Middle-East Journal of Scientific Research 23 (10): 2586-2591, 2015
ISSN 1990-9233

DOI: 10.5829/idosi.mejsr.2015.23.10.22788

A Novel Approach for Multiple Object Resource Allocation Using Hybrid Algorithm

[1]A.M. Senthil Kumar, [2]M. Venkatesan and [3]A. Rajivkannan

[1]Dept. of CSE, Tejaa Shakthi Institute of Technology for Women, Coimbatore, Tamilnadu, India
[2]KSR Institute for Engg & Tech, Tiruchengode, Tamilnadu, India
[3]Dept. of CSE, KSR College of Engineering, Tiruchengode, Tamilnadu, India

Abstract: One of the main aspect in cloud computing is the problem of resource allocation. The Multiple Objects Resource Allocation (MORA) is implemented to optimize the allocation of nodes and its activity with efficiency and the objectives square measure is also optimized at the same time. The MORA has to acquire the minimum possible resolution in cloud computing. During the implementation of this technique, we should minimize varied parameters like minimum value, operate best value and the total completion time. The foremost necessary feature and objective of MORA is to get the most utilization of the recourses at minimum value and it also optimizes the simplest job sequence with total completion time from supply to destination. The planned hybrid genetic rule with Particle Swarm Optimization (GA-PSO) rule incessantly changes PSO parameters in line with analysis state. The designed hybrid Algorithm is bigger than existing optimization approaches like Bestowed Genetic Rule (GA), Particle Swarm Optimization (PSO) and Hymenopterans insect Colony Optimization (ACO). The hybrid approach acquires the minimum possible solutions with minimum value, less total project length and with minimum resources allocation moments. The simulation results, square measure is obtained by achieving the spare minimum optimization value and best position with minimum completion time.

Key words: GA · PSO · Resource scheduling · Optimization · Cloud computing

INTRODUCTION

A new approach for cloud computing is hybrid architecture and it describes the demand of computing services in cloud computing. This architecture consists of a larger infrastructure that is used to access the data or application available anywhere around the globe [1-3]. The cloud computing service providers consider storage, software, other applications to provide and authenticate the services to help any organization [4, 6]. In recent days the exponential growth of cloud computing, is been widely adopted by the industry and thus making a rapid expansion in availability of resource in the Internet [5]. The increasing size of the service providers in a cloud system require a massive requests, since the major challenge is to keep the performance and provide the best services whenever such an outburst occurs [7-9].

A number of Elementary Algorithms (EA) is used for task scheduling in cloud environment like Genetic Algorithm (GA), Simulated Annealing (SA), Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO)[10-11], out of which genetic algorithm (GA) has been proved quite effective. Genetic algorithm is applicable in different domains like robotics, machine learning, circuit designing, time-series forecasting, auto sensing, signal processing, game playing etc [12, 13]. The main advantage of GA is that it has great ability to handle multi objective problems and can be easily interfaced with existing simulations and models. Since it can explore big search space it does not impose any fixed length of the solution for a particular problem. Moreover, Genetic algorithm is very easy to understand and it does not require too much knowledge of mathematics.

The main aspect of Multiple-Objective Resource Allocation (MORA) allocates the number of resources to a number of industries or destination is optimized with feasible time, cost and position. It can be applied for various applications in industries, distribution, health care resource allocation, etc. MORA is a one of the very efficient method to allocate the resources with efficient manner in cloud computing [14-17]. Usually the need for cloud computing to allocate constrains resource among

Corresponding Author: A.M. Senthil Kumar, Department of CSE, Tejaa Shakthi Institute of Technology for Women, Coimbatore, Tamilnadu, India.

