Nature Inspired Self organization for Adhoc Grids

Tariq Abdullah[‡], Ashiq Anjum, Nik Bessis, Stelios Sotiriadis
School of Computing and Mathematics,
University of Derby, UK
{m.t.abdullah, a.anjum, n.bessis, s.sotiriadis}@derby.ac.uk

Koen Bertels
[‡]CE Laboratory, EWI Faculty,
TUDelft, The Netherlands
{m.t.abdullah, k.l.m.bertels}@tudelft.nl

Abstract

Ant Colony Optimization (ACO) and other similar nature inspired mechanisms like artificial neural networks, swarm intelligence and evolutionary algorithms are based on naturally existing Complex Adaptive Systems (CAS). Human immune system, sand dune ripples, and ant foraging are some examples of the naturally existing CAS. Participating agents in these systems interact according to simple local rules which result in complex behavior and self-organization at system level. Adhoc grids are dynamic in nature and participating nodes show intermittent and volatile participation. Resource availability fluctuates over time in adhoc grids and results in a new adhoc grid state. These changes require adoption of the adhoc grid to a new state by applying some self organizing mechanism.

In this paper, we present nature-inspired (ACO), micro-economic based mechanisms for infrastructure level self-organization in adhoc grids. These mechanisms help in achieving a scalable, dynamic and a self-organizing adhoc grid infrastructure. These mechanisms are evaluated with varying workloads in different network conditions. Study of these mechanisms helped in understanding the effect of ACO based self-organization mechanism on the infrastructural spectrum, ranging from completely centralized to fully decentralized.

1. Introduction

Adhoc grid is sometimes referred to as desktop grid computing [1], public resource computing [2], or volunteer computing. Adhoc grid has geographically distributed, heterogeneous, and ephemeral nodes. The participating nodes may have different ownership with varying usage policies. Resource management in adhoc grid nodes with above characteristics becomes a challenging undertaking. Different underlying network infrastructures and the variable participation of nodes further increase the administrative complexity of the adhoc grid. Resource availability fluctuates over time and makes resource management even more complex. These changes require adaption of the adhoc grid to a new system state by applying some self organizing mechanism.

In this paper, we present and evaluate the nature inspired, micro-economic based infrastructure level self-organization mechanisms in adhoc grids under different network and varying work load conditions. This paper is an extension of the work presented in [3]. We define infrastructure-level self-organization (Figure 1) in adhoc as a mechanism that emerges from self-organization at two levels; at the individual node level and the system level. An individual adhoc grid node can be a consumer or a producer of resources. Node level self-organization enables an individual node to maximize its utility from the adhoc grid in terms of task execution or resource consumption. System level self-organization involves understanding under what conditions/circumstances a fully centralized or a fully decentralized infrastructure (or anything in between) is most appropriate. The presented mechanisms enable the adhoc grid to adapt itself in order to provide best resource allocation under different network and varying work load conditions.

Our proposed self-organizing mechanism is motivated from naturally existing CAS that are characterized by decentralized control, emergent behaviour, robustness and self organization. The participating agents in these systems interact according to simple local rules which result in self-organization and complex behaviour. We are specially interested in applying ACO [4] inspired, micro-economic based self-organizing mechanism in adhoc grids. In the proposed self-organizing mechanism, an adaptive, complex, and self-organizing behaviour at adhoc grid level emerges from local, decentralized interactions of the participating adhoc grid nodes. We are also interested in understanding the effect of ACO based mechanism on the

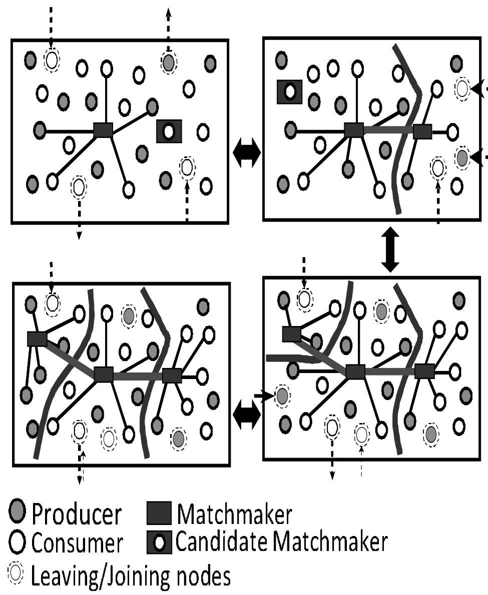


Figure 1: System level adaptation.

infrastructural spectrum that ranges from completely centralized to fully decentralized extremes.

The rest of the paper is organized as follows. Section 2 describes related work, whereas, sections 3 and 4 describe the proposed architecture by describing the modified ACO mechanisms and algorithms for self-organizing adhoc grid segments. Experimental setup and the discussion of experimental results are described in Sections 5 and 6 respectively. Section 7 concludes the paper.

2. Related Work

A work closely related to the work presented in this paper is described in [5]. This work focused on self-organizing broker topologies in publish/subscribe systems and used bloom filters as self-organizing mechanism. It only used similarity of the notification messages in publish/subscribe system as a parameter for self-organization. Similarity of the notification message is also used for reducing the cost incurred for forwarding and processing the received notification messages from the participants in a publish/subscribe system. Our work differs from this approach by presenting algorithms for segmentation and de-segmentation of the adhoc grid and by applying the micro-economic approach for node level self-organization. A hybrid ant colony optimization algorithm for selecting an appropriate scheduler in heterogeneous computing environments is proposed in [6]. This approach is not evaluated for dynamic computing environments and is only tested in a static environment for solving a scheduling problem for independent jobs. Hence, we are unable to access the approach for dynamic and complex computing environments like adhoc grids. Messor [7] implemented a load balancing approach for P2P grid computing systems on top of the Anthill framework. An ant wanders randomly in the environment and searches for an overloaded and underloaded node by using Search-Max and Search-Min states respectively. The ant balances the load of overloaded and underloaded node and starts again for next pair of overloaded and underloaded nodes. Since P2P grids and similar distributed computing environments are dynamic in nature, the overload and underload conditions of nodes may change before the ant performs load balancing among the overloaded and underloaded nodes and may cause erroneous load balancing decision making. Comparison of different algorithms for self-organization and adaptive service placement in dynamic distributed environments is done in [8] and search for best task scheduling for grid computing using ant colony optimization based algorithm is done in [9]. A simple ant-like self-organization mechanism to achieve overall load balancing in grid is described in [10]. The mechanism distributed jobs evenly among the available resources by a collection of simple local interactions. The number of ants and the number of their (ants) steps are defined by the user and don't change during the load balancing process. This restriction is impractical for grid like environments. *Ant level load balancing* [11] is proposed to overcome the limitations of [10]. This works well for a fixed number of grid nodes but can't handle varying number of nodes in a computational grid.

Main contributions of this paper are that this paper presents and evaluates a nature inspired, microeconmic based infrastructure level self-organization mechanism in adhoc grids under different network and varying work load conditions. Secondly, it also presents and evaluates a history-based dynamic pricing mechanism for the presented self-organization mechanism.

3. Micro-economic based ACO Approach

Algorithms, formulas and mapping of an ACO [4] system to an adhoc grid were explained in our earlier work [3], [12]. An overview of our earlier work is presented in this section for better understanding of the adhoc grid adaptation mechanisms and the results presented in this paper.

Basic concepts in an ACO approach are *ants, food sources* and *pheromone value*. Adhoc grid nodes represent *ants*, matchmaker(s) are *food sources* and *pheromone value* indicates weight of a matchmaker in the ant system. A matchmaker with higher pheromone value indicates that it has a higher probability of finding a compatible resource offer for a submitted resource request and vice verse.

Participating nodes can request the required resources from adhoc grid or can offer their excess resources to the adhoc grid. Resource allocator (hereafter referred to as matchmaker) uses Continuous Double Auction (CDA) as its resource allocation mechanism [13]. Matchmaker performs the resource allocation process by matching available resource-requests and resource-offers. In the market based CDA approach, individual consumer/producer nodes use price of a computing resource as an indicator of the resource availability/scarcity. Consumer/producer nodes self-organize themselves by increasing/decreasing their resource price according to their utility.

A matchmaker promotes a normal node as a new matchmaker in overload status and demotes itself when underloaded. An overloaded matchmaker shares its excess workload with a new matchmaker by promoting a normal node to a matchmaker. This promotion and demotion of the matchmakers results in segmentation and de-segmentation of the adhoc grid. In this way, the adhoc grid is segmented when new matchmakers are introduced, and the adhoc grid segments are merged back when the matchmakers are demoted. This approach enables the adhoc grid in achieving the system level self-organization in the presence of consumer/producer nodes of same/different resource types. Section 4 explains the proposed algorithms for infrastructure level self-organization in adhoc grids.

3.1. History-based Dynamic Pricing Strategy

In an adhoc grid, a manual price setting mechanism for each requested/offered resource quantity is not feasible and is difficult to implement. Furthermore, a dynamic pricing strategy in an adhoc grid allows adapting to the supply/demand changes in the adhoc grid and helps consumer/producer agents to maximize their utility i.e., producer can maximize its earnings and consumer can acquire required resources with as little spending as possible.

Each consumer/producer agent uses a history based dynamic pricing strategy [14] to calculate its ask/bid price. This history-based dynamic pricing strategy is based on the interactions of an agent with its environments. A consumer/producer agent learns about its past resource/offer utilization from its interactions with the environment and uses this information for its future pricing decisions. This process is graphically depicted in Figure 2 (adapted from [15]), where a consumer/producer agent receives feedback from its environment about its previous resource/offer utilization at time t as $r(t)$ and updates its price as $price(t+1)$ at time $t+1$ in form of its action/input to the environment**.**

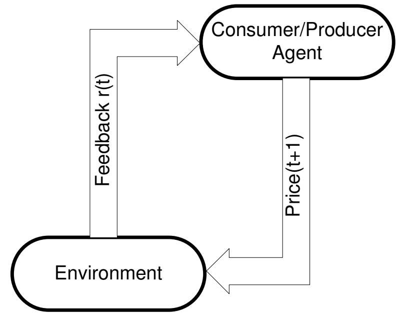


Figure 2: Consumer/producer agent interaction with the environment.

In this approach, each consumer/producer node joins the adhoc grid with an initial price and dynamically updates it by using a history-based dynamic pricing strategy. The calculated price is thus a reflection of the value of each resource unit which the consumer/producer nodes are willing to buy or sell. An overview of the pricing function is given here.

Nodes perceive the demand and supply of resources through their previous experiences and update their prices accordingly. Based on this strategy, ask/bid price at time interval t is defined as:

$$P(t) = P(t-1) + \triangle P \tag{1}$$

where $P(t)$ is the new price and $P(t-1)$ denotes the previous price. $\triangle P$ for the seller and buyer is defined based on the previous history of resource/task utilization. For seller, $\triangle P$ is defined as:

$$\triangle P = \alpha(\mu(t) - \mu_{thR})p(t-1) \tag{2}$$

and for the buyer $\triangle P$ is defined as:

$$\triangle P = \beta(\mu_{thT} - \mu(t))p(t-1) \tag{3}$$

where α and β are the coefficients to control the price change rate. In this paper, $\alpha = \beta = 0.8$ is used. μ_{thT} and μ_{thR} represent the task and resource utilization thresholds. μ_t is the task/resource utilization of an individual node. μ_t is defined as:

$$\mu_t = \sum_{i=t_0}^{t} x(i) / \sum_{i=t_0}^{t} N(i) \tag{4}$$

where $x(i)$ is a sold/purchased resource during time period $[t_o, t]$ and $\sum_{i=t_0}^{t} N(i)$ is the total number of the offered/requested resources in the time period $[t_o, t]$.

