

Load Balancing of Distributed Systems Based on Multiple Ant Colonies Optimization

¹Al-Dahoud Ali, ²Mohamed A. Belal and ¹Moh'd Belal Al-Zoubi
¹Department of Computer Science, Faculty of Information Technology,
Al-Zaytoonah University, Jordan
²Department of Computer Science, Halwan University, Egypt

Abstract: Problem statement: Ant Colony Optimization (ACO) provides a meta-heuristic optimization tool and collective intelligence model to several applications such as routing and load balancing. A lot of work found in the literature on using ACO in load balancing. However, as far as our knowledge, there was no work relating load balancing in distributed systems with ACO. **Approach:** In this study, a proposed ACO algorithm for load balancing in distributed systems will be presented. This algorithm is fully distributed in which information is dynamically updated at each ant movement. Multiple colonies paradigm will be adopted such that each node will send a colored colony throughout the network. **Results:** In this study, colored ant colonies are used to prevent ants of the same nest from following the same route and hence enforcing them to be distributed all over the nodes in the system and each ant acts like a mobile agent that carries newly updated load balancing information to the next visited node. **Conclusion:** Finally, the performance of the proposed ACO algorithm is compared with the work-stealing approach for load balancing in distributed systems.

Key words: Load balancing, ant colony optimization, distributed systems

INTRODUCTION

Distributed system load balancing is still an active area of research in which load balancer attempts to improve the performance of a distributed system by using the processing power of the entire system to smooth out periods of high congestion at individual nodes (Ferrari and Zhou, 1987; Zhou and Ferrari, 1987), this is done by transferring some of the workload of heavily loaded nodes to other nodes for processing. Decisions on how to balance loads among the nodes are either static (Rao *et al.*, 1979; Tantawi and Towsley, 1985; Pinter and Woltstahl, 1987; Chen and Shin, 1990) or dynamic (Ni *et al.*, 1985; Eager *et al.*, 1986; Shin and Chang, 1989; Xu and Hwang, 1993; Ali, 2000; 2001). A static decision is independent of the current system state. Static load balancing can also be viewed as a deterministic allocation of jobs in a system, where an overloaded node will transfer some of its jobs to another node with a certain probability, which is independent of the current system state. Although static load balancing is simple and easy to analyze with queuing models, but its potential benefit is limited, since it does not adapt itself to time-varying system state (Eager *et al.*, 1986). On the other hand, a dynamic

decision is depending on the system state at the time of the decision. When a dynamic load balancing is used, an over loaded node can transfer its jobs to other nodes using the information on the current system state. The dynamic policy is inherently more complex than any static policy because it requires that each node must to know the states of the other nodes. The load balancing algorithms are further divided into several clusters according to the amount of information required for them (Shin and Chang, 1989).

Artificial swarm intelligence, in particular Ant Colony Optimization (ACO), is a relatively new computational and behavioral paradigm for solving optimization and combinatorial problems and hence it can be used for load balancing; it is based on the principles that control the behavior of natural systems. In such a simulation model, many distributed agents evolve and interact with each other in order to reach a global goal, such as ant colonies and bird flocks. This approach emphasizes the distributed structure of the problem, direct or indirect interactions among relatively simple agents and also among the agents and their environment.

The application of swarm intelligence to networks problems arises when a group of autonomous programs

Corresponding Author: Al-Dahoud Ali, Department of Computer Science, Faculty of Information Technology,
Al-Zaytoonah University, Jordan

(agents) are working together. This is referred to be "Ant Colony Optimization" ACO or multi-agent system. Each individual or program or autonomous module can be represented as an agent, these multi-agents can be used for network applications such as finding the shortest path, routing, load balancing and management and so on.

