

HYBRIDIZATION OF MODIFIED ANT COLONY OPTIMIZATION AND INTELLIGENT WATER DROPS ALGORITHM FOR JOB SCHEDULING IN COMPUTATIONAL GRID

P. Mathiyalagan¹, S. N. Sivanandam² and K. S. Saranya³

¹Department of Computer Science and Engineering, Sri Ramakrishna Engineering College, India

E-mail: p_mathi2001@yahoo.co.in

²Department of Computer Science and Engineering, Karpagam College of Engineering, India

E-mail: sns12.kit@gmail.com

³PSG College of Technology, India

E-mail: r_balaus@yahoo.com

Abstract

As grid is a heterogeneous environment, finding an optimal schedule for the job is always a complex task. In this paper, a hybridization technique using intelligent water drops and Ant colony optimization which are nature-inspired swarm intelligence approaches are used to find the best resource for the job. Intelligent water drops involves in finding out all matching resources for the job requirements and the routing information (optimal path) to reach those resources. Ant Colony optimization chooses the best resource among all matching resources for the job. The objective of this approach is to converge to the optimal schedule faster, minimize the make span of the job, improve load balancing of resources and efficient utilization of available resources.

Keywords:

Grid Computing, Grid Scheduling, Ant Colony Optimization, Intelligent Water Drops, Pheromone

1. INTRODUCTION

In recent years, there are many complex scientific problems that need huge space and computing power. Grid computing is a new technology for solving those complex problems.

Job scheduling is a process of finding out optimal resource for the incoming jobs and executes those jobs in the corresponding resource. There are three main phases of job scheduling [1]. Phase one is resource discovery which involves selecting a set of resources to which user have access and which meets the job requirements specified by user. Phase two is system selection wherein from a group of possible resources which meets job requirements, a single resource is selected. Phase three is job execution where the job is executed in the selected resource.

Job scheduling is a fundamental issue in achieving high performance in grid computing systems as the resources are distributed heterogeneously and the jobs have to be remotely executed at times. In order to overcome issues in scheduling heuristic approaches such as simulated annealing, hill climbing, genetic algorithm have been tried. Ant colony optimization is one of those heuristic approaches proposed by Dorigo [3]. Job scheduling is a NP- Complete problem. ACO has been used to solve many NP- hard optimization problems [4]. ACO is capable of solving both static and dynamic problems. As job scheduling problem is dynamic in nature, heuristics of Ant colony optimization is applied to solve grid scheduling problem [5].

Ant colony optimization is a biologically inspired population based search method that seeks to converge to a solution using

an evolutionary process. ACO follows the principles of real ant colonies where each ant is inspired by pheromone trail laid by another ant in order to move to food from nest and thereby having indirect communication between them known as stigmergy. Initially each ant starts its tour randomly in order to search for food from nest and as time passes all paths from food to nest will be explored by ants and they finally converge to a path which is the shortest path and also pheromone content will be high in that particular path as more number of ants follow the same path. Pheromone evaporation decreases the pheromone trail in a path exponentially as time passes. Naturally ants take long time in order to find shortest path. This concept is used in ACO where artificial ants have been constructed to find the optimal resource for the incoming jobs by making use of pheromone trail and pheromone evaporation factors.

Intelligent water drops (IWD) is an optimisation algorithm that uses a swarm of water drops to collectively search for optimal solutions in the environment of the given problem proposed by Hamed Shah-Hosseini. It is based on how the natural water drops from rivers reach the destination (lakes, seas or oceans) by traversing the search space. IWD was able to found optimal or near optimal solution for optimization problems like TSP, n-queen and knapsack problems. IWD easily adapts to changes in dynamic environment. Time to convergence in IWD is better compared to that of other swarm intelligence approaches. Hence it used here to solve job scheduling problem in grid. Artificial water drops in the algorithm imitates all the concepts of natural water drops.

2. RELATED WORKS

There are various versions of ACO. They are Ant system [6], Max-Min Ant system [7], Rank -based Ant system [8] and Elitist Ant system [9]. There are many researches about job scheduling in grid using ACO.

Yaohang Li [10] applied bio-inspired adaptive mechanism of ants in grid. The objective here is to minimize the makespan. A pheromone table of resources are maintained where the resource with high execution time receives higher pheromone value. The pheromone values are updated based on current status of resource. Incoming jobs are assigned to machines with high pheromone value. In this paper, the results of bio-inspired algorithm is compared with other static scheduling algorithms and stated that the performance of ACO in terms of robustness and adaptability is good compared to other static scheduling algorithms.

Hui Yan [11] modifies existing ant algorithm for job scheduling in grid. Pheromone update is done based on two parameters i.e. encouragement factor and punishment factor. If a assigned job is successfully completed by a resource, encouragement factor is considered to deposit additional pheromone on that particular resource so that it favours the next coming jobs to choose the same resource as it is a best resource. If a resource fails to execute the assigned job, then it is punished by adding less pheromone. Load balancing factor is included in order to check that no resource loads heavy at any time.

Ruay-Shiung Chang [12] aims at minimizing the completion time of jobs and also concentrated on load balancing. Pheromone matrix ($r \times j$) is constructed by matching r resources with n jobs by considering resource parameters such as bandwidth, CPU speed and current load and job parameters such as Job size, CPU time needed by the job. Jobs are assigned to corresponding resource which has higher pheromone value. Local update is done only to the resource to which a job is currently assigned in order to decrease the probability of choosing the same resource again by the next job and global update is done once the assigned job is completed by a resource where the entire pheromone matrix is recalculated based on current status of resources. The experimental results of this paper have shown that this BACO achieve good system load balance and also takes less time to execute jobs.

Hamed Shah-Hosseini [13] solved the optimization problem TSP using a new population based swarm intelligence approach Intelligent Water Drops (IWD). Artificial water drop is developed which imitates the properties of natural water drops and traverses the search space in all possible paths and converges to optimal solution (shortest path) but in more no of iterations.

Hamed Shah-Hosseini [14][15] applied the concepts of intelligent water drops and tries to solve multiple knapsack problem and n-queen problem which are NP-hard problems and obtained optimal or near-optimal solutions.

Yuhui Deng [16] performed resource discovery process in the hierarchical grid system where the optimal resource for the job is identified for the incoming job by matching job requirements with resources and the job is executed in the corresponding resource.

3. ANT COLONY OPTIMIZATION

The ant colony optimization algorithm (ACO) is a swarm-intelligence based heuristic algorithm for solving large computational problems like grid scheduling. Ants communicate via stigmergy (an indirect communication) where each real ant deposits a chemical substance called pheromone which increases the probability for other ants to follow the same path while exploring their environment. In case of ant algorithms, a form of artificial stigmergy is used in order to find the optimal solution. As ACO is capable to solve both static and dynamic problems, Ant Colony Algorithm is chosen to solve job scheduling problem here.

3.1 INTELLIGENT WATER DROPS

Intelligent Water Drops (IWD) is a nature inspired technique which is based on how the water drops in the rivers finds their ways to lakes, oceans or seas. IWD always prefers the path with low soils on its beds to path with high soils. Based on the path in which water drops move, the velocity changes i.e. amount of soil in the path is inversely proportional to the velocity. The water drops loads some amount of soil from the path it travels and deposits in the path where it travels with low velocity. Hence both deposition and removal of soil from the path being traversed takes place. All these techniques favours intelligent water drops to choose the shortest path from source to destination among many possible paths. The convergence speed of IWD is proved better compared to that of other existing swarm intelligence approaches as velocity changes over iterations based on the amount of soil in the path being traversed.

3.2 JOB SCHEDULING IN GRID USING IWD & ACO

The grid is composed of number of virtual organizations (VO). Each VO may have one or more MDS (Monitoring and Discovery System). The MDS's in VO are arranged in a hierarchical manner. Each MDS includes one or more grid nodes. The resource information of grid nodes within each MDS is maintained by the MDS. The resources within each MDS may be homogeneous or heterogeneous. The grid scheduler finds out the better resource of the incoming job and submits that job to the selected host. The grid scheduler does not have control over the resources and also on the submitted jobs. Any machine in grid can execute any job, but the execution time differs. The resources are dynamic in nature. As compared with the expected execution time, the actual time may be varied at the time of running the jobs to the allocated resource. The grid scheduler's aim is to allocate the jobs to the available nodes. The best match must be found from the list of available jobs to the list of available resources. The selection is based on the prediction of the computing power of the resource. So, lots of problems are needed to be solved in this area. The grid scheduler must allocate the jobs to the resources efficiently. The efficiency depends upon the criteria; one is make span and the other is load balancing among resources.

4. PROPOSED IWDACO ALGORITHM FOR JOB SCHEDULING IN GRID

This algorithm aims to minimize the computational time of each job that must be processed by available resources in grid computing system. The algorithm will select the resources by matching the job requirements with resources and IWD helps in finding the MDS containing matching job requirements and the routing information(shortest path) to reach the matching MDS's. For the matching nodes within matched MDS, ACO is applied to find out best resource for each incoming job.

There may be one or more jobs having same job requirements and thus matching with same MDS. In such case, preference is given to the job which has higher priority or the job

which arrived earlier. The representation of the grid system is a graph composing of MDS and grid nodes within each MDS.

IWD starts its travel from the source node and checks whether any MDS matches with job requirements and iterates from source to destination (matching MDS) to find out the best path from source to destination. IWD works as specified in the following steps.

1. Initialisation of static parameters:

For velocity updating, parameters are $a_v = 1$, $b_v = 0.01$, $c_v = 1$

For soil updating, parameters are $a_s = 1$, $b_s = 0.01$, $c_s = 1$.

InitSoil = 10000 and InitVel = 200.

Initialisation of dynamic parameters:

Visited node list of each IWD, $v_c(IWD) = \{ \}$. Initially it will be empty.

2. Repeat steps 2.1 to 2.4 for IWD's with partial solution.

2.1 From starting node, IWD chooses the next node to move based on probability,

$$p(i, j) = \frac{f(soil(i, j))}{\sum_{k \notin v_c(IWD)} f(soil(i, k))} \quad (1)$$

Such that,

$$f(soil(i, j)) = \frac{1}{\epsilon_s + g(soil(i, j))} \text{ and}$$

$$g(soil(i, j)) = \begin{cases} soil(i, j) & \text{if } \min(soil(i, j)) > 0 \\ soil(i, j) - \min(soil(i, j)) & \text{elsewhere } 1 \notin v_c(IWD) \end{cases}$$

2.2 Update the velocity of IWD moving from node i to j

$$vel(t+1) = vel(t) + \frac{a_v}{b_v + c_v \cdot soil^2(i, j)} \quad (2)$$

2.3 Compute the soil $\Delta soil(i, j)$ that IWD loads from the path it travels,

$$\Delta soil(i, j) = \frac{a_s}{b_s + c_s \cdot time^2(i, j; vel(t+1))} \quad (3)$$

Such that, $time(i, j; vel(t+1)) = \frac{HUD(j)}{vel(t+1)}$ where

HUD(j) is the distance from the current node to the next node.

2.4 Update the soil traversed by IWD using,

$$soil(i, j) = (1 - \rho_n) \cdot soil(i, j) - \rho_n \cdot \Delta soil(i, j) \quad (4)$$

$$soil^{IWD} = soil^{IWD} + \Delta soil(i, j)$$

3. Find the iteration best solution among all the solutions.

4. Update the soils on the paths in the iteration best by,

$$soil(i, j) = (1 + \rho_{IWD}) \cdot soil(i, j) - \rho_{IWD} \cdot \frac{1}{(N_{IB} - 1)} \cdot soil_{IB}^{IWD} \quad (5)$$

where, N_{IB} is the number of nodes in the solution.

5. Increment the iteration and go to step 2 until maximum number of iteration is reached.

After finding the matching MDS with job requirements and the routing information to the matching MDS, ACO is used to

find the optimal resource for the job among all matched resources. Here, an ant represents a job in the grid system. The Pheromone value of each matching resource with the job requirements indicates the capacity of each resource in grid system. Pheromone value will be determined by two types of pheromone update technique which are local pheromone update and global pheromone update. The initial pheromone value of each resource for each job is calculated based on the job parameters and resource parameters which is defined by,

$$PV_{ij} = \frac{\text{Computation Resource Demanded}}{\text{Computation Resource Available}} + \frac{\text{Storage Resource Demanded}}{\text{Storage Resource Available}} * \frac{1}{\text{load \%}} \quad (6)$$

The transition probability value of job with each resource is calculated using,

$$\frac{(\tau_{ij})^\alpha (\eta_{ij})^\beta}{\sum (\tau_{ij})^\alpha (\eta_{ij})^\beta} \quad (7)$$

where, η_{ij} is the soil value of the path computed by IWD. Then the pheromone value is updated by updated by,

$\tau_{ij} = [(1 - \rho) \text{ soilvalue}] - 10$ for edges in best solution

$\tau_{ij} = [(1 - \rho) \text{ soilvalue}]$ otherwise

5. EXPERIMENTAL RESULTS

The proposed IWDACO approach is simulated using gridsim toolkit. When a job is submitted in the grid environment, it is being processed by IWDACO scheduler and gives the result of optimal grid node id for the job within the MDS and the best routing information to it.

Table.1. Example - Incoming Job Information in the Grid Environment

Job Id	No of PE'S	MIPS	OS	Affordable cost	RAM (MB)
1	2	250	Linux	20	200
2	3	300	Win 7	18	300
3	4	400	Win vista	17	350
4	3	500	Win XP	16	400
5	4	270	Free DOS	9	550
6	1	330	Win 95	16	450
7	2	420	Win7	23	360
8	4	480	Win XP	20	500
9	3	530	Mac OS x	8	600
10	3	600	MS-DOS	7	550

Table.2. Example - Resource Information in the Grid Environment

Rid	Mid	No of PE's	MIPS	OS	Cost	RAM (MB)
1	1	2	280	Linux	12	250
1	2	3	300	Mac OS x	15	150

2	1	1	450	Win vista	16	200
2	2	2	200	MS-DOS	5	350
3	1	2	200	Win 95	18	250

Intelligent water drops traverses the graph which includes MDS's and performs resource matching [17] for each job with each resource in each MDS and returns the result as shown below in Table.3.

Table.3. Example - Resource Matching of each Job with each Resource in the Grid Environment

Rid	CPU	RAM	Match status
R1	0	0	Exact match
R2	-0.37	-0.11	Subsume match
R3	1	1	Plug-in match

R1, R2, R3 are the resources and the deviation matrix is calculated for each job with each resource within MDS. If the deviation value is exactly equal to zero, then it is said to be exact match. If deviation is greater than zero and less than are equal to one, then plug-in match. If deviation value is less than zero, then subsume match. If it is exact match or plug-in match, job can be submitted to that particular resource. If it is subsume match, job cannot be submitted to that particular resource as job requirements are greater than resource availability.

Finally, IWDACO scheduler returns the matching MDS id, Resource id, Machine id and the routing information to the resource for each job.

Table.4. Example-Schedule Information for each Job

Job id	MDS id	Rid	Mid	Path
1	5	3	1	1-2-3-5
2	6	2	2	1-2-3-5-6
3	7	2	3	1-2-3-5-7
4	4	1	1	1-2-4
5	5	2	3	1-2-3-5
6	6	1	2	1-2-3-6
7	7	1	2	1-2-3-5-7
8	8	2	3	1-2-3-5-7-8
9	9	1	3	1-2-3-5-7-9
10	5	1	2	1-2-3-5

Table.5 shows that the execution time shows better reduction in proposed method when compared to that of existing method. The Fig.1 shows the difference in execution time between the existing method and the proposed method.

Table.5. Comparison of Execution Time between existing ACO Technique and Proposed IWDACO Technique

S. No.	No. of tasks	No. of resources	Existing ACO method (Execution time in sec)	Proposed IWDACO method (Execution time in sec)
1	10	6	18	13
2	20	6	39	30

3	30	6	60	52
4	40	8	71	63
5	50	8	88	79

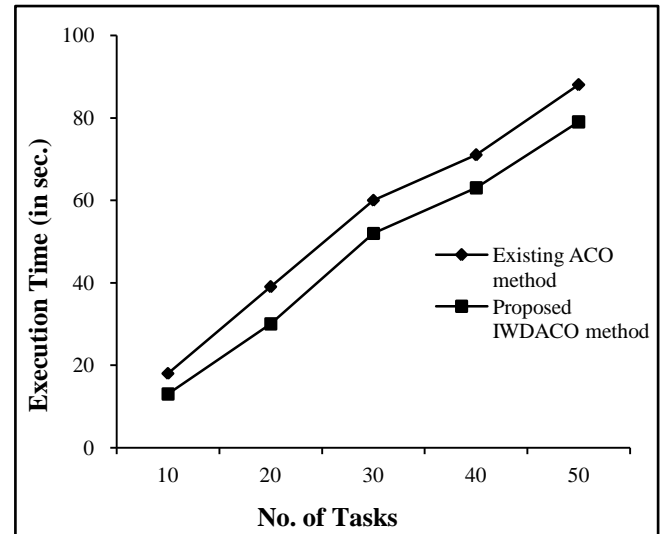


Fig.1. Graph showing comparison of execution time between proposed IWDACO technique and existing ACO techniques

This proposed technique will result in making scheduling decision faster, reduces the makespan of the executing jobs and improves load balancing of jobs among resources.

6. CONCLUSION

In this paper, the newly proposed hybridization technique using Intelligent water drops with Ant colony optimization approach is used to solve the scheduling problem in grid environment. This paper aims at finding out the optimal schedule decision for the job faster, minimizing the makespan of the executing jobs, maximizing the load balancing of jobs among resources. The experimental results evince the real potential of this proposed technique for job scheduling problem in grid environment as the job is executed in the best resource with minimum makespan compared to that of existing ACO techniques and also with improved load balancing of resources as jobs are executed in distributed manner.

REFERENCES

- [1] J. M. Schopf, "A General Architecture for Scheduling on the Grid", *Special Issue of the Journal of Parallel Distributed Computing on Grid Computing*, 2002.
- [2] D. Fernandez-Baca, "Allocating Modules to Processors in Distributed System", *IEEE Transactions on Software Engineering*, Vol. 15, No. 11, pp. 1427-1436, 1989.
- [3] M. Dorigo, G. Di Caro, "The Ant Colony Optimization meta-heuristic", *New Ideas in Optimization*, McGraw-Hill, pp. 11-32, 1999.
- [4] M. Dorigo, L. M. Gambardella, "Ant Colony System: A cooperative Learning Approach to the traveling salesman problem", *IEEE Transactions on Evolutionary Computation*, Vol. 1, No. 1, pp. 53-66, 1997.

- [5] S. Fidanova and M. Durchova, "Ant Algorithm for Grid Scheduling Problem", *Proceedings of the 5th International Conference on Large-Scale Scientific Computing*, pp. 405-412, 2006.
- [6] T. Stutzle, "MAX-MIN Ant System for Quadratic Assignment Problems", Technical Report, AIDA-97-04, Department of Computer Science, Darmstadt University of Technology, Germany, 1997.
- [7] B. Bullnheimer, R. F. Hartl and C. Strauss, "A new rank-based version of the ant system: A computational study", *Central European Journal for Operations Research and Economics*, Vol. 7, No. 1, pp. 25-38, 1997.
- [8] M. Dorigo, V. Maniezzo and A. Coloni, "Ant system: Optimization by a colony of cooperating agents", *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, Vol. 26, No. 1, pp. 29-41, 1996.
- [9] Yaohang Li, "A Bio-inspired Adaptive Job Scheduling Mechanism on a Computational Grid", *International Journal of Computer Science and Network Security*, Vol. 6, No. 3B, pp. 1-7, 2006.
- [10] Hui Yan, Xue-Qin Shen, Xing Li and Ming-Hui Wu, "An Improved Ant Algorithm For Job Scheduling In Grid Computing", *Proceedings of International conference on Machine Learning and Cybernetics*, Vol. 5, pp. 2957-2961, 2005.
- [11] Ruay-Shiung Chang, Jih-Sheng Chang and Po-Sheng Lin, "An ant algorithm for balanced job scheduling in grids", *Future Generation Computer Systems*, Vol. 25, No. 1, pp. 20-27, 2009.
- [12] K. R. Ku-Mahamud, and H. J. A. Nasir, "Ant Colony Algorithm for Job Scheduling in Grid Computing", *Proceedings of Fourth Asia International Conference on Mathematical/Analytical Modelling and Computer Simulation*, pp. 40-45, 2010.
- [13] Mohammad Reza Bonyadi, Mostafa Rahimi Azghadi and Hamed Shah-Hosseini, "Population-Based Optimization Algorithms for Solving the Travelling Salesman Problem", Travelling Salesman Problem, Intech, 2008.
- [14] Hamed Shah-Hosseini, "Intelligent water drops algorithm: A new optimization method for solving the multiple knapsack problem", *International Journal of Intelligent Computing and Cybernetics*, Vol. 1, No. 2, pp. 193-212, 2008.
- [15] Hamed Shah-Hosseini, "The intelligent water drops algorithm: A nature-inspired swarm-based optimization algorithm", *International Journal of Bio-Inspired Computation*, Vol. 1, Nos. 1/2, pp. 71-79, 2009.
- [16] Yuhui Deng, Frank Wang and Adrian Ciura, "Ant Colony optimization inspired resource discovery in P2P Grid systems", *The Journal of Supercomputing*, Vol. 49, No. 1, pp. 4 -21, 2009.
- [17] P. Balakrishnan, S. Thamarai Selvi and G. Rajesh Britto, "Service Level Agreement based Grid Scheduling", *Proceedings of IEEE International Conference on Web Services*, pp. 203-210, 2008.