3.2. Pheromone calculation

This section describes formulas for calculating the pheromone value of the matchmakers in the presented approach. Pheromone value indicates weight of a matchmaker and is calculated periodically according to the following formula.

$$\tau_{new} = \begin{cases} \gamma * \tau_{old} + (1-\gamma) * \triangle\tau & if\ \triangle\tau > 0 \\ (1-\gamma) * \tau_{old} + \gamma * \triangle\tau & if\ \triangle\tau < 0 \end{cases} \quad (5)$$

The parameter A represents the pheromone evaporation rate. The value of γ varies between 0 and 1. τ_{old} represents the pheromone value during time interval $T1 = [t_{s_1}, t_{e_1}]$. Whereas, $\triangle\tau$ is the change in the pheromone value between the time interval $T1 = [t_{s_1}, t_{e_1}]$ & $T2 = [t_{s_2}, t_{e_2}]$. The start time of both intervals is represented by t_{s_1} & t_{s_2} and t_{e_1} & t_{e_2} represent the end time of both the intervals, such that $T2 > T1$ & $t_{s_2} = t_{e_1}$. Details for calculating $\triangle\tau$ and explanation of the formula is described in [3].

In the experimental results presented in Section 6, pheromone value of each ant is initialized by 1. We are well aware of the fact that ants wander randomly in nature, without any initial pheromone value as used in our experiments. Pheromone value will always be zero, if the initial pheromone value is set to 0 in our experiments. We initialize pheromone value for accommodating this factor. The value of γ (rate of pheromone evaporation) is set to 0.8 in these experiments.

4. ACO based Self-organizing Adhoc Grid Segments

Algorithms for dynamic segmentation and de-segmentation of the adhoc grid were presented in [3]. An overview of these algorithms is presented in this section for better understanding the results presented in this paper.

The segmentation and de-segmentation of adhoc grid is a dynamic process and is triggered from changing resource availability in adhoc grid. New segments are either created for a specialized pool of resources or when existing matchmaker(s) (aka foodsource) is overloaded. The pool of resources may consist of specialized hardware for a video rendering application, or specialized software for data processing, or a pool of resources for remote collaboration on a scientific experiment. These segments are removed/merged back when a specialized pool of resources is not needed. Following algorithms are used for dynamic segmentation and de-segmentation of the adhoc grid.

4.1. Change Segment

An adhoc grid node may change its resource category at any time during the experiment according to its resource requirements. The node sends its new task-requests to its existing responsible matchmaker. The matchmaker discovers the right food source for this resource-request by following the steps listed in Algorithm 2. The node receives matched message reply from the new matchmaker, changes it responsible matchmaker, and sends its new resource-request/resource-offers to the new matchmaker. In this way the node leaves its previous virtual segment and becomes part of the adhoc grid segment according to its changed resource category. These steps are listed in Algorithm 1.

Responsible matchmaker of a node may also be change in overload condition of the matchmaker. In that case a matchmaker promotes a node to a matchmaker and shares its excess work load with the new matchmaker. These segments are merged back when a matchmaker(s) is underloaded. Algorithms and experiments for promoting a node to matchmaker and demoting a matchmaker node back to normal node are explained in out previous work [16].

4.2. Discover right food-source

A matchmaker compares food/resource category of a received resource-request/resource-offer with its food category. It also checks that it has the highest pheromone value of its resource category as compared to its predecessor/successor matchmakers from its local knowledge. The matchmaker follows Algorithm3 for finding a matching resource-request for a resource-offer and vice verse.

Algorithm 1 Change segment.

1:IF $(CPNode_{FS-ID} \; != matchedMessageSender_{FS-ID})$ THEN
2: $CPNode_{FS-ID} = matchedMessageSender_{FS-ID}$
3: Join new virtual segment
4: Send request/offer ant to $CPNode_{FS-ID}$
5:ELSE
6: No change in virtual segment
7: Send request/offer to old $CPNode_{FS-ID}$
8:END IF

Algorithm 2 Discovering the right food source.

```
IF (($Ant_{resCategory}$ equals $FS_{resCategory}$) AND ($FS_{phValue}$ is highest)) THEN
    CALL Find matching request/offer algorithm
    IF (no match found) THEN
        IF ($Ant_{resCategory}$ equals $pFS/sFS_{resCategory}$ ) THEN
            $Ant$ visit $pFS/sFS$
    GO TO Step 1
        END IF
    END IF
    ELSE IF ($Ant_{resCategory}$ not equals $FS_{resCategory}$) THEN
    $Ant$ visit $successorFS$
    GO TO Step 1
    END IF
END IF
```

When food/resource category of the received resource-request/resource-offer does not match with the food/resource category of the matchmaker or the matchmaker does not have the highest pheromone value then the received resource-request/resource-offer is forwarded to the successor matchmaker of the current matchmaker. The successor matchmaker performs these steps from the beginning. This process is listed in Algorithm 2.