Some related works to using ACO in load balancing are the use of multi-agents system, two algorithms has been proposed by (Heusse *et al.*, 1998): The first one is based on round trip routing agents that update the routing tables by backtracking their way after having reached the destination. The second one relies on forward agents that update the routing tables directly as they move toward their destination. On the other hand, Salehi and Deldari (2006) presents an echo system of intelligent, autonomous and cooperative ants. The ants in this environment can procreate and also may commit suicide depending on existing conditions. A new concept called Ant level load balancing is presented for improving the performance of the mechanism. Sim and Sun (2003a) has presented a Multiple Ant Colony Optimization (MACO) approach for load balancing in circuit-switched networks. MACO uses multiple ant colonies to search for alternatives to an optimal path. One of the impetuses of MACO is to optimize the performance of a congested network by routing calls via several alternatives paths to prevent possible congestion along an optimal path. In MACO, each group of mobile agents corresponds to a colony of ants and the routing table of each group corresponds to a pheromone table of each colony (Sim and Sun, 2003b). By adopting the MACO approach, it may be possible to reduce the likelihood that all mobile agents establish connections using only the optimal path (Sim and Sun, 2003b). The advantage of using MACO in circuit-switched routing is that it is more likely to establish connections through multiple paths to help balance the load but does not increase the routing overhead (Sim and Sun, 2003b). Balancea *et al.* (1996), used mobile agents to encapsulate tasks, in distributed memory message-passing computers. In order to increase system flexibility and to balance an arbitrary graph topology computing environment, in the proposed model, the agents have the possibility to explore, learn and share information about the load. The proposed model was compared with deterministic DASUD and evolutionary algorithms. The allocation on physical processors of the proposed agent algorithm was generally better than that given by the DASUD algorithm when the task's time requirement is increased.

MATERIALS AND METHODS

Ant colony optimization: The intelligent group-behavior characterizes the whole colony of social insects, such as ants; examples of that emergent behavior include foraging and nest building. This collective behavior can be viewed as a powerful problem-solving system that can solve problems such as load balancing. Starting from simple interacting agents with rules of interaction among individuals and between individuals and the environment, the ant colony can provide intelligence far away from any individual capability. Properties associated with their group behavior, such as self-organization, flexibility and robustness, can be shown as characteristics that should exist in complex system for control, optimization and problem solving techniques.

The problem of how almost blind ants could cooperate in order to find the shortest path to the food source has attracted ethologists for several years. It was found that they communicate by placing pheromone trails, a chemical substance that attracts other ants, along with their movements towards the source of food (Dorigo *et al.*, 1996). An ant will prefer with high probability to follow the path having more pheromone, thus enforcing the trail with its own pheromone. This collective behavior is called autocatalytic behavior, in which it is defined to be a positive feedback process in which a process reinforces itself to enhance its performance, this feedback enforces the process towards a rapid convergence towards the final solution (Dorigo *et al.*, 1996).

The ants do not forage only for finding food, rather they forage in order to wander, search, return to home, attract to a target, trace pheromone and carry food (Dorigo and Di Caro, 1999; Dorigo, 2001). The pheromones also evaporate over time. As a consequence, the pheromone will become less detectable after a while and the longer trails will be less attractive to other ants (Tarasewich and Patrick, 2002). Ants can construct the shortest path from their nest to the source of food, through the use of pheromone trails. The ant leaves some quantities of pheromone on the ground while it walks. The next one will sense it and based on a probability proportional to the amount of pheromone, it will choose its path (http://www.nero.uni-onn.de/lehreWS02/skript_al_pfeifer_chap4.pdf). However, artificial ants have some major differences with real ones: First, ants have a memory. Second, ants are not completely blind, finally, the time of artificial ants is discrete (Krohn, 2001).

Generally, Ant Colony Optimization (ACO) is a general-purpose heuristic algorithm, which can be used

to solve different combinatorial optimization problems (Dorigo *et al.*, 1996). In ACO, the search activities is distributed over artificial ants, which mimic the behavior of real ants. The advantages of that system are positive feed-back, distributed computation and the use of a constructive greedy heuristic. Positive feed-back refers to the ability to rapid discovery of good solutions. The ACO is also a population-based approach in which parallelization can easily be achieved (Dorigo *et al.*, 1996).

One of the main important concepts added by ACO is that finding the solution is an emergent process of the cooperation and the interaction of simple agents. Another concept is the use of indirect stigmergetic communication by changing the environment (http://www.nero.uni-onn.de/lehreWS02/skript_al_pfeifer_chap4.pdf). Ant algorithms can be considered multi-agents systems that exploit artificial stigmergy as a means for coordinating artificial ants for the solution of computational problems

The collective activities of social insects are self-organizing, meaning that complex group behavior emerges from the simple interactions of individuals. The results of self-organization are of global nature, but come basically from local information and interactions (Tarasewich and Patrick, 2002). The interaction within a society of insects can take one of the two forms: direct and indirect. Direct interactions can take the form of bodily contact, visual contact and food exchange. Indirect interaction is also important, it can occur when agents exchange information through the environment in which they exist. Thus the storage of information occurs at the colony level as well as individual level. This cooperation through modification is called stigmergy (Krohn, 2001).

Basic operations of ant colony optimization systems:

In this part; the basic and necessary operations that should exist in any ACO system, including our proposed model are summarized and discussed. In ACO, the ants are adaptive, i.e., if the environment changes, the ants will look for a better solution. The ACO is suitable to discrete optimization problems (http://www.nero.uni-onn.de/lehreWS02/skript_al_pfeifer_chap4.pdf). The main characteristics of AC system are positive feedback, distributed computation and a constructive greedy heuristic (Dorigo *et al.*, 1996).

In the following; a short description will be given to the basic operations involved in ACO:

Stigmergy: The indirect communications via the pheromones is an example of positive feedback called

autocatalytic behavior. Conversely, the negative feedback is introduced through the evaporation of pheromone trail (Krohn, 2001).

The characteristics of the swarm intelligence model of ACO are the new added concepts of self-organization and stigmergy. In a distributed system such as Ant System, communication between agents is of the great importance. The form of communication is indirect. This communication can be viewed as a space deformation of the system in which ants reside (Krohn, 2001). The behavior of the whole ant colony is highly structured, interactions is based on a very simple flows of information (Colorni *et al.*, 1992).

The general idea of the ACO algorithm is that two opposite forces enforce the optimization method to reach the solution. The first force is autocatalytic process that drives the good solutions to emerge. The other force is the greedy force that always selects the first shortest path to find a solution. Neither of the two forces can reach the optimal solution alone. But when they are working together, it seems that the greedy force can give the right suggestions to the autocatalytic force and let it converge to the optimal value very quickly (Colorni *et al.*, 1992). However, one of the problems of ACO is that whenever there is a significant change in the environment, it takes some time before ants discover it and modifies their information (http://www.nero.uni-onn.de/lehreWS02/skript_al_pfeifer_chap4.pdf).

Pheromone evaluation: The pheromone that each ant lays attracts the following ants so that they will likely search in the same region of the search space. In general, it is assumed that pheromone evaluation is done locally by the ants. Merkle *et al.* (2000) proposed to extend the local view of the ants by a look-forward strategy.

Merkle *et al.* (2000) showed that the behavior of an ant algorithm can be improved scientifically when the ants use more information than just the local information values for their decision.

Information exchange in ant colony optimization: Information exchange between ant colonies is one of the most important factors that influence the optimization behavior.

It was shown that several colonies could perform very effectively with a little exchange of information (Middendorf *et al.*, 2000). Improvement is achieved by adding a little bit of information exchange. It was also shown exchanging the local best solution only and not to often is sufficient to have a very good performance.

In (Middendorf *et al.*, 2000), different information exchange policies are investigated. It was shown that exchanging a small amount of information significantly affects the performance of the optimization performance. This can be done by permitting the other colony to make benefit from good solutions obtained by other colonies.

The proposed strategy: The load balancing strategy, in this work, is time dependent and follows the natural dynamics of the pheromone in real life. At a specified time period, each node will act like a nest and sends number of ants (the number of ants depends on the loading status of each node; the overloaded and underloaded nodes will send more ants).

Each ant will travel a tour (the tour length depends on the size of the system and the loading status of the node). Finally, the proposed algorithm will be compared with the standard work-stealing algorithm

Our model is fully distributed, i.e., each node nodes behaves independently as well as each ant or agent, this mean that each node or ant is autonomous. Table 1 represents the attached information to each node or ant.

Table 1: The attached information to each node or ant

Node ID	Loading status	Pheromone
---------	----------------	-----------

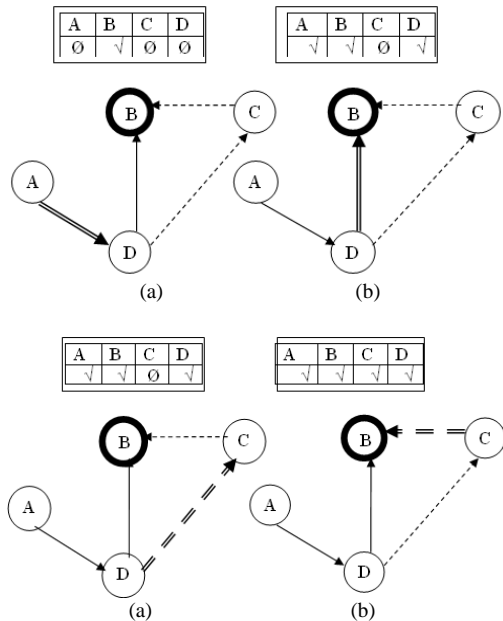


Fig. 1: Example of the operations of the model. (a) An ant moves from A to D; (b) The ant moves from D to B; (c) Another ant moves from D to C; (d) The ant moves from C to B

In our model, each node contains information about other nodes in the system (Fig. 1). At the initial state, the table entries are Null.

In each ant tour, the ant will carry the updated information about all nodes that the ant has been passed through. Upon arrival of the ant at each node, the following actions will be done:

- If the node does not have the information contained in the ant table, these information will be passed to the node table without any update
- If the node contains information that does not exist in the ant's table, the ant table will be updated
- If both of them share the same information, the newly updated one will replace the other

RESULTS AND DISCUSSION

The proposed strategy was simulated and tested, the number of nodes in the distributed system was assumed to be 30, an ant is assumed to travel from one node to another in 1 time step, each task is assumed to take 40 steps. In order to emphasis the efficiency of the proposed algorithm, we consider the case when the distributed system is very irregular; it is assumed that node number 1 is busy with 60 tasks and the other nodes are idle, Fig. 2 shows the efficiency of both the work-stealing approach and ant-colony approach. It is clear that the efficiency of the ant-colony approach comes from the ability to distribute loading information through all the nodes, the tour of ants was randomly chosen, however the cleverness of the ants to carry the new loading-status to each nodes increases the chance of each node to quickly find a good food source or a busy node.

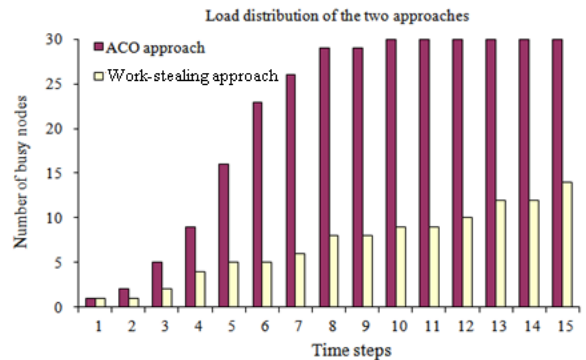


Fig. 2: Load distribution: The number of busy nodes in the time-steps

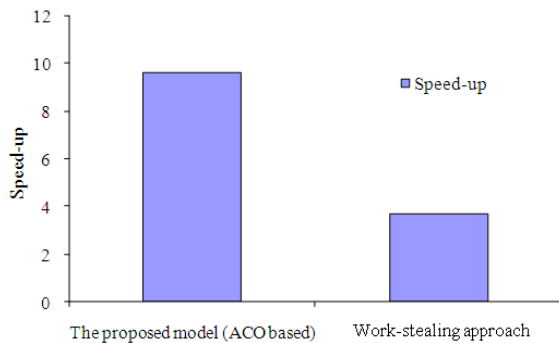


Fig. 3: A comparison of the speed-up of the proposed model and the work-stealing model

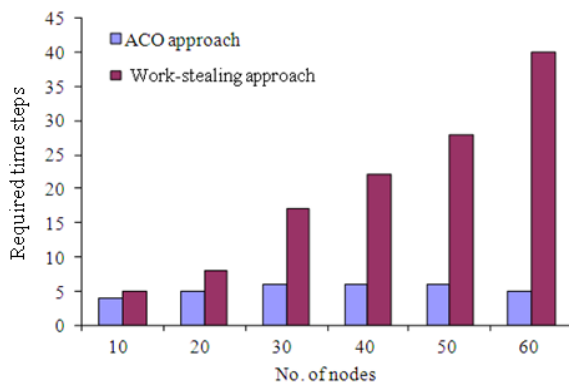


Fig. 4: Load balancing dynamics: The required time steps needed to reach a load distribution of 50% versus the number of nodes

Figure 3 compares the speed-up of proposed model versus the standard work-stealing approach. Based on the above settings, the proposed model achieved a speed-up of 9.6, while the work-stealing approach's speed-up was 3.7.

Figure 4 studies the effect of enlarging the number of nodes against the number of steps needed to raise the load distribution up to 50%. It is shown that as the number of nodes becomes larger, the time-steps needed for the work-stealing approach increases dramatically, while the proposed model remains almost constant.

This approach and its results give an example that highlights the importance of the swarm system in the decision-making process in general, where each agent can play a small role and the global behavior could be robust and reliable.

CONCLUSION

In this study, an approach, for load balancing in distributed systems based on multiple ant colonies

optimization, was proposed. The use of multiple nests, or ant colonies in the search process, helped in raising the rate of information exchange all over the nodes in the system. Besides, the dynamical information exchange and the fully distribution are other main characteristics that distinguishes our approach.

Results have shown the efficiency of the proposed model compared with the standard work-stealing algorithm in terms of number of busy nodes and the elapsed time to reach an efficiency of 50%.

REFERENCES

- Ali, A.D., 2000. A robust load balancer for heterogeneous distributed systems type-A. Proceeding of the Symposium on Performance Evaluation of Computer and Telecommunication Systems, July 16-20, Vancouver, British Columbia, Canada, p: 1-1. <http://www.scs.org/confernc/scsc/scsc00/text/spec2.html>
- Ali, A.D., 2001. A dynamic cluster constructor for load balancing in big heterogeneous distributed systems. Proceeding of the Symposium on Performance Evaluation of Computer and Telecommunication Systems, July 15-19, Orlando, Florida, USA., pp: 47-55.
- Bulancea, C., B. Paechter and A. Carter, 1996. Using ACO metaheuristics on load balancing algorithm. Trans. Syst. Man Cybernet.
- Chen, M.S. and K.G. Shin, 1990. Subcube allocation and task migration in hypercube multiprocessor. IEEE Trans. Comput., 39: 1146-1155.
- Colomi, A., M. Dorigo and V. Maniezzo, 1992. Distributed optimization by ant colonies. Proceeding of the 1st European Conference on Artificial Life, (ECAL'92), Elsevier Publishing, Paris, France, pp: 134-142.
- Dorigo, M. and G. di Caro, 1999. Ant colony optimization: A new meta-heuristic. Proceedings of the 1999 Congress on Evolutionary Computation, July 6-9, Mayflower Hotel, IEEE Xplore Press Washington DC., USA., pp: 1470-1477.
- Dorigo, M., 2001. Swarm intelligence, ant algorithms and ant colony optimization, adaptive computation and simulation: Swarm intelligence-25/07/01. Budapest. <http://www.mti.ugm.ac.id/~adji/courses/resources/doctor/mobile/AntIntelligence.pdf>
- Dorigo, M., V. Maniezzo and A. Colomi, 1996. Ant system: Optimization by a colony of cooperating agents. IEEE Trans. Syst. Man Cybernet. B Cybern., 26: 29-41. PMID: 18263004
- Eager, D.L., E.D. Lazowska and J. Zahorjan, 1986. Adaptive load sharing in homogeneous distributed systems. IEEE Trans. Software Eng., 12: 662-673.

- Ferrari, D. and S. Zhou, 1987. An empirical investigation of load indices for load balancing applications. Proceeding of the 12th International Symposium on Computer Performance Modeling Measurement and Evaluation, Dec. 7-9, North-Holland Publishing Co., Amsterdam, The Netherlands, pp: 515-528. <http://portal.acm.org/citation.cfm?id=725013>
- Heusse, M., S. Guerin, D. Snyersy and P. Kuntz, 1998. A new distributed and adaptive approach to routing and load balancing in dynamic communication networks. University of Washington. <http://staff.washington.edu/paymana/swarm/heusse98-aco.pdf>
- Krohn, J., 2001. Ant algorithms and the swarm intelligence model of problem solving. Proceedings of the Conference on the UMM Computer Science Discipline Seminar, (CCSDS'01), UMM University of Minesota Morris, Morris, pp: 1-6. <http://www.morris.umn.edu/~krohnk/pdf/AntSystem.pdf>
- Merkle, D., M. Middendorf H. Schmeck, 2000. Pheromone evaluation in ant colony optimization. Proceedings of the 26th Annual IEEE International Conference on Industrial Electronics, Control and Instrumentation, Oct. 22- 28, IEEE Xplore Press, Nagoya, Japan, pp: 2726-2731.
- Middendorf, M., F. Reischle and H. Schmeck, 2000. Information exchange in multi colony ant algorithms. Lecture Notes Comput. Sci., 1800: 645-652. DOI: 10.1007/3-540-45591-4_87
- Ni, L.M., C.W. Xu and T.B. Gendreau, 1985. A distributed drafting algorithm for load balancing. IEEE Trans. Software Eng., SE-11: 1153-1161.
- Pinter, S.S. and Y. Woltstahl, 1987. On mapping processes to processor in distributed systems. Int. J. Parall. Programm., 16: 1-15.
- Rao, G.S., H.S. Stone and T.C. Hu, 1979. Assignment of tasks in a distributed processor system with limited memory. IEEE Trans. Comput., C-28: 291-299. <http://portal.acm.org/citation.cfm?id=1310399>
- Salehi, M.A. and H. Deldari, 2006. Grid load balancing using an echo system of intelligent ants. Proceedings of the 24th IASTED International Conference on Parallel and Distributed Computing and Networks, Feb. 14-16, ACTA Press, Innsbruck, Austria, Anaheim, CA, USA., pp: 47-52.
- Shin, K.G. and Y.C. Chang, 1989. Load sharing in distributed real-time systems with state-change broadcasts. IEEE Trans. Comput., 3: 1124-1142.
- Sim, K.M. and W.H. Sun, 2003b. Ant colony optimization for routing and load-balancing: Survey and new directions. IEEE Trans. Syst. Man Cybern.-Part A: Syst. Hum., 33: 560-572.
- Sim, K.M. and W.H. Sun, 2003a. Multiple ant colony optimization for load balancing. Lecture Notes Comput. Sci., 2690: 467-471. DOI: 10.1007/b11717
- Tantawi, A.N. and D. Towsley, 1985. Optimal static load balancing in distributed computer systems. J. Assoc. Comput. Mach., 32: 445-465.
- Tarasewich, P. and P.R. McMullen, 2002. Swarm intelligence: Powers in numbers. Commun. ACM, 45: 62-67.
- Xu, J. and K. Hwang, 1993. Heuristic methods for dynamic load balancing in a message-passing multicomputer. J. Paral. Distribut. Comput., 18: 1-13.
- Zhou, S. and D. Ferrari, 1987. An experimental study of load balancing performance. University of California, Berkeley, CA., pp: 26.