IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 11, November 2016

Comparative Analysis on Efficient Heuristics Based Algorithms in Grid Computing

Avriti[1], Er. Ranbir Singh[2]

Bhai Maha Singh College of Engg, Muktsar, Punjab[1, 2]

Abstract: The grid computing pacts controlling and active structure, with the several resources, distributed CPU loads, beside with the extent of futile memory continually varying. This paper shows virtual study of many familiar grid computing built scheduling methods by pleasing into account the completing environment as well as metrics used similar relaxed computing, environment besides metrics used through the goal to determine the proficiency of every existing raising procedure. This study discovered that ACO has properly weighty results over the presented techniques.

Keywords: Grid Computing, Resource Discovery, Heuristics Based Algorithms.

1. INTRODUCTION

Grid Computing is a conception that enables the effective consumption of the grid resources. A Grid System is designed using many dissimilar or similar resources to pact with big-scale scientific complications [1].

Grid can attain the similar level of calculating power as a workstation does, however at considerable reduced cost.

A vital service in the Grid computing is source discovery that finds the suitable resources for demanded task to tie the user's application requirements.

Source discovery methods in Grid computing usually are distributed into five chief categories:

Centralized, Distributed, Peer to Peer, Classified and Agent based methodologies [2] [3].

1.1 Phases of scheduling: There are three chief phases of scheduling-

Resource Discovery: This is the first phase in which resource detection grid scheduler's demeanor a global search to create a list of all open resources as well as their restrictions and history contours in a system [4].

Matchmaking: In the second segment, matchmaking schedulers attempt to determine preeminent choices for performing jobs and copy data files.

Capabilities of computational nodes/storage nodes as well as worth of the network joining them are among the simple characteristics that need to be deliberated by schedulers to accomplish this phase [4].

Job execution: It is the last stage where job completing schedulers produce instructions for computational nodes and storage nodes to execute jobs and imitate data files [4].

2. JOB SCHEDULING TECHNIQUES IN GRID

2.1 Artificial bee Colony Optimizaton

The ABC is an procedure motivated by the smart behavior showed by honeybees when examining for food. In ABC, the group of artificial bees comprises three groups of bees: working bees, spectator bees and detective bees.

For each food source (FS) there is only one working bee. A fitness task is used to allocate a quality or 'nectar' value to the food sources.

Each working bee hunts for a new food source within its particular neighborhood and travels to it if it has a higher nectar worth. Each spectator bee then picks one of the working bee food sources probabilistically in a manner similar to roulette maneuver selection [5].

ABC has been used in several applications in numerous different fields. One of the most motivating application areas is exercise neural networks [6].

$$\text{Dance floor} = \cup_{i=0}^{Ncn} \{B1, B2, \ldots\ldots Bm\}$$

$$M_i = CN_i^{spd} * CN_i^{prcs}$$

Each bee guesses a profit for succeeding a dancing bee on dance floor and thus calculated as

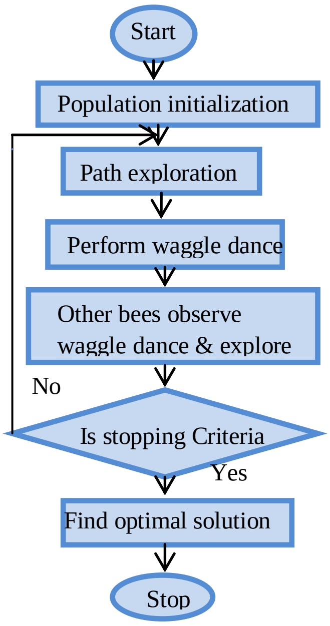
$$B(J_i, CN_j) = MAX\{ B_{Scout}(j_i, CN_j), B_{Follower}(j_i)\}$$

Where B is Benefit

$$B_{Scout}(j_i, CN_j) = CN_j^{FrPrcs} - \frac{J}{CN}$$

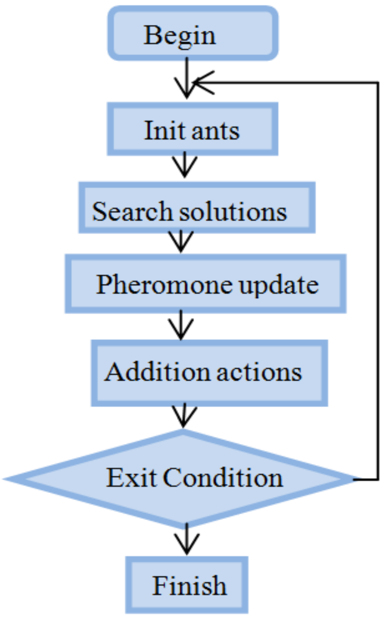
And

$$B_{Follower}(j_i) = MAX\{ \bigcup_{B_w \in DanceFloor} Sim(J_i, B_w) * B_{yy}^{Benefit}$$



