [3] DR.KAMRAN ZAMANIFAR

[1] Department of Computer Engineering, Islamic Azad University, Najafabad brunch, Iran

[2] Department of Computer Engineering, University of Isfahan, Isfahan, Iran

[3] Department of Computer Engineering, University of Isfahan, Isfahan, Iran

**ABSTRACT**

In a multi-agent environment, the accomplishment of a task may necessitate the co-operation of a number of agents. In order to allocate a task to a group of agents, methods of coalition formation in multi-agent systems can be used. In multi-agents systems, the formation of an optimal coalition is one of the main challenges to be overcome. With increase in the number of agents, the number of possible coalitions potentially capable of carrying out a specific task increases exponentially. Determining the value of each potential coalition in order to choose the best option involves an excess of time and memory usage. Some algorithms have solved the problem by distributing the calculation among the agents. Considering the fact that the agents are mobile the distance between the agents and the task is significant and that certain capabilities are required for any one task to be carried out we have presented a new algorithm for coalition formation. Here, the optimal coalition is determined without having had to ascertain the value of each potential coalition beforehand. Another advantage of this new algorithm is that there is no need for any connection between agents in the coalition formation stage. Moreover, the result of simulation and comparison show that this novel algorithm saves agents transfer time and cost.

**Keywords:** *Task allocation, coalition formation, multi-agent system, mobile agent*

## 1. INTRODUCTION

One of the issues which has been deliberated regarding multi-agent systems, is the coalition of a number of agents for the achievement of a single task. This issue arises when more than one agent is needed for one task, thus necessitating the temporary grouping of a number of agents i.e. coalition. Such a coalition is perpetuated only till the completion of the targeted task. As researchers that study on multi-agent systems domain, they invariably encounter the question: “which agents should form coalition and which coalition should execute which task?” This question must be answered, even for relatively simple multi-agent systems, and the importance of task allocation grows with the complexity, in size and capability, of the system under study. Even in the simplest case of homogeneous agents with fixed, identical roles, intelligent allocation of tasks is required for good system performance.

The number of coalitions possible for the completion of any one task is large and the aim is to choose from among the potential coalitions the one with optimal advantage. The advantage of different coalitions may be calculated based on their value-which may vary according to the initial premise. For example it may not be possible for two agents to group together because of dialogue limitations in the network and therefore only a decrease in value with such a coalition can result. Also, it is possible to consider the amount of profit the completion of a task has for the agents as the value-determing factor. Our premise was that the agents were in different position in the environment; therefore each agent has its own specific distance from the targeted task. This is a parameter not established in previous algorithms. In the present algorithm which we are presenting this is the parameter used to determine the value element. In practice, the decrease in distance results in an increase in speed and performance quality. In the formation of relief units, for instance this is a crucial matter. The shortest way to solve this problem is to consider all the possible lection to evaluate each me and finally choose the most advantageous. Since an increase in the number of agents leads to an exponential increase in the number of potential coalitions, taking such a course would not be economical neither in terms of time nor memory usage. The existing algorithms confront this challenge by seeking to lessen the number of possible coalitions and carrying out the calculations in a distributed manner. This is effected by the sending and receiving of a number of messages between the agents. In the present algorithm there is no need for message passing between the agents for coalition formation. Moreover the best coalition can be determined without any need for the calculation of coalition value. In our estimation, the optimal coalition is that of the group closest to the task, which can perform the task without any waste of potential. The results of simulation show that this algorithm is, in comparison with previous work, in addition to being scalable, more effective.

In the following section we focus on previously done work and existing algorithms. In section 3 the new coalition formation algorithm is presented. In section 4 the proposed algorithm is evaluated. The conclusion in section 5 closes the article.

## 2. PREVIOUS WORK

Much research has been done concerning coalition formation. It has aimed at solving problems in multi-agent, multi-task and multi-robot environments and various algorithms have been proposed. The differences between these algorithms relates to the initial premise, viewpoints, performance environment s, protocols, the kind of interactions and associations. The aim of presenting new algorithms and methods has been optimum gain in one or more parameters. The issue of quality and primacy in distributed systems has been endeavoured in various aspects of distributed artificial intelligence. One of the aspects concerns the allocation of distributed sources or tasks [2,3,8,10]. Another issue is that of allocating tasks to agents.

In 1995 Rothkopf [4] presented an algorithm in which dynamic programming was used to answer the problem of coalition formation. The method he carried out was based on sub-dividing the problem and solving each sub-division only once and saving the answer, thus avoiding excess work. The advantage of this algorithm is its execution time which is short, with an $O(3^n)$ order –the normal execution time would be $O(n^n)$. This is an algorithm suitable for execution with a small number of agents. Since the algorithm uses dynamic programming, it needs the use of memory to save the solution details which is a disadvantage.

Shehory and Kraus[9] presented a new “greedy” algorithm in 1998. This algorithm is based on an optimum local choice and the hope that this choice will lead to an optimal global solution.

Haque et al. [14] in 2009, proposed a model of the first-order alliance, where each agent builds candidate sets, based on “association coefficients”, that contain up to two other agents in the network and a hybrid automaton was produced for each agent based on the status of requests sent to candidates to form a coalition. Also Haque et al. [15] in 2010, model the multi-level alliance forming ability of male bottlenose dolphins to develop a decentralized multi-level coalition formation algorithm for a multi-agent system. The goal is to produce a model that is rich enough to capture the biological phenomenon of forming alliances, yet remain simple so that it can be implemented on engineered systems, such as network of unmanned vehicles.

Sandholm [6] presented an algorithm in 1999 in which the process of coalition formation is considered as a search site in a graph; in this graph each node represents a potential coalition structure. This researcher proved that in order to reach the optimal solution there need only be a survey of two planes of the graph in the primary search, and in this manner the process will lead to finding the optimum solution. Even though this algorithm draws a suitable balance between execution time and quality, but in order to reach the optimal solution it is necessary to study an exponentially increasing number of potential solutions.

Sen and Dutta[7] presented a genetic algorithm in 2000 for coalition formation. This algorithm starts with an initial set of candidate solutions called the population (e.g. a set of coalition), then gradually approaches any better solution. This process involves the three principle stages of evaluation, choice and re-organization. The algorithm does not guarantee an optimal solution.

Cheng and Dasgupta[16] in 2010 illustrate a method of calculating the weight of a robot that can be used in a Weighted Voting Games (WVG), based on its coverage history. Also, they extend the weighted voting game with robot domain knowledge to calculate a unique coalition called the Best Minimal Winning Coalition(BMWC). they give two acceptable methods to find the BMWC.

Experiment results show that the greedy method uses lower computational time to find the approximate BMWC solution, while the heuristic method can find the optimal solution.

Tosic and Agha[11] presented a complete algorithm for the coalition formation of autonomous agents in 2005. This algorithm has a graphic representation in which the connection between agents is shown by the connection of node; the grade of these connections is used to form a group of agents.

Rahwan[5] presented a new algorithm in 2008, the aim of which was to solve the problem of coalition structure shaping. In this algorithm a portion of the calculations is given to each agent; these calculations are independant of each other and their entirety embraces the problem. The algorithm is of a non-central type, it needs no interaction between agents and uses a minimum degree of memory.

Vig and Adams[12] studied coalition formation in 2007 consideration of its differences in relation to software and robotic agents. They then presented a coalition of robots in robotic environments and environments of practical appliance.

We now present our algorithm with the supposition that agents in the environment are mobile. Mentionable is the fact that the algorithm has been simulated and not tested in a real robotic environment.

## 3. THE NEW ALGORITHM

In multi agents systems there are a number of agents to carry out various tasks. These agents have specific capabilities. In our view the best coalition can be formed by those agents which are closest to the targeted task in terms of distance and can execute it without wasting un-needed capabilities. In any potential problem, that which is most significant for the formation of a coalition can be regarded as the foremost parameter, and the coalition and formation of all possible coalitions can be avoided. In the present algorithm, instead of considering all coalitions and subsequently imposing limitations on them, we chose those which are more suitable for the formation of a coalition from the beginning. The coalition resulting from this method will naturally be the optimum coalition for the parameter we had established. In the next part we first present the necessary definitions for problem-solving, then we present the algorithm with explanation of its stages.

### 3.1 PROBLEM PREMISES

Let us suppose the set A={A1,A2,…,An} has n agents; each agent has a specific number which we will call id. Each agent has the potential of three different capabilities. If the agent possesses a capability we assign 1 to its field, otherwise zero. The array RAi shows the capability array of agent i. let us also suppose that T={T1,T2,…,Tm} is the set of m tasks. Each task may be independent or in association with other tasks; we show this relation with a precedence graph, a model of which has been used in [12]. In the precedence graph of figure1 the task T3 is dependant on the two tasks T1 and T2 and must start after them. But tasks T1, T2 and T4 are independent. Tfree in the algorithm shows whether a task is independent or otherwise. If the task is dependant the field is set with the numbers of the tasks on which it is dependant and if it is independent with another number (for instance 99- if the number of no id is this number). After the completion of the task this field is assigned the quantity zero.

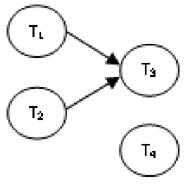

*Figure1. Task precedence graph*

Each task necessitates certain capabilities for its accomplishment which is shown by RT. RTj shows the necessary capabilities for the completion of task Tj. One of the conditions for the formation of a coalition of K size is that the entirety of the coalition capabilities be equal or more than that which the task necessitates.

The position of the agent and the task in the environment is shown by(x,y), we will suppose that the position of the agents in the environment is known by themselves. Each agent can, on the basis of these co-ordinates calculate its own and other agents distance from the task. We have shown this distance with “dis”. The other condition which renders the coalition optimal is having the sum of the coalition agents’ distance from the task less than it is in all other coalitions. Any coalition will remain in formation only till the completion of its targeted task, after which the agents can participate in the formation of another coalition. Every agent has a list of the tasks to be carried out in the environment; it has also received from the other agents their number and their capabilities. This process, however, is carried out only once, afterwards the coalitions are formed based on the required tasks. The coalition agents' id is put on the LC list. Figure2 shows the proposed algorithm.

## 3.2 EXPLICATION OF THE ALGORITHM

All agents execute the below algorithm locally and simultaneously. As a result each agent can carry out its own calculations completely and independently without having had to receive information from other agents. This increases algorithm execution speed. We now explicate the algorithm stages.

First stage: each agent sends a list to other agents in which its id, capabilities array and environment coordinates. Each agent receives the list sent from other agents and creats a table of contents including id, distances from tasks, x and y coordinates and task execution times. Initially the task distance and execution time are quantified as zero.

Second stage: if the task execution list is not empty, each agent will choose a task starting from the top of the list, the task chosen is one that is independent and/or the tasks dependant on it have been accomplished. The task execution lists of all agents are similar; therefore all agents will definitely choose the same task. Then on basis of the previously drawn table, the distance of each agent from the task is calculated and the distance column is quantified. Subsequently the table is stored in order of distance increase.

Third stage: each agent, starting from the top line of the table, adds the capabilities of the free agents to the extent that the sum of capabilities equals or surpasses the task requirement. The id of those agents is added to the coalition list. In case the sum of capabilities should not meet task requirement, the task remains unaccomplished and is put at the end of the task list, and the process reverts to stage two in order to initiate the next task. There may be, in the formed coalition, agents without which the task can nevertheless be accomplished, these are omitted in order to avoid the waste of agent capability. The process of omission is carried out such that the agents further away from the task are omitted before those which are closer.

Fourth stage: the agents which form a coalition for task accomplishment adjust their coordinates to those of the task. This means what actually happens is that the agents move to the task location. Then for the targeted task Tfree is set at zero which means the task has been accomplished. We then go to stage two for the selection of the next task.

This algorithm continues as long as there remain unaccomplished task. If an agent participates in the forming of a coalition, after the achievement of the coalition task it is freed and able to participate in another coalition. From the time of coalition formation until the time of task accomplishment, all agents in the coalition are in the "involved" status. This length of time is the task time which is adjusted in the fourth stage for the coalition agents.

```
Stage1:
DOALL   i=1..n // each agent i executes locally and parallel with all other agent
Send [id(i),BA(i),x(i),y(i)]  to all other agents
For all j:$A_j \in$ A  do
    Recive [id(i),BA(i),x(i),y(i)]  from $A_j$
    Add recived message to agenttable
End for
END DOALL
Stage 2:
DOALL  i=1..n  // stage 2-4 are repeated until  T =null
If (T =empty)
  Goto END OF PROGRAM
Else
set   LC$\leftarrow$ {}  // NULL
set  task $\leftarrow T_{next}$ : $T_{next} \in$ T   //next task from list T
For  all j:$A_j$ € A  do
     compute dis($A_j$) from task
     update tableagent
End for all
sort tableagent  with respect to dis
end else
stage 3:
sum(BA)$\leftarrow$0
row$\leftarrow$1
While (sum (BA)<$BT_j$  and  row<n) do
  If ( time($A_{row}$)=0)
      sum(BA)=sum(BA)+$BA_{row}$
      add  $id_{row}$  to LC
End if
row$\leftarrow$row+1
End while
If(row>n)
```

```
  Move task to end of list T
  Goto stage 2
End if
k←row
while (k>=1) do
    sum(BA)=sum(BA)-$BA_k$
   If (sum(BA)>=$BT_j$ )
      delete ($id_k$)  from  LC
   Else
         sum(BA)=sum(BA)+$BA_k$
  k←k-1
End while
stage 4:
For  all j:$A_j \in$ A  do
    set time($A_j$)←time (task)
     set x($A_j$)←x (task)
    set y($A_j$)←y (task)
End forall
remove task from T
Goto stage 2
END DOALL
END OF PROGRAM
```

*Figure 2. proposed algorithm for coalition formation in multi-agent systems*

## 4. ALGHORITHM EVALUATION

The proposed algorithm aims at solving the problem of task allocation by coalition formation in multi-agent systems. Calculating the value of all possible coalitions and selecting the best needs much time, memory use and interaction between agents. In the algorithm here presented, by adding a distance parameter, the need for a calculation of all coalition values is dismissed. This algorithm introduces only one coalition for task accomplishment. This one coalition has all the capability required for the targeted task and prevents the waste of capability in other agents. In addition, the agents involved in the formation of the coalition are closest in terms of distance to the targeted task. With the movement of the coalition agents, all agents adjust to the new location without receiving any message. Each agent sends its features to all other agents only once; this can occur when an agent enters the system. In other words the agent is “born”. With the decrease in communication there follows a decrease in agent interaction error.

We used JADE [1] for simulation. All created agents are the same and there exists no coordinating agent. In the code written for 25 agents, each agent sends 10 kilobytes which means each agent sends 400 bytes of information. Moreover since the dispatch and reception of information occurs only once and in one stage only, increase in the number of agents does not have negative effect on the function of the algorithm. Where each agent knows the other agents and their features, there is no need for messaging between agents; this method has been used in many algorithms.

Rahwan in [5] and Shehory & Kraus in [8] calculated the value of possible coalitions in distribution in their algorithms. It need be mentioned that their intention was to do exactly this.

Although we do not calculate the value of all possible coalitions, we select the optimum coalition based on the values we calculate; therefore we can compare their algorithm in terms of the number of bytes transferred between agents and the amount of memory each agent needs to store the coalitions with our algorithm. Table 1 shows the number of bytes transferred between agents in three algorithms. The amounts assigned to the Rahwan(dcvc) and Shehory & Kraus (SK)algorithms are based on [5].

In the written code, the local memory of each agent adequate for the storing of all the information needed by that agent is less than one kilobyte. The number of bytes needed for coalition storage is, at most, equal with the number of the agents since in this algorithm any agent at any one time holds only one coalition. Table 2 shows the amount of memory needed for the storage of necessary coalitions in the three algorithms. In this table the amount assigned to the Rahwan(dcvc) and Shehory & Kraus (SK)algorithms accord to [5]. (What is meant by Pi is the set of possible coalitions which are given initially for calculation to the ith agent).

The execution order of the algorithm for the allocation of any task, is O($n^2$) for each agent, since the highest number of commands carried out in this algorithm is in relation to table sortment. With the addition of an agent only one line is added to the table, therefore the algorithm can continue working efficiently because in terms of memory usage only a number of byte are needed to store agent characteristic, whereas if, with the addition of one agent, we should want to consider all the additional coalitions made possible by the addition, we would have to multiply the coalition by two. All the above features point to the efficiency of the proposed algorithm and the fact that it can be scaled.

*Table 1. The total number of bytes that had to be sent between the agents*

| Number of agent | DCVD | SK (99% confidence) | OUR ALGORITHM (with first message) | OUR ALGORITHM (without first message) |
|---|---|---|---|---|
| 10 | 0 | 8,799 ± 1 % | 1600 | 0 |
| 11 | 0 | 20,447 ± 0.8 % | 1936 | 0 |
| 12 | 0 | 45,076 ± 1.2 % | 2304 | 0 |
| 13 | 0 | 99,538 ± 0.3 % | 2704 | 0 |
| 14 | 0 | 217,080 ± 0.5 % | 3136 | 0 |
| 15 | 0 | 469,173 ± 0.7 % | 3600 | 0 |
| 16 | 0 | 101,1217 ± 0.7 % | 4096 | 0 |
| 17 | 0 | 3,242,544 ± 1.5 % | 4624 | 0 |
| 18 | 0 | 6,888,787 ± 1.6 % | 5184 | 0 |
| 19 | 0 | 14,644,832 ± 2 % | 5776 | 0 |
| 20 | 0 | 30,913,264 ± 1.1 % | 4600 | 0 |
| 21 | 0 | 65,114,817 ± 0.3 % | 7056 | 0 |
| 22 | 0 | 136,877,925 ± 0.2 % | 7744 | 0 |
| 23 | 0 | 286,712,976 ± 0.3 % | 8464 | 0 |
| 24 | 0 | 573,494,824 ± 0.6 % | 9216 | 0 |
| 25 | 0 | 1,146,989,648 ± 0.2 % | 10000 | 0 |

*Table 2. The minimum number of bytes required per agent to save the necessary coalitions.*

| Number of agent | DCVD | DCVC (maintain $p_i$) | SK | OUR ALGORITHM |
|---|---|---|---|---|
| 10 | 2 | 206 | 1024 | 10 |
| 11 | 2 | 374 | 2048 | 11 |
| 12 | 2 | 684 | 4096 | 12 |
| 13 | 2 | 1262 | 8192 | 13 |
| 14 | 2 | 2342 | 16384 | 14 |
| 15 | 2 | 4370 | 32768 | 15 |
| 16 | 2 | 8192 | 65536 | 16 |
| 17 | 3 | 23133 | 196608 | 17 |
| 18 | 3 | 43692 | 393216 | 18 |
| 19 | 3 | 82785 | 786432 | 19 |
| 20 | 3 | 157287 | 1572864 | 20 |
| 21 | 3 | 299595 | 3145728 | 21 |
| 22 | 3 | 571953 | 6291456 | 22 |
| 23 | 3 | 1094169 | 12582912 | 23 |
| 24 | 3 | 2097153 | 25165824 | 24 |
| 25 | 4 | 5368712 | 67108864 | 25 |

Now, we will discuss an applied instance which was proposed and implemented on robots by Lovekesh Vig [13]. Then we simulate this case with our proposed algorithm and present the results.

The above mentioned researcher proposed a coalition formation algorithm in a multi-robot environment in order to solve the problem. In this instance there are four robots in the environment, all with similar capabilities. Figure 3(a) shows three boxes which the four robots are supposed to move to the position shown in figure 3(b). Each box needs two robots for its transfer. In the experiment carried out by Lovekesh Vig[13], robots 1 and 2, and robot 3 and 4 push boxes T1 and T2 respectively toward each other simultaneously. Then robots 1 and 2 change position in order to push box 3. The results have been shown in figure 4.

When simulated with the proposed algorithm, section 4(c) changes to that shown in figure 5. As can be seen in figure 5, those robots form a coalition in order to push box 3 which are closer to it; this gains in terms of time and agent transfer cost. On a large scale basis the gain would be significant; for example in the formation of rescue squads where time is an important factor.

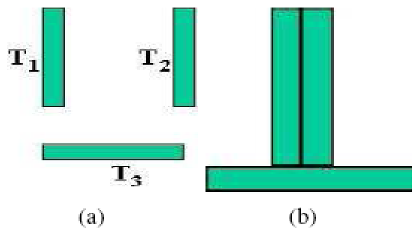

*Figure 3. section (a) shows the original and section (b) shows the targeted position [13].*

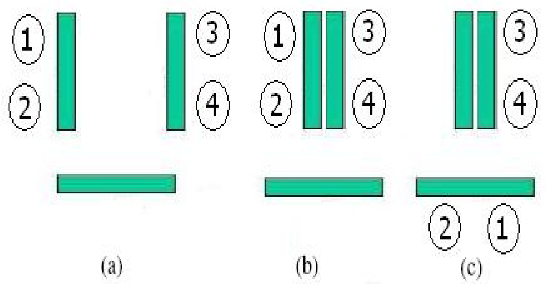

*Figure 4. section (a), the position of robots in two groups; section (b), robots have pushed boxes toward each other; section (c) robots 1 and 2 chosen to push next box.*

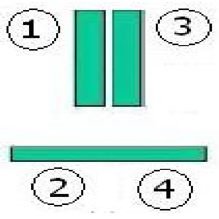

*Figure 5. last robot position using proposed algorithm.*

## 5. CONCLUSION AND FUTURE WORK

In this article we presented a new algorithm for problem solving in relation to task allocation using coalition formation in multi-agent systems. The target in this regard was to find coalition of mobile agents with optimum benefit -forming a coalition capable of executing the targeted task with least possible distance from it- in the execution of a number of tasks. Through the comparison and evaluation of the algorithm it became clear that there is gain in its implementation in terms of time and agent transfer cost. Moreover, the number of messages which need be sent and received decreases which in turn increases data safety, since during message transmission the message text may be hacked or messages carrying important information may be lost.

a major limitation on this algorithm (and on algorithms that have limitation on message passing in general) is that fault tolerance is low. Also this algorithm only can use for homogeneous agents. We envisage that this algorithm can support heterogeneous agents with different memory and communication limitations.

## REFRENCES:

[1] Bellifemine, Fabio and Caire, Giovanni and Greenwood, Dominic, “Developing Multi-Agent Systems with JADE”, *Telecom Italia S.p.A., John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester,West Sussex PO19 8SQ*, England, 2007.

[2] Modi.P.J, Jung.H ,Shen.W, Tamb.M,Kalkarni.S. , “A dynamic distributed constraint satisfaction approach to resource allocation ”. *In proceeding of the 7th international conference on principles and practice of constraint programming*, 2001.

[3] Modi.P.J, Jung.H ,Shen.W. “Distributed resource allocation : formalization, complexity result and mapping to distributed CSPs ”. technical report, November 2002.

[4] Rothkopf, M. H., Pekec, A., and Harstad, R. M, “Computationally manageable combinatorial uctions”, *Management Science*, 1995, 44(8):1131–1147.

[5] Rahwan,Talal, “Algorithms for Coalition Formation in Multi-Agent Systems”, *A thesis submitted in partial fulfillment for the degree of Doctor of Philosophy*, 2007.

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

[7] Sen, S. and Dutta, P, “Searching for optimal coalition structures”, *In Proceedings of the Fourth International Conference on Multiagent Systems*,2000, pages 286–292.

[8] Shehory, O. and Kraus, S., “Coalition formation among autonomous agents: strategies and complexity”,*in from reaction to cognition , spring's lecture notes in artificial intelligence*, 1995, vol.957,pp.57-72.

[9] Shehory, O. and Kraus, S. , “Methods for task allocation via agent coalition formation” Artificial Intelligence, 1998 , 101(1–2):pp.165–200.

[10] Shehory, O. and Kraus, S. , “Task allocation via coalition formation among autonomous agents”, *in proceedings of the 14th international joint conference on artificial intelligence* , Montreal,Quebec , august 1995.

[11] Tosic,Predrag , Agha, Gul, “ Maximal Clique Based Distributed Coalition Formation for Task allocation in Large-Scale Multi-agent

Systems" , Springer-Verlag Berlin Heidelberg 2005, pp. 104–120.

[12] Vig, L., Adams, J.A., "Coalition Formation: From Software Agents to Robots" , *author's proof ,Springer Science , Business Media B.V*, 2007.

[13] Vig, L., " Multi-Robot Coalition Formation" , dissertation submitted to the faculty of the graduate school of vanderbilt university in partial fulfillment of the requirements for the degree of doctor of philosophy in computer science , December, 2006.

[14] M. A. Haque and M. Egerstedt. "Coalition formation in multi-agent systems based on bottlenose dolphin alliances". *In Proceedings of the American Control Conference, St. Louis, MO, USA*, Jun. 2009.

[15] M. A. Haque, A.Rahmani and M.Egerstedt, "Biologically Inspired Coalition Formation of Multi-Agent Systems", *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010) ,Toronto, Canada*, may 2010, pp.1427-1428

[16] K. Cheng, P. Dasgupta, "Multi-Agent Coalition Formation for Distributed Area Coverage: Analysis and Evaluation", *In Proceedings of the IEEE International Conference on Web Intelligence and Intelligent Agent Technology*, 2010, pp.334-337