VOL. 10, NO. 8, MAY 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

INTELLIGENT WATER DROP ALGORITHM POWERED BY TABU SEARCH TO ACHIEVE NEAR OPTIMAL SOLUTION FOR GRID SCHEDULING

D. Thilagavathi, M.C.A. and Antony Selvadoss Thanamani
Department of Computer Science, Nallamuthu Gounder Mahalingam College, Pollachi, Coimbatore, Tamilnadu, India
E-Mail: jthilagam76@gmail.com

ABSTRACT

Grid computing is a network of computer resources where every resources are shared, turning a computer network into a powerful super computers. In which, Grid Scheduling is a non linear multi-objective problem. In this paper, intelligent water drop algorithm is hybridized with Tabu Search algorithm to solve scheduling problem in computational grid. The proposed algorithm named EIWD-TS is a meta-heuristic algorithm based on swarm intelligence. The optimization objective of this research is to find the near optimal solution considering multiple objectives namely makespan, slowdown ratio, failure rate and resource utilization of grid scheduling. The result of the proposed model of this paper is tested with PSA (Parameter Sweep Application) dataset and the results are compared with Risky-MinMin (RMM), Preemptive-MinMin (PMM), Particle Swarm Optimization (PSO) and IWD. Experimental evaluation shows that the EIWD-TS algorithm has good convergence property and better in quality of solution than other algorithms reported in recent literature.

Keywords: intelligent water drop, tabu search, multi-objective, makespan, PSA.

1. INTRODUCTION

The Grid is emerging as a wide-scale, distributed computing infrastructure that promises to support resource sharing and coordinated problem solving in dynamic, multi institutional Virtual Organization [1]. The primary benefit of grid computing is the ability to coordinate and share distributed and heterogeneous resources such as the European Grid Infrastructure (EGI) which is a series of efforts to provide access to high-throughput computing resources across Europe using grid computing techniques [2].

Grids are usually dedicated for a single application. But they can also be used for other variety purposes. For this grids are constructed with middleware libraries. Management of resources both local and global becomes a great issue. One such issue is the scheduling of jobs to machines. Researchers have provided solutions to it from diverse perspective. Scheduling of jobs in grid environment aims to allocate resources optimally to a set of jobs while meeting multiple constraints and objectives. A Grid scheduler will be permanently running as follows: receive new incoming jobs, check for available resources, select the appropriate resources according to feasibility (job requirements to resources) and performance criteria and produce a planning of jobs (making decision about job ordering and priorities) to selected resources. In order to perform the scheduling process, the Grid scheduler has to follow a series of steps as in Figure-1.

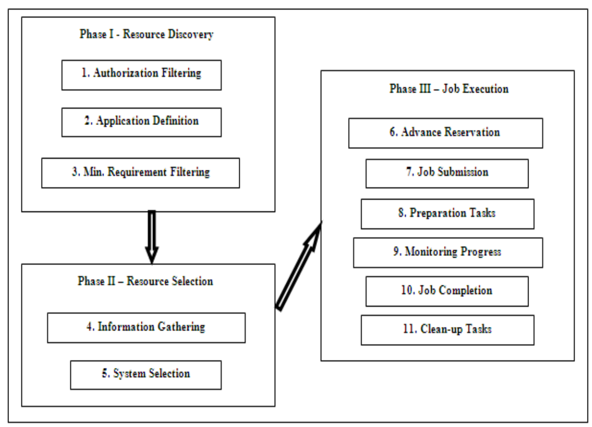

Figure-1. Grid scheduling process.

To solve job scheduling problem a number of heuristic optimization techniques like Simulated Annealing(SA), Genetic Algorithm(GA) and Tabu search have been presented. Recently, the new metaheuristic algorithm "Intelligent Water Drops," has been introduced in the literature and used for solving various problems like the traveling salesman problem (TSP), N-Queens problem, Multiple Knapsack problem, continuous optimization, etc. It is a population based constructive optimization algorithm which has been inspired from natural rivers and exploit the path finding strategies of rivers. Through a number of simulated experiments, we prove that the proposed IWD algorithm utilizes the available resources more efficiently and also reduces the makespan time.

The rest of the paper is organized as follows: Related works are discussed in Section 2. Section 3 introduces the IWD algorithm. The proposed EIWD-TS algorithm is explained in Section 4. Experiments settings and results

ARPN Journal of Engineering and Applied Sciences

are discussed in Section 5. Finally conclusions are presented in Section 6.

2. RELATED WORK

Ant Colony Optimization (ACO) is a heuristic algorithm with efficient local search for combinatorial problems. ACO imitates the behavior of real ant colonies in nature to search for food and to connect to each other by pheromone laid on paths travelled. Many researches use ACO to solve NP-hard problems such as travelling salesman problem, graph colouring problem, vehicle routing problem, and so on. Ruay-Shiung, Jih-Sheng, Po-Sheng Lin [3] suggests modified ant algorithm as Balanced ACO (BACO) algorithm which reduces makespan time and also tried to balance the entire system load. This work was implemented in the Taiwan UniGrid Platform. The BACO algorithm selects a resource for submitting the request (job) by finding the largest entry in the Pheromone Indicator (PI) matrix among the available jobs to be executed. This work was carried for independent jobs and not for workflow jobs.

Xhafa applied unstructured MAs and Xhafa, Alba, Dorronsoro and Duran [4] proposed Cellular MAs (structured MAs) for the independent scheduling problem under ETC model. Abraham, Chang, Liu, Zhang [5] proposed an approach for scheduling jobs on Computational Grids using fuzzy PSO algorithm.

Mohd Kamir Yusof and Muhamad Azahar Stapa [6], focuses on Tabu Search (TS) algorithm, a scheduling technique in grid computing was tested and evaluated on universal datasets using GridSim tool. The results indicate performance of tardiness is directly related to number of machines up to certain number of resources. The basic principle of TS is to pursue Local Search (LS) whenever it encounters a local optimum by allowing non-improving moves; cycling back to previously visited solutions is prevented by the use of memories, called Tabu lists, that record the recent history of the search, a key idea that can be linked to Artificial Intelligent concepts. TS deal with various techniques for making the search more effective. These include method for exploiting better information that becomes available during search and creating better starting points, as well as more powerful neighborhood operators and parallel search strategies.

Another important trend in TS is hybridization, i.e. using TS in conjunction with other solution approaches such as Genetic Algorithm, Lagrangean relaxation, Constraint Programming, column generation and integer programming technique.

PSO is a population-based search algorithm based on the simulation of the social behavior of bird flocking and fish schooling. Hu, Ouyang, Yang, Chen [7] proposes an Immune Particle Swarm Optimization (IPSO) algorithm. The basic idea of the IPSO is to record the particles with a higher fitness in the evaluating process, and make the new particles which satisfy neither the assumption nor the constraint condition replaced by the recorded ones. In addition, immune regulation should be done to maintain the species diversity while it decreases. This paper mainly discusses the independent task scheduling. Experiments show that the PSO algorithm has the best integrate performance. Whatever we consider the scheduling creating time, the makespan and the mean response time, it all has good performance.

Xhafa, Gonzalez, Dahal and Abraham [8] addressed the hybridization of GA and TS heuristics. Another hybrid approach for the problem is due to Ritchie and Levine [9, 10] who combine an ACO algorithm with a TS algorithm for the problem.

Shah-Hosseini [11] used IWD Algorithm to solve Travelling Salesman Problem (TSP) with the experiment of IWD algorithm using artificial and benchmark TSP environments that are resulting in fast convergence to optimum solutions. In a study conducted by Rayapudi [12], IWD Algorithm has been applied to solve economic load dispatch problems (ELDP). Using 6-unit and 20-unit thermal systems, the viability of IWD algorithm in solving ELD problem is demonstrated. The results obtained numerically disclosed the IWD algorithm that is converged to good solutions compared to the use of GA, PSO and BBO in which the standard deviation of IWD method is less than other approaches.

Niu, Ong and Nee [13] presented a promising optimization algorithm named MOJSS-IWD and applied to solve the multi objective job shop scheduling problem. Customized IWD algorithm identified the Pareto non-dominance schedules efficiently.

Thilagavathi and Antony Selvadoss Thanamani [14, 15] proposed a hybrid approach of IWD algorithm with a PSO algorithm for the job scheduling problem in Grid Computing. The proposed algorithm was evaluated and tested on NAS (Numerical Aerodynamic Simulation) data set showing promising results.

3. INTELLIGENT WATER DROP ALGORITHM

Intelligent Water Drops algorithm (IWD) [16] is a swarm based nature-inspired optimization algorithm, which has been inspired by the movement of natural water drops which flow in rivers, lakes, and seas. It is a population-based meta-heuristics where the IWDs construct a better solution through cooperation with each other. This algorithm can be applied to solve optimization problems. As pointed out by the authors, a stream can find an optimum path considering the conditions of its surroundings to reach its ultimate goal, which is often a lake or a sea. In the process of reaching for the destination, the water drops and the environment react with each other as the water drops move through the river bed. The water drops can change the environment (river beds) in which they are flowing; the environment can also influence the moving directions of the water drops. The gravitational force of the Earth powers the IWDs moving toward the destination. If there are no barriers or obstacles, the IWDs will move in a straight path to the destination. However, in the real scenario, as there are different types of obstacles when IWDs are forming their paths, the real path of the IWDs may be different from the ideal path. In a river path, many twists and turns can be observed. However, by considering the distance to the destination and the

VOL. 10, NO. 8, MAY 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

environmental constraints, the constructed path seems to be an optimal one.

In the original IWD algorithm, the IWDs are associated with two attributes namely:

- the amount of soil and
- the velocity of the IWDs.

The velocity enables the water drops to transfer soil from one place to another. Faster water drops can gather and transfer more soil from the river beds. Besides, the velocity of the IWDs is also affected by the path condition. The amount of soil in a path has an impact on the IWDs' soil collection and movement. A path with less soil allows the IWDs to move faster along that path, and the IWDs can attain a higher speed and collect more soil from that path, while a path with more soil is the opposite.

In the IWD algorithm, the movement of IWDs from source to destination is performed in discrete finite-length time steps. When an IWD moves from one location to the next, the increase in its velocity is proportional (nonlinearly) to the inverse of the soil of the path between the two locations, and soils of the IWDs increase because the IWDs remove some soil from the path. The soil increase is inversely proportional to the time needed for the IWDs passing between the two locations. The time duration to travel from one location to the second location depends on the distance between these two locations and the velocity of the IWDs. In the original IWD algorithm, the undesirability of a path is reflected by the amount of soil in the path. When an IWD has to choose a path among several candidate paths, it would prefer an easier path, i.e. a path with less soil than with more soil. The IWDs select a path based on a probabilistic function. The IWD algorithm uses a parameterized probabilistic model to construct solutions, and the value of the parameters is updated in order to increase the probability of constructing high-quality solutions.

4. ENHANCED INTELLIGENT WATER DROP ALGORITHM

In Algorithm EIWD-TS, N_{IWD_iter} gives number of iterations. Each iteration gives a feasible solution (schedule). The soils on the edges where the IWDs pass and velocities of the IWDs are updated during the travelling of the IWDs. After each iteration, the soils on elite IWDs are updated. Next, a group of best solutions are chosen. Combined local search is done using Tabu Search algorithm to improve the solution. After a local search, a best iteration solution identified and thereby global solution updated. After all the iterations, again local search for global best solution using Tabu Search algorithm is done to achieve a near optimal schedule.

Algorithm: EIWD-TS algorithm
Inputs: Resource details, Job requirements
Outputs: Makespan time, slow down ratio, failure rate, Grid utilization.

Method:

1. Submit jobs to the grid environment
2. Grid Resource Broker automatically submits the selected jobs to Scheduling Module
3. Scheduling Module calls EIWD-TS algorithm to get the best near optimal schedule in the pareto set.
4. Jobs are scheduled to grid resources by EIWD-TS algorithm.
5. If (Grid resources not available)

Job kept at the Grid Job Pool
Else
Job Processing is finished

6. If all the jobs are processed then statistical reports are generated automatically

EIWD-TS

1. Initialize a pareto set P.
2. for (each optimization objective) {initialize an IWDs group A;

initialize static and dynamic parameters
K:=0
while ($K<N_{IWD\_iter}$){
for (each time step t) {
for (each IWD ε A which feasible solution has not been discovered) {
Choose the next node for the IWD_g to visit in its scheduled candidate pool
updateVelocity of g^{th} IWD
computeDeltaSoil of g^{th} IWD
updateEdgeSoil of the edge(i,j)
update IWDSoil of g^{th} IWD }}
for (N_{elite}IWDs)
Update the soils of the edges in the current elite IWD Solutions
Set up a best solution group S_{BD}
pareto local search is done to evaluate the schedules using TabuSearch algorithm and initialized as local best solution S^{IB}
Update the soil of the path associated with S^{IB}
Dominance checking is done for the new schedule
Update the pareto set P
Update the global best solution S^{TB}
K=K+1}

3. Pareto local search is done on global best solution S^{TB} using Tabu Search algorithm and initialized as new S^{TB}
4. Update the pareto set P
5. Return the best near optimal schedule in the pareto set P

5. RESULTS AND DISCUSSIONS

Gridsim [17] provides a discrete-event framework for simulating core Grid entities such as jobs, resources, and information services. It is used to evaluate the performance of the proposed algorithm EIWD-TS. The recital of proposed algorithm is tested using PSA workload.

PSA Workload: The Parameter Sweep Application (PSA) model has emerged as a "killer application model" for composing high-throughput

VOL. 10, NO. 8, MAY 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

computing applications for processing on global Grids [18]. The parameter sweep application is defined as a set of independent sequential jobs (i.e., no job precedence). The independent jobs operate on different data sets. A range of scenarios and parameters to be explored are applied to the program input values to generate different data sets. The execution model essentially involves processing K independent jobs (each with the same task specification, but a different data set) on M distributed sites, where K is typically much larger than M (Table-1). We simulated the execution of a PSA workload of 10, 000 jobs over 20 Grid sites.

Table-1. Parameter settings of PSA workload.

Parameter	Value setting
Number of jobs N	10000
Number of sites M	20
Job arrival rate	0.008jobs/second/site
Job workloads	20 levels (0 - 300000)
Site processing speed	10 levels (0 - 10)
Job security demands (SD)	0.6 - 0.9 uniform distribution
Failure and Delay coefficients	$\lambda = 3; \gamma = 2$

The following section discuss about the metrics used to measure the performance and to compare with RMM, PMM, PSO and IWD.

Performance metrics: The performance metrics used for evaluation in this research work are discussed below:

Makespan: It is the aggregate execution time turnaround. From the Figure-2 it is obvious that the proposed EIWD-TS perform better than other methods.

Slowdown: It is the ratio of response time divided by runtime. From the Figure-3 it is shown that the proposed EIWD-TS achieve better performance.

Failure rate: It is the quantity of failed and rescheduled occupations. From the Figure-4 can be observed the failure rate is dropped down to remarkable level in the proposed EIWD-TS.

Grid utilization: Rate of transforming force allotted to effectively executed employments out of the aggregate preparing force accessible of a worldwide Grid. It is remarkable that the proposed EIWD-TS achieve better grid utilization which is showed in Figure-5.

Figures 2, 3, 4 and 5 shows the performance Analysis of Risky-MinMin (RMM), Preemptive-MinMin (PMM), Particle Swarm Optimization (PSO) and Intelligent Water Drop (IWD) and the proposed algorithm EIWD-TS in 10, 000 PSA jobs over 20 simulated Grid sites.

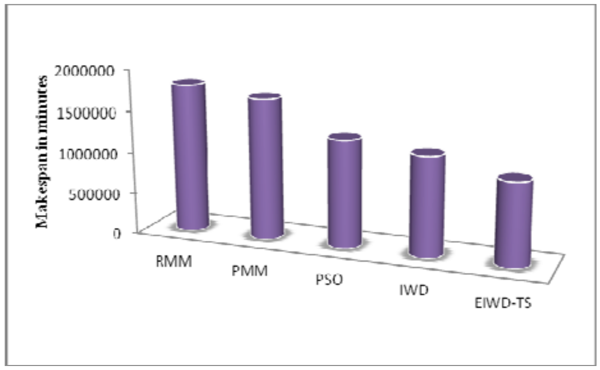

Figure-2. Makespan in PSA jobs.

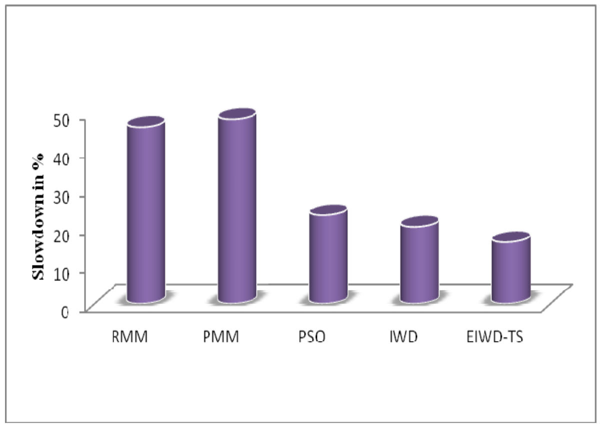

Figure-3. Slowdown in PSA jobs.

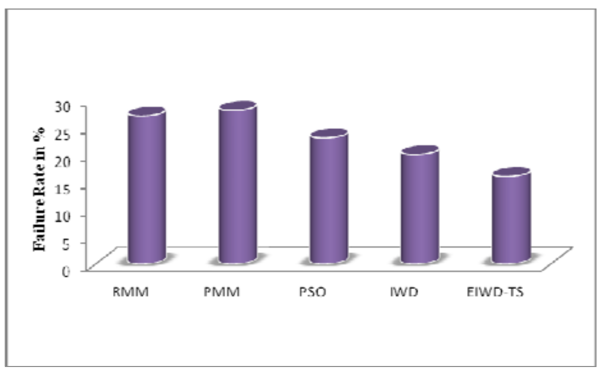

Figure-4. Failure rate in PSA jobs.

ARPN Journal of Engineering and Applied Sciences

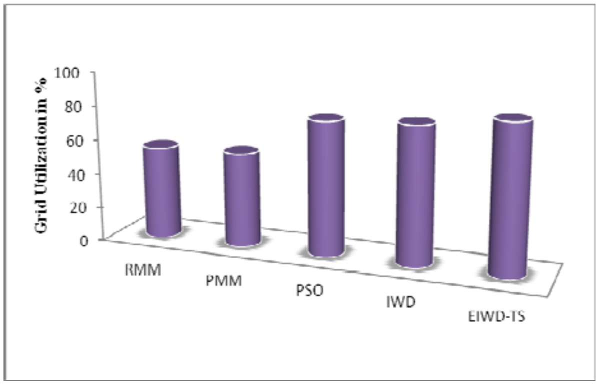

Figure-5. Grid utilization in PSA jobs.

6. CONCLUSIONS

In this paper, we have proposed EIWD-TS approach for job scheduling in grid environment. We have tested the proposed algorithm with PSA dataset. The experimental result shows that it is capable of achieving near optimal solution efficiently and effectively. It also outperforms the other algorithms namely RMM, PMM, PSO and IWD. Future scope of the research work can be handled with different optimization algorithm which can still increase the performance. Moreover the proposed work EIWD-TS can be implemented to different computing namely Cloud Computing and Super Computing.

REFERENCES

[1] I. Foster *et al*, "The anatomy of the grid: enabling scalable virtual organizations", Int. Journal of High Performance Computing applications, Vol. 15, No. 3, pp. 200-222, 2001.

[2] Christopher Lazou, "Grid infrastructure organization", European Parliament Magazine 265, Retrieved 2 October 2011, pp. 8-9.

[3] Ruay-Shiung Chang, Jih-Sheng Chang, Po-Sheng Lin, "An ant algorithm for balanced job scheduling in grids", Future Generation Computer Systems 25, 2009. pp. 20-27. (FGCS- Elsevier).

[4] F. Xhafa *et al*, "Efficient batch job scheduling in grids using cellular memetic algorithms", Journal of Mathematical Modelling and Algorithms (accepted, 2008) Published Online DOI: http://dx.doi.org/10.1007/s10852-008-9076-y.

[5] A. Abraham *et al*, "Scheduling jobs on computational grids using fuzzy particle swarm algorithm", Proc. 2006 10th Int. Conf. on Knowledge-Based and Intelligent Information and Engineering Systems. England. pp. 500-507.

[6] Mohd Kamir Yusof and Muhamad Azahar Stapa, "Achieving of tabu search algorithm for scheduling technique in grid computing using gridsim simulation tool: Multiple jobs on limited resource", Int. Journal of Grid and Distributed Computing. Vol. 3, No. 4, December, 2010

[7] Hu Xu-Huai *et al*, "An IPSO algorithm for grid task scheduling based on satisfaction rate", Int. Conf. on Intelligent Human-Machine Systems and Cybernetics, 2009.

[8] F. Xhafa *et al*, "A GA (TS) hybrid algorithm for scheduling in computational grids", Springer-Verlag Berlin Heidelberg 2009, HAIS 2009, LNAI 5572, 2009, pp. 285-292.

[9] G. Ritchie, "Static multi-processor scheduling with ant colony optimisation and local search", M.S. thesis, School of Informatics, Univ. of Edinburgh, 2003

[10] G. Ritchie and J. Levine, "A hybrid ant algorithm for scheduling independent jobs in heterogeneous computing environments", in 23rd Workshop of the UK Planning and Scheduling Special Interest Group (PLANSIG 2004), 2004.

[11] H. Shah-Hosseini, "Optimization with the nature-inspired intelligent water drops algorithm", Int. Journal of Intelligent Computing and Cybernetics, Vol. 1, No. 2, 2008, pp. 193-212.

[12] S. R. Rayapudi, "An intelligent water drop algorithm for solving economic load dispatch problem", Int. Journal of Electrical and Electron Eng. 5, 2011, pp. 43-49.

[13] S.H. Niu *et al*, "An improved intelligent water drops algorithm for solving multi-objective job shop scheduling", Eng. Appl. Artif. Intel., 2013, http://dx.doi.org//1.1016/j.engappai.2013.07.011

[14] Thilagavathi and Antony Selvadoss Thanamani, "Intelligent Water Drop Algorithm based Particle Swarm Optimization (IWDPSO) towards multi objective job Scheduling for grid computing", Research Journal of Applied Sciences, Engineering and Technology, accepted December 2014, in press.

[15] Thilagavathi and Antony Selvadoss Thanamani, "An improved intelligent water drop algorithm for job scheduling in grid computing", Int. Journal of Applied Engineering Research (IJAER), unpublished.

[16] H. Shah-Hosseini, "Problem solving by intelligent water drops", In: IEEE Congress on Evolutionary Computation. Singapore, 2007. pp. 3226-3231.

[17] R. Buyya and M. Murshed. "Gridsim: A toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing",

ARPN Journal of Engineering and Applied Sciences

Concurrency and Computation: Practice and Experience (CCPE), 14, 2002.

[18] H. Casanova, A. Legrand, D. Zagorodnov and F. Berman, "Heuristics for Scheduling Parameter Sweep Applications in Grid Environments", Proc. Heterogeneous Computing Workshop (HCW '00), pp. 349-363, May 2000.