Flowchart 1: Block diagram of ABC

2.2 Ant Colony Optimization: An ant colony optimization (ACO) method is evaluated in solving scheduling snags in a dynamic job shop environment. The most shared approach is to allocate one transmitting rule for an entire, generally linear, system [7]. ACO Algorithm is centered on the Ant algorithm and adapted it to suit the grid environment. An inspiring factor, a punish factor and a load harmonizing factor are also added into the algorithm [1].



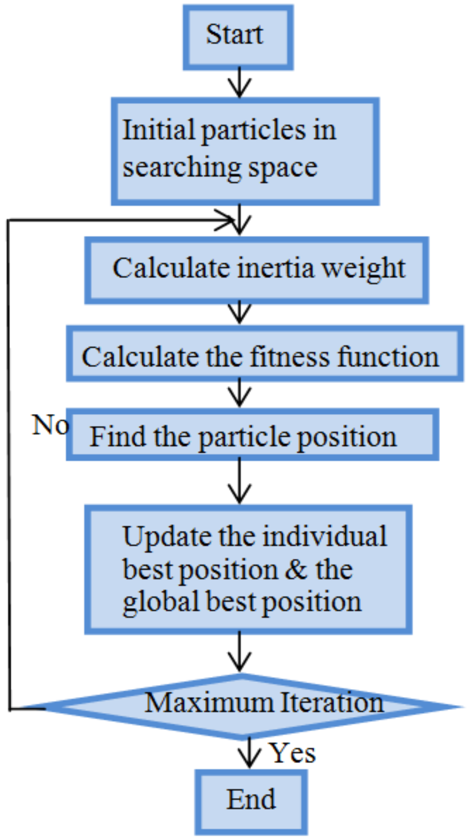
Flowchart 2: Block Diagram of ACO

$$PR = \pi_{ij}(t)_{new} \leftarrow \rho.\tau_{ij}(t)_{old} + \Delta\tau_{ij}(t) \quad (1)$$

$$MP = \pi_{ij}(t)_{new} = \left(\rho.\tau_{ij}(t)_{old}\right) + \left(\frac{\rho}{\rho+1}\right).\Delta\tau_{ij}(t)). \quad (2)$$

Here, PR is Pheromone Rule, MP is Modified Pheromone, j is a job, i is the machine assigned to the job j.

2.3 Particle Swarm Optimization In a PSO method, a swarm of individuals (called particles) fly through the examine space. Each particle signifies a candidate clarification to the optimization problem [8]. PSO is adjusted with a bunch of random particles (solutions) then examines for targets by updating compeers [9].PSO is a self adaptive global search based optimization technique. It has become widespread due to its straightforwardness and its effectiveness in comprehensive range of application with low computational cost. We will give a brief description of PSO algorithm [19].



Flowchart 3: Block Diagram of PSO

$$V_i^{k+1} = wv_i^k + C_1 rand_1 * \left(pbest_i - x_i^k\right) + c_2 rand_2 * (gbest - x_i^k),$$
$$X_i^{K+1} = X_i^k + V_i^{k+1}$$

Where:

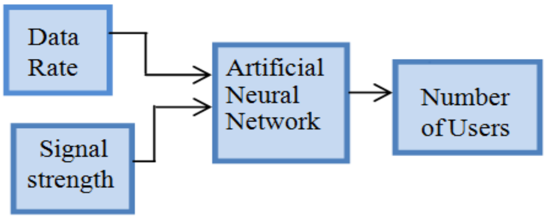
v_i^k particle i of velocity
v_i^{k+1} velocity at iteration k+1
w inertia weight
c_j acceleration with coefficients: j = 1,2
$rand_i$ arbitrary number between 0 and 1
x_i^k presentpoint of particle i at iteration k
$pbest_i$ finest position of particle i
gbest position of finest particle in a population
x_i^{k+1} position of i at iteration k+1

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 11, November 2016

2.4 Artificial Neural Network: The neurons function coordinately to solve the difficulties and transfer the evidence using synapses. In fact synaptic linkage between neurons is recycled to store the gained knowledge [14]. We calculates: [3]

$\text{Completion}[i] = \text{ready}_i + \sum_{j \in N_i} \text{ETC}[j][i]$ $\text{Makespan} = \max(i \in m)\ \text{completion}[i]$

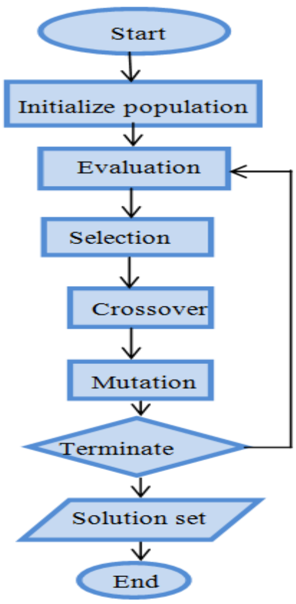
$$\text{Flowtime}[i] = \text{ready}_i \sum_{j \in \text{sort } [i] s[j=i]} \text{ETC}[j][i]$$



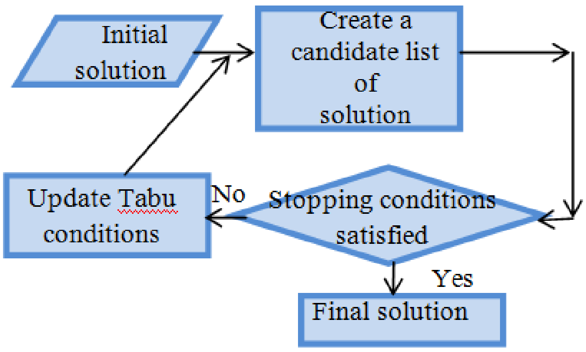
Flowchart 4: Block Diagram of Neural Network

2.5 Genetic Algorithm: Genetic algorithms are pursuit algorithms based on the scheme of natural selection, well known by the evolutionary procedure. In the method of evolution, the individual is constantly changing genetic factor to acclimate to the environment of his life. Phases of genetic algorithms are 1) The formation of the initial population (chromosome), 2) The computation of fitness values, 3) Selection, 4) Restoration (parent, crossover and mutation), 5) Formation of new populace regenerated [10]. GA will escape from the native best to search for the worldwide best [9].

2.6 Tabu Search: This manner alone has been used in several fields, comprising resource planning, broad castings, scheduling, logistics, space planning and so on. TS normally do not use the creation of candidate solutions at chance [10]. We use the solution produced from greedy algorithm as the novel solution for Tabu search [11] [15].

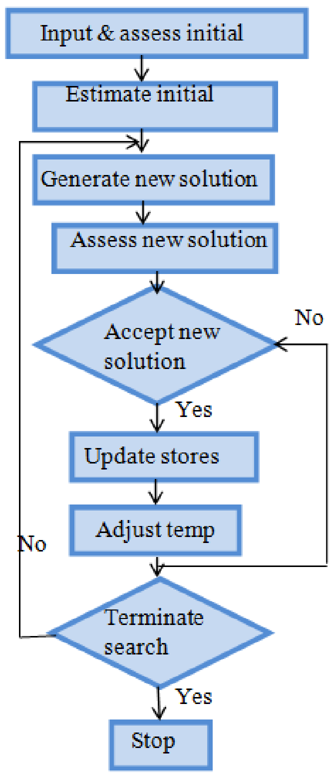


Flowchart 5: Block Diagram of Genetic Algorithm



Flowchart 6: Block Diagram of Tabu Search

2.7 Simulated Annealing: The precise characteristic and asset of the SA it that it retains exploring worse resolutions throughout the entire optimization process. This means that it does not goal at systematically declining the value of a given multivariable purpose to find its absolute lowest. This conduct helps avoiding being fixed in local minima, which is an especially valuable property when a large search space, supposedly full of local minima, has to be discovered [12] [13].



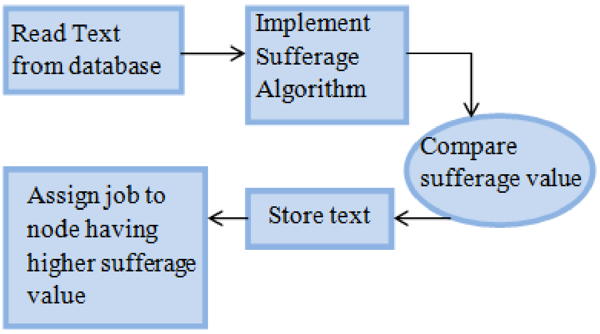
Flowchart 7: Simulated Annealing

2.8 Sufferage Algorithm: In this algorithm, every job is assigned allowing to its sufferage value. The sufferage value is demarcated as the difference between its second earliest accomplishment time and its earliest accomplishment time. [8]. Suffrage is that a job should be appointed to a assured resource and if it doesn't execute that resource, it will suffer the dominant [9].

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 11, November 2016



Flowchart 8: Sufferage Algorithm

3. CONCLUSION

Grid computing is a system that permits us to connect to network resources and amenities and create a large dominant system that has the capability to perform very difficult operations that a sole computer cannot achieve. To employ the microgrid functionalities efficiently and acquire the maximum benefit from such a arrangement, it is vital to manage the dispersed generation and depletion by using various operative heuristics algorithms.

REFERENCES

[1] Lee, Yun-Han, Seiven Leu, and Ruay-Shiung Chang. "Improving job scheduling algorithms in a grid environment." Future generation computer systems 27.8 (2011): 991-998.
[2] Souri, Alireza, and Nima Jafari Navimipour. "Behavioral modeling and formal verification of a resource discovery approach in Grid computing." Expert Systems with Applications 41.8 (2014): 3831-3849.
[3] Chang, Ruay-Shiung, Chih-Yuan Lin, and Chun-Fu Lin. "An adaptive scoring job scheduling algorithm for grid computing." Information Sciences 207 (2012): 79-89.
[4] Taheri, Javid, et al. "Hopfield neural network for simultaneous job scheduling and data replication in grids." Future Generation Computer Systems 29.8 (2013): 1885-1900.
[5] Kim, Sung-Soo, et al. "Optimal job scheduling in grid computing using efficient binary artificial bee colony optimization." Soft Computing 17.5 (2013): 867-882.
[6] Karaboga, Dervis, et al. "A comprehensive survey: artificial bee colony (ABC) algorithm and applications." Artificial Intelligence Review 42.1 (2014): 21-57.
[7] Korytkowski, Przemysław, Szymon Rymaszewski, and Tomasz Wiśniewski. "Ant colony optimization for job shop scheduling using multi-attribute dispatching rules." The International Journal of Advanced Manufacturing Technology 67.1-4 (2013): 231-241.
[8] Pooranian, Zahra, et al. "An efficient meta-heuristic algorithm for grid computing." Journal of Combinatorial Optimization 30.3 (2015): 413-434.
[9] Dev, S. Gokul, and R. Lalith Kumar. "User Deadline Based Job Scheduling in Grid Computing." International Journal of Computer Science and Network Security (IJCSNS) 15.3 (2015): 62.
[10] Darmawan, Irfan, Yoga Priyana, and Ian Joseph. "Grid computing process improvement through computing resource scheduling using genetic algorithm and Tabu Search integration." Telecommunication Systems, Services, and Applications (TSSA), 2012 7th International Conference on. IEEE, 2012.
[11] Ravula, Anusha, and Byrav Ramamurthy. "A tabu search approach for joint scheduling of resources in a lambda grid network." Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE. IEEE, 201
[12] Velik, Rosemarie, and Pascal Nicolay. "Energy management in storage-augmented, grid-connected prosumer buildings and neighborhoods using a modified simulated annealing optimization." Computers & Operations Research 66 (2016): 248-257.
[13] Dai, Min, et al. "Energy-efficient scheduling for a flexible flow shop using an improved genetic-simulated annealing algorithm." Robotics and Computer-Integrated Manufacturing 29.5 (2013): 418-429.
[14] Ansari, Abdollah, and Azuraliza Abu Bakar. "A Comparative Study of Three Artificial Intelligence Techniques: Genetic Algorithm, Neural Network, and Fuzzy Logic, on Scheduling Problem." Artificial Intelligence with Applications in Engineering and Technology (ICAIET), 2014 4th International Conference on. IEEE, 2014.
[15] Yi, Pan, Hui Ding, and Byrav Ramamurthy. "A tabu search based heuristic for optimized joint resource allocation and task scheduling in grid/clouds."Advanced Networks and Telecommuncations Systems (ANTS), 2013 IEEE International Conference on. IEEE, 2013.

 DOI 10.17148/IJARCCE.2016.51121