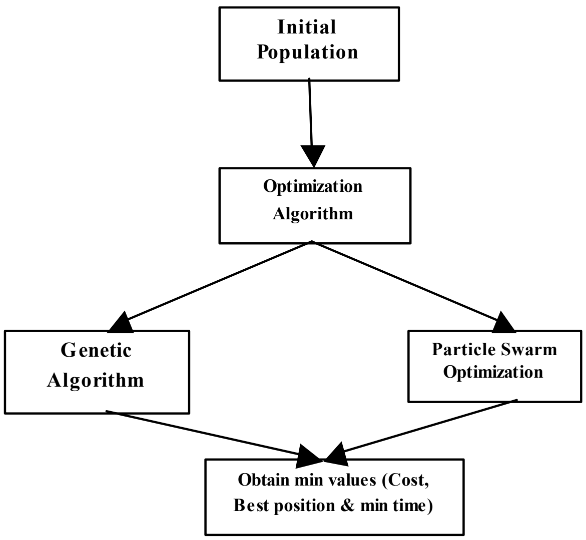

Fig. 1: Basic Architecture for Resource Allocation

activities for optimizing the objectives. In real time situation the resource allocation (RA) is used several formulations with different problem scenarios [15]. Single Objective RA (SORA) seeks to optimize a single goal, such as benefit maximization or cost minimization. Multiple-Objective RA (MORA) optimizes a set of goals which may involve cost, position and time to be minimized. The resource allocated can be either discrete or continuous and the amount of resource units to be allocated to an activity may be constrained in a specified range. In the branch-and-bound tree the branch is cut-off when the cost of the partial trial solution is worse than the current bound. The Dynamic Programming (DP) algorithm with an efficient pruning heuristic is used for managing the range optimization for solving MORA [16].

- The horizon of waiting upon non-dominated accustomed the Pareto-optimal order so that the obligation can be minimized;
- In most cases, uniform distribution of the solutions is found attractive. The afflict of this pattern is strengthened and it can be supported by a unconditional background metric and
- The mass of the derived non-dominated accomplishment demand can be maximized, i.e., for a constant objective, a round parade-ground of plan can be changed by the non-dominated solutions.

Since, the Algorithms have bigger capabilities in optimizing advanced issues with widespread solutions. We have to apply EAs to construction issues. The advantage in exploitation optimization algorithm is to resolve multi-objective improvement issues into attributable to its capability by looking at the collection of doable solutions to seek out the best Pareto front with a fewest runs of formula. Moreover, EAs square measure is at less risk in the continuity of the Pareto front. 3 recent important in EAs are: Genetic formula (GA), Particle Swarm improvement (PSO), Ant Colony improvement (ACO).

In this paper, the comprehensive comparison approach of multi-objective optimization algorithms to minimize Time, Cost and Best position in Resource Optimization problems are presented. Section 2 proposes that the GA algorithm and PSO algorithm that can be used for load balancing. Section 3 presents the proposed hybrid GA-PSO algorithm. Section 4 contains the simulation results and finally Section 5 concludes the work.

Problem Formulation

Genetic Algorithm: Genetic algorithmic rule that comes from biological science and Darwin trade is the widely used methodology. It will be used for combinatorial optimization, pattern recognition, machine learning and coming up with strategy and data process. This algorithmic rule has several benefits such as, doing a world search quickly; identifying an easy method to utilize the analysis operations to search randomness iterated by chance mechanism and well associatively to mix with alternative optimization techniques.

However, It cannot be utilized to make an in depth use of the feedback data from the system and would build the city of search meantime. Once the finding progress reaches a particular range a lot of redundancy iterations square measures are required. Then the speed of the convergence to the optimum resolution will drop quickly. That would result in solving the optimum resolution inefficiently.

GA for Load Balancing in Cloud Computing: Though Cloud computing is dynamic at any specific instance the aforementioned drawback of load equalization will be formulated as, allocating N variety of jobs submitted by cloud users to M variety of process units within the Cloud. Every process unit can have a Process Unit Vector (PUV) indicating current standing of process unit utilization. This vector consists of Number of Nodes (NoN), indicating the number directions that can be dealt by the service provider, α, price of execution of instruction and delay price L. The delay price is (AN) which is a estimate of penalty that Cloud service supplier has to pay to client within the event of job finish over the point in time source by the service supplier.

$$SUV = f(NoN, \alpha, L) \quad (1)$$

Similarly each job submitