*International Journal of Computer Applications (0975 – 8887)*
*Volume 128 – No.15, October 2015*

# A Survey on Ant Inspired Metaheuristic Algorithm-Parallel Approaches

Shobhit N. Sharma
Gyan Ganga
College of Technology

Vikram Garg
Gyan Ganga
College of Technology

## ABSTRACT
Although ant is not one of those smart creatures, when swarm, however co-operate with each other while foraging(in search of food) they have this great ability to unearth an optimal solution to their problem thus coming up with the shortest path from their nest to the food source in case of foraging. Ants grant excellent efficiency while solving combinatorial problems and also have the potential of combining with other algorithms with ease. This survey includes: 1) Generalized ant inspired algorithm and different applications of ant Colony Optimization (ACO). 2) Proposed parallel approaches to those applications. As ACO is intrinsically parallel so in this survey GPU implementation using OpenCL is proposed to parallel approaches and only those application areas are explored whose parallel approaches are discussed.

## Keywords
Ant Colony Optimization(ACO), Swarm intelligence, meta-heuristics, OpenCL.

## 1. INTRODUCTION
With a rapidly growing world and technology there are many challenges to solve complex problems in an efficient way and in a quick time. Ant colony optimization is one of the optimization techniques that have the capability of combining with other algorithms and improving their efficiency. It gives even better results when it is parallelized as it is intrinsically parallel. ACO is one of the naturally inspired metaheuristic algorithm. ACO takes the inspiration from ants.In ACO, colony of ants moves from nest to food source through different paths initially. While moving they lay a special chemical substance called pheromone. This pheromone is used for stigmergy (ants interaction). Using this all ants select an optimal path from nest to food source after some time. This time period is the convergence time. Thus, ant colonies are capable of finding shortest path for their purpose which is the optimal solution. In fact, Ant colony optimization is a multi-agent system as here multiple colonies of ants are used to come across a solution for the given problem. The role of agents is depicted by artificial ants. These artificial ants explore for the solution space probabilistically to uncover different solutions and finally converge to find the optimal solution out of these candidate solutions.

The motivation to ACO comes from swarm intelligence. In fact, ACO is a subset of swarm intelligence and is inherently a parallelizable technique [1]. In this survey, first we have generalized the ACO algorithm and then, we have explored few application areas in which initially we have explained the main concrete problem which have an exponential worst case complexity or have large applications but require more time. Finally, parallel approach and GPU implementation to make it even more efficient is proposed using OpenCL [2].

For GPU implementation of ACO using a standard frame-work like OpenCL, these multiple colonies of ant (MACO [3]) or ants of a single colony will traverse the graph or solution space concurrently to find solution efficiently. Thus, parallel implementation will speed-up the algorithm and decrease the convergence time. Section II discusses the background of ACO and generalized ant inspired algorithm and diverse application areas of ACO. Parallel approaches to some of the applications are explained in Section III. Further, Section IV concludes the survey with some optimization techniques which are applicable on OpenCL code thus enhancing its efficiency.

## 2. BACKGROUND AND APPLICATIONS OF ACO
Ant colony optimization was proposed in early 90s. The first algorithm proposed was Ant system (AS) by Dorigo [4] [5] subsequent to which many variants were proposed. MAX-MIN ant system and $AS_{rank}$ are among the efficient algorithms of ACO. All the variants use a general concept in which stochastic state transition rules which are function of pheromone value and heuristic information and two other parameters and are used. and are used to control the effects of pheromone and heuristic information so that in different scenarios we can adjust these two values accordingly and obtain the desired optimal result. Alpha and beta are adjustable parameters to adjust relative influence of probability and heuristic information. Local pheromone updates are done by the ant after local solution construction. After the evaluation of all local solutions global pheromone update is performed thereafter. The two main phases of ACO are solution construction and pheromone update. During Solution construction decisions are undergone on the basis of above probabilistic rule and on discovery and evaluation of local optimal solutions, global pheromone update is carried out.

Loop

Loop

Ants or colony of ants will apply state transition rules and local pheromone update.

Until each finds a local solution

Global pheromone update after solution evaluation

Until no further updates in optimal solution or end condition

Rules used are different in different ACO algorithms

like AS, $AS_{rank}$, Max-Min AS, ACS

This choice depends on the type of problem we encountered with. There exists a diverse problem range that acquires bulk of time accompanied with a large search space. Consequently, their performance can be enhanced by using some metaheuristic algorithms in an attempt to shrink the search space. Some of the metaheuristic algorithms are particle swarm optimization, ant colony optimization and genetic algorithm.ACO unearth a large number of applications, especially problems which are NP-Hard. As NP-Hard problems have an exponential worst case complexity [6], so ACO can be pooled with them to trim down the complexity to polynomial time which can further be improved if it is parallelized. There are an assortment of areas where ACO algorithms are applied like in routing, assignment problems, scheduling problems [7], data mining and classification [8], emergency path rescue [9] and wireless sensor networks [10]. This survey will be confined to only a handful of these problems therefore restricting our discussion to those.

- **ACO applications in routing problems**

1) Concrete Problem: In routing there are various applications like sequential ordering, vehicle routing and others. Vehicle routing problem(VRP) [4] is actually a distribution management problem in which we need to serve n customers located at some given distance from a central depot. We are provided with a fleet of vehicles, customers demand and travel time between two customers. The requirement is to minimize the travel time starting and ending at depot and each customer should be visited exactly once. Vehicle routing problem with time limitations is referred as VRPTW [4] vehicle routing problem with time window constraints.

Our goal is:

- to minimize number of vehicles used and
- to minimize total travel time [4] along with the constraint to visit each customer only once.

2. Ant inspired approach: Till date the most successful ACO algorithm applied on VRPTW is multiple ant colony system (MACS). In MACS-VPRTW multiple colonies of ants are used to attain each of the above specified goals. The core idea is to reduce the convergence time, thats why both colonies are implemented parallel.

The constructed graph consists of the node representing customer locations and central depot which we call the source. Weights assigned to the edges connecting the nodes are the measure of distance between them and the pheromone trails $\tau_{ij}$associated with each edge are the measure of desirability of visiting cutomer j after i. Initially, these weights are assigned a constant value. Then each ant uses a probabilistic rule to take decisions and update the pheromone on the edges. Both the colonies of ants having different objectives use different set of pheromones. When both the colonies converge to local solutions satisfying the above two constraints, evaluation is done and final global optimal solution is established.

ACO is accompanied by some local search procedure like simulated annealing and 2-opt local search. In local search generally some permutations or exchanges are performed in the local solution aimed to uncover closer optimal solutions in a quicker time.

- **ACO applications in Scheduling problems**

1) Concrete Problem: Scheduling problems in a general-ized way deals with assigning available resources to tasks. There a number of ACO algorithms proposed for scheduling problems ranging from job shop problem(JSP) and open shop problem(OSP) to permutation flow shop problem(PFSP). In flow shop scheduling(FSP) [7] problem independent jobs and machines are given and these jobs are to be assigned to machines. Every job has some operations and these operations are to be processed on one of the machines available in a non-preemptive fashion. Our goal is to assign all these jobs to machines in such a way, that all the operations are processed in a minimum time period. The total completion time thus obtained should be minimum. In this scenario, initially a different order of job execution on machines will be generated followed by preparation of Gantt chart depicting the completion time of jobs. The order which offers minimum completion time span in the Gantt chart acts as our solution.

2) Ant inspired approach: here will be a number of permu-tations to this assignment problem, so ACO is applied to this problem in which each ant generates a solution concurrently and among these solutions best one is selected and global pheromone update rule is applied. Therefore, using probabilis-tic rules and pheromone update rule we manage to drop off our search space and eventually the number of permutations and ACO leads to comparatively faster solution detection. In this problem, we can use the standard and earliest form of ACO which is Ant system [4]. So state transition and update rules of Ant system will be applied and a table Job/order will be maintained having each cell representing pheromone value associated with assigning job a particular order. Here also some local search procedure will be used in which some jobs will be interchanged to come up with closer optimal solutions.

- ACO applications for Ad hoc networks

1) Concrete Problem: In mobile ad hoc network (MANET) [1] there is no fixed network topology as all the nodes are mobile which described the reason they are referred as infrastructure less networks. In ad hoc networks routing approaches consists of table driven approaches and on demand approaches. We will be using on demand routing here. One algorithm named Ant colony based routing algorithm(ARA) has already been implemented for ad hoc networks in [1]. Our goal is to trace the shortest path between each pair of nodes that is APSP for MANET.

2) Ant inspired approach: Here every node will maintain a routing table having all the nodes in the network and pheromone value indicating the preference with which this node can be selected from the neighboring nodes. Ant keeps the track of all the visited nodes and maintains a stack of these nodes. This offers convenient selection of the highest probability node from the unvisited nodes which may lead to the best path.

## 3. PARALLEL APPROACHES AND NEW DIRECTIONS

Many parallel approaches have been given earlier for ACO algorithms but in this survey we have proposed GPU implementation using OpenCL. Proposed parallel approach will have two components, one will be implemented on CPU referred as host code and the other will be implemented on GPU using OpenCL called kernel [2].The host code will control the execution and memory transfers between different memory sections and also govern the execution of kernel. Kernel implemented using OpenCL will have the code which will run parallel on GPU. GPU will be responsible for implementation of parallel approach of solution construction. Each work item will invoke kernel.In case of serial implementation there are multiple cycles but here all one ant or one colony of ant will be responsible for one cycle and this will be represented by work item [2] and operate concurrently. Whenever a new cycle starts kernel of OpenCL is invoked.In case of multiple constraint satisfaction problem we can implement parallel multiple ant colonies, one for each constraint rule. Each will find solution satisfying particular constraint and after the evaluation of those solutions final solution will be selected.

**1. Algorithm**

*Initialization*

*Loop In-parallel do*

*Ants or colony of ants will apply state transition rules and local pheromone update.*

*Until each finds a local solution*

*Global pheromone update*

*Until no further updates in optimal solution.*

When all ants or ant colonies find the solutions then at the time of global pheromone update synchronization and memory consistency is necessary which is provided using barriers in OpenCL [2].

**1) Memory management:** There are different sets of memory defined by memory model of OpenCL. Private memory is there for every work item. All the work items of same work group share same local memory. And different work groups share same global memory so synchronization is needed. Frequently accessed data like local solution details and local pheromone values are stored in local memory whereas write once data like final path information and global pheromone values are stored in global memory.

### B. Parallel approaches for routing problem and ad hoc net-works

In vehicle routing problem algorithm MACS-VPRTW, we will implement two colonies, one for each objective and their local pheromone values are stored in local memory and global pheromone matrix is stored in global memory which will be updated after evaluation of local optimal solutions. Final optimal path information is also stored in global memory. Graph information is stored in a read only manner in global memory as well. Each work item will update its local pheromone value in matrix concurrently.

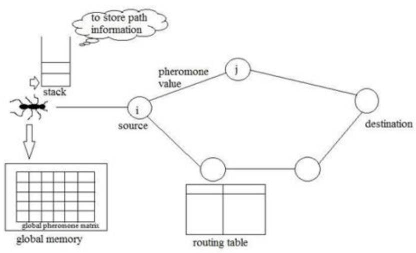


**Fig-1. Illustrating global pheromone matrix and stack for every ant for routing**

Similarly, in ad hoc networks each ant is represented by a work item which is identified by local and global id. And each work item will store its routing table and stack of the nodes visited in local memory. Performance of routing can further be improved by applying some load balancing techniques as explained in [3].

## 4. CONCLUSION

In this survey we have exposed basic solution finding capabilities of ACO with a good convergence rate. Furthermore, we proposed some parallel approaches to existing algorithms to come across an optimal solution in an efficient way. We can further increase the efficiency of GPU based implementations by means of some of the optimization techniques like memory coalescing and vectorization for OpenCL implementation [2].

## 5. REFERENCES

[1] M. Islam, P. Thulasiraman, and R. Thulasiram, “A parallel ant colony optimization algorithm for all-pair routing in manets,” in Parallel and Distributed Processing Symposium, 2003. Proceedings. International, pp. 8 pp.–, 2003.

[2] A. Munshi, B. Gaster, T. G. Mattson, J. Fung, and D. Ginsburg, OpenCL Programming Guide. Addison-Wesley Professional, 1 ed., July 2011.

[3] K. M. Sim and W. H. Sun, “Ant colony optimization for routing and load-balancing: survey and new directions,” Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on, vol. 33, no. 5, pp. 560–572, 2003.

[4] M. Dorigo and T. Stutzle,¨ Ant Colony Optimization. Scituate, MA, USA: Bradford Company, 2004.

[5] M. Dorigo, M. Birattari, and T. Stutzle, “Ant colony optimization,” Computational Intelligence Magazine, IEEE, vol. 1, no. 4, pp. 28–39, 2006.

[6] M. R. Garey and D. S. Johnson, Computers and Intractability; A Guide to the Theory of NP-Completeness. New York, NY, USA: W. H. Freeman & Co., 1990.

[7] R.-M. Chen, S.-T. Lo, C.-L. Wu, and T.-H. Lin, “An effective ant colony optimization-based algorithm for flow shop scheduling,” in Soft Computing in Industrial Applications, 2008. SMCia ’08. IEEE Conference on, pp. 101–106, 2008.

[8] D. Martens, M. De Backer, R. Haesen, J. Vanthienen, M. Snoeck, and B. Baesens, “Classification with ant colony optimization,” Evolutionary Computation, IEEE Transactions on, vol. 11, no. 5, pp. 651–665, 2007.

[9] B. Jin and L. Zhang, “An improved ant colony algorithm for path opti-mization in emergency rescue,” in Intelligent Systems and Applications (ISA), 2010 2nd International Workshop on, pp. 1–5, 2010.

[10] Y. Lin, J. Zhang, H.-H. Chung, W. H. Ip, Y. Li, and Y. hui Shi, “An ant colony optimization approach for maximizing the lifetime of heteroge-neous wireless sensor networks,” Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, vol. 42, no. 3, pp. 408–420, 2012.