4.3. Find Match

A matchmaker tries to find a matching resource-offer for a received resource-request or vice verse after determining that it is the right matchmaker/food-source. It first stores the received resource-request/resource-offer in its request/offer queues and then compares the resource-request constraints with resource-offer parameters. A resource-request and resource-offer pair is declared matched when a resource-offer satisfies all constraints of the resource-request. The matching consumer/producer nodes are notified about the match [13].

Pheromone value of a matchmaker increases after finding a successful match and decreases when it is unable to find a match. Formulas for calculating pheromone value are explained in Section 3.2. A matchmaker periodically notifies its immediate neighboring successor and predecessor matchmaker/food-source nodes about its pheromone value.

Algorithm 3 Find matching request/offer.

```
Store request/offer in request/offer repositories
Match request parameters with offer parameters
IF ($Offer_{parameters}$ match $Request_{parameters}$) THEN
    Send matched message to consumer
    Send matched message to producer
    Update pheromone value
END IF
```

5. Experimental Setup

Number of participating adhoc grid nodes varies in the presented experimental results from $15-650$. Proposed nature-inspired, micro-economic based approach is implemented on top of Pastry [17], a structured overlay network. Pastry takes care of overlay network management, node join and node leaving. Scalability and robustness of the presented approach is tested by executing the experiments on PlanetLab [18].

Different network conditions are evaluated in the results presented in this paper. These network conditions are summarized in Table 1. Ratio of resource-offers to resource-requests represents the approximate percentage of resource-requests and resource-offers generated by the participating nodes in an experiment. For example, $20\%-80\%$ ratio in Resource Intensive Network (RIN) condition means 20% resource-requests and 80% resource-offers are generated by the participating nodes during an experiment. A similar explanation applies to Task Intensive Network (TIN) and Balanced Network (BN) network conditions.

Number of resource categories is 3 and number of matchmakers is 5 in the presented results. Time to Live (TTL)/validity period of resource-requests and resource-offers is 10000 milliseconds for accommodating delays observed in PlanetLab.

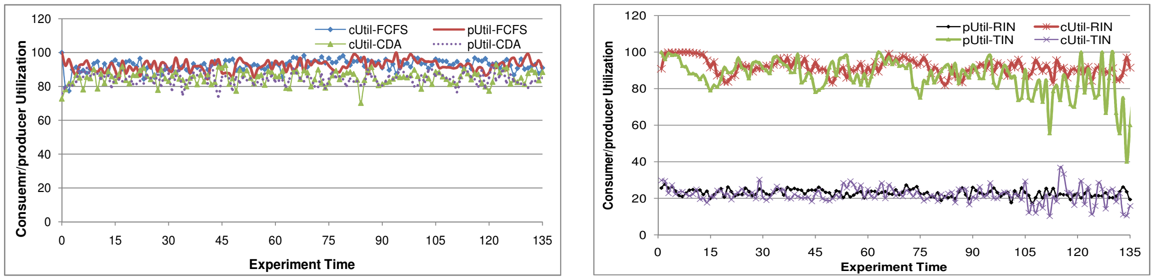
Analysis parameters are pheromone evolution, consumer/producer utilization, and average ask/bid price of the participating producer/consumer nodes. The formulas for calculating the pheromone value are described in Section 3.2. Consumer/producer utilization is calculated for each matchmaker according to the following formula:

$$(\frac{Matches}{N}) * 100$$

where $Matches$ represents count of matched resource-request/resource-offer pairs and N denotes the total number of resource-requests/resource-offers processed

	Ratio of Resources-Offers	Ratio of Resource-Requests
RIN	20%	80%
BN	50%	50%
RIN	80%	20%

Table 1: Evaluated Network Conditions.



(a) (b)

Figure 3: Consumer/producer utilization. **(a)** Consumer/producer resource utilization in BN condition of an adhoc grid. **(b)** Consumer/producer resource utilization in RIN & TIN conditions of adhoc grid.

by the matchmaker(s) in a unit time (60000 milliseconds).

The following simplifying assumptions are applied for the results presented in this paper. These assumptions will be relaxed in our future work. **(1)** One node can only submit one resource-offer/resource-request at any given time. **(2)** There exists at least one food source/matchmaker for each resource category during an experiment. **(3)** A newly joinig consumer/producer node knows about its food-source/matchmaker at the time of joining an adhoc grid.

6. Experimental Results & Discussion

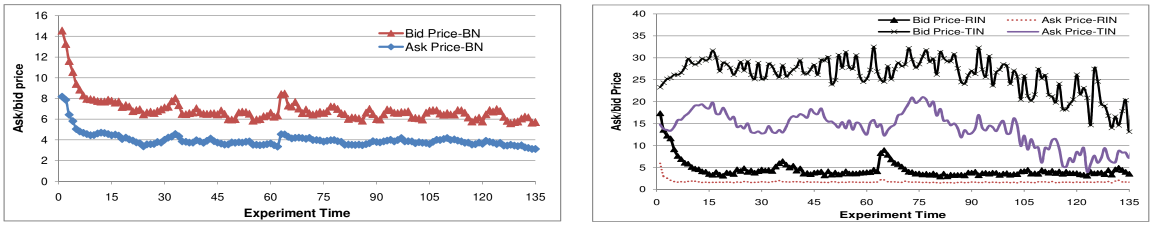
6.1. Consumer/Producer Resource Utilization

Consumer/Producer resource utilization is one of the evaluation parameters of the proposed nature inspired micro-economic based self-organization mechanism. Proposed micro-economic based mechanism is compared with the simplest and less compute intensive First Come First Served (FCFS). Average consumer/producer utilization of the participating nodes in adhoc grid with different matchmaking mechanisms in different network conditions is depicted in Figures 3a and 3b.

The consumer/producer utilization in RIN and TIN conditions is depicted Figure 3b. The task-resource ratio is $20\% - 80\%$ and $80\% - 20\%$ in RIN and TIN conditions respectively. In general, both the consumer utilization in TIN condition ($cUtil - TIN$) and the producer utilization in RIN condition ($pUtil - RIN$) are proportional to the ratio of scarce commodity in the respective network conditions. Whereas, the producer utilization in TIN condition ($pUtil - TIN$) and the consumer utilization in RIN condition ($cUtil - RIN$) vary between 80% and 100%. The higher consumer utilization in RIN and producer utilization in TIN conditions are due to the abundance of resource requests and resource offer of each category in the respective network condition during simulation. Therefore, a request in RIN and an offer in TIN condition get matched as soon as it is received by its respective food source (matchmaker). The higher fluctuations in $cUtil - RIN$ and $pUtil - TIN$ refer to the points during the simulation when ants change their resource categories and the adhoc grid becomes unstable. The consumer utilization in RIN ($cUtil - RIN$) and the producer utilization in TIN ($pUtil - TIN$) show a stable behavior and the variations in stable behavior are due to the change of consumer/producer resource category. In general, adhoc grid shows a stable behavior in terms of consumer/producer utilization and accommodates the dynamic resource category changes of the participating adhoc grid nodes according to their changing resource requirements.

Consumer utilization and producer utilization remains above 80% in CDA and FCFS schemes in BN condition. The fluctuations in the consumer/producer utilization refer to the change in resource category of the participating nodes in the adhoc grid. Similar consumer/producer utilization in both CDA and FCFS indicates that compute intensive nature of CDA does not affect the matchmaking efficiency of the individual matchmakers.

It can be concluded from the above discussion that a nature inspired ACO-based, self-organizing mechanism with CDA scheme is preferred over an ACO-based mechanism with FCFS scheme. The compute intensive CDA based mechanism performs as good as a less compute intensive FCFS mechanism in terms of consumer/producer utilization, and the pheromone value evolution. CDA based mechanism enables the individual consumer/producer nodes to value their resource requests and resource offers



(a) (b)

Figure 4: Ask/bid price evolution in different network conditions. **(a)** Ask/bid price evolution in BN condition of the adhoc grid. **(b)** Ask/bid price evolution in RIN & TIN conditions of the adhoc grid.

according to their previous experiences from the adhoc grid. The consumer/producer nodes can increase/decrease their bid/ask prices according to the resource demand/availability in the adhoc gird. Thus, a CDA based ACO mechanism enables the node level self-organization along with the system level self-organization.

6.2. Ask/Bid Price Evolution

As the consumer/producer nodes applied a history based dynamic pricing mechanism (Section 3.1) for determining the value of the resources requested from the adhoc grid or offered to the adhoc grid in CDA based scheme, therefore, the affect of ask/bid price on the proposed mechanism is also analyzed. Consumer bid price and producer ask price for matched pairs in a BN condition is depicted in Figure 4a. A match is found when the bid price of a request is higher than the ask price of an offer. As obvious from Figure 4a, bid price is always higher than the ask price for a matched request/offer pair.

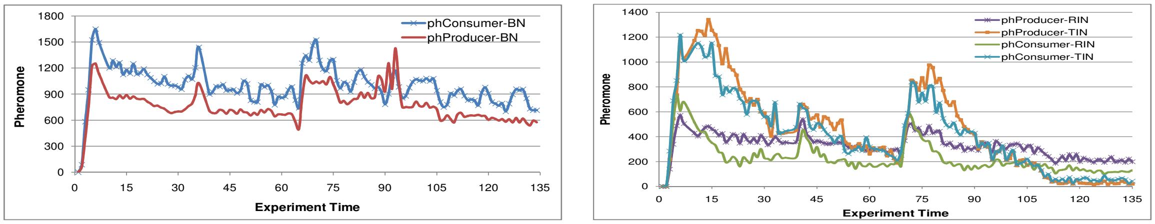
Furthermore, fluctuations in ask/bid price refer to the intervals when nodes change their resource category and consumer/producer nodes increase their ask/bid prices for their requested/offered resources. Bid price of consumer in RIN and TIN conditions and ask price of producers in RIN and TIN conditions depict a similar behavior as explained for BN condition (Figure 4b). The bid price in RIN condition is lower than that of the bid price in BN condition. As, there are more resources in RIN condition than in BN condition and, hence, the competition among consumer nodes is lesser in RIN condition than in BN condition. Lesser competition results in a lower bid price in RIN condition in comparison to BN condition. Furthermore, resource are scarce in TIN condition and there is more competition for available resources. This situation results in higher ask and bid prices in TIN condition in comparison with other network conditions. Ask/Bid price evolution helps in node level self-organization. Participating nodes increase/decrease their ask/bid price according to availability/scarcity of resources and attempt to increase their utility from adhoc grid.

6.3. Pheromone Evolution

The pheromone behavior of the individual consumer/producer nodes in BN condition in represented in Figure 5a, and in TIN and RIN conditions is represented in Figure 5b. Horizontal axis of both of these figures represent the experiment time and vertical axis represents the pheromone value of the participating consumer/producer nodes in BN, RIN or TIN conditions of the adhoc grid. The initial increase of the pheromone value for the consumer/producer nodes in BN, RIN and TIN conditions is followed by a decreasing trend that leads to a stable status of the adhoc grid. Similar to the individual category pheromone value evolution, the overall consumer/producer pheromone also evolves similarly in different network conditions. The pheromone pattern is disturbed after a change of the ant's distribution in different resource categories. The proposed algorithms enable the adhoc grid to restructure itself into different virtual resource and in attaining a stable state.

7. Concluding Remarks

An ACO inspired, micro economic based infrastructure level self-organizing approach for adhoc grids is presented in this paper. We used matchmaking performance as the basic factor for calculating the pheromone value. The proposed ACO inspired CDA based approach enables node level as well as system level self-organization and supports resource specialization in an adhoc grid. From the experimental results it can be concluded that the proposed mechanism gives a stable behavior of the system in resource management, and shows better load balancing.



(a) (b)

Figure 5: Pheromone evolution for all resource categories of adhoc grid. **(a)** BN condition of adhoc grid. **(b)** RIN & TIN conditions of adhoc grid.

Our future work will mainly focus on relaxing the simplifying assumptions assumed in this work. We would like to modify and evaluate the presented mechanism for non-atomic tasks and multiple simultaneous resource-requests being submitted by consumer/producer nodes.

References

[1] A. Chien, B. Calder, S. Elbert, and K. Bhatia, "Entropia: Architecture and performance of an enterprise desktop grid system," *Journal of Parallel and Distributed Computing*, vol. 63(5), pp. 597–610, 2003.

[2] D. P. Anderson, "BOINC: A system for public-resource computing and storage," in *Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing*, 2004.

[3] T. Abdullah and K. Bertels, "Nature-inspired self-organization in p2p ad hoc grids," in *The Seventh International Conference on Autonomic and Autonomous Systems*, pp. 134–139, 2011.

[4] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant System: Optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man and Cybernetics-Part B*, vol. 26, no. 1, pp. 29–41, 1996.

[5] M. A. Jaeger, H. Parzyjegla, G. Muhl, and K. Herrmann, "Self-organizing broker topologies for publish/subscribe systems," in *ACM symposium on Applied computing*, pp. 543–550, 2007.

[6] G. Ritchie and J. Levine, "A hybrid ant algorithm for scheduling independent jobs in heterogeneous computing environments," in *Proceedings of the 23rd workshop of the UK planning and scheduling special interest group*, (Cork, Ireland), pp. 1–7, 2004.

[7] A. Montresor, H. Meling, and O. Babaoglu, "Messor: Load-balancing through a swarm of autonomous agents," Technical report, University of Bologna, 2002.

[8] A. Andrzejak, S. Graupner, V. Kotov, and H. Trinks, "Algorithms for self-organization and adaptive service placement in dynamic distributed systems," Tech. Rep. HPL-2002-259, HP laboratories Palo Alto, 2002.

[9] S. Fidanova and M. Durchova, "Ant algorithm for grid scheduling problem," in *Proceedings of the 5th International Conference on Large-Scale Scientific Computations (LSCC '05)*, pp. 405–412, 2006.

[10] J. Cao, "Self-organizing agents for grid load balancing," in *Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing*, pp. 388–395, 2004.

[11] M. A. Salehi, H. Deldari, and B. M. Dorri, "Balancing load in a computational grid applying adaptive, intelligent colonies of ants," *Informatica*, vol. 32, pp. 327–335, 2008.

[12] T. Abdullah, K. Bertels, and L. O. Alima, "Ant colony inspired microeconomic based resource management in ad hoc grids," in *Proceedings of the 4th International Conference on Grid and Pervasive Computing*, (Geneva), pp. 189–198, May 2009.

[13] T. Abdullah, V. Sokolov, B. Pourebrahimi, and K. Bertels, "Self-organizing dynamic ad hoc grids," in *Proceedings of the 2nd IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshops*, (Venice), pp. 202–207, October 2008.

[14] B. Pourebrahimi, K. Bertels, S. Vassiliadis, and L. O. Alima, "A dynamic pricing and bidding strategy for autonomous agents in grids," in *6th International Workshop on Agents and P2P Computing*, 2007.

[15] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[16] T. Abdullah, L. O. Alima, V. Sokolov, D. Calomme, and K. Bertels, "Hybrid resource discovery mechanism in ad hoc grid using strucutred overlay," in *Proceedings of the 22nd International Conference on Architecture of Computing Systems*, pp. 108–119, March 2009.

[17] A. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems," in *Proceedings of the 18th IFIP/ACM International Conference on Distributed Systems Platforms*, vol. 2218, pp. 329–350, 2001.

[18] https://www.planet-lab.org/, PlanetLab Online, last visited December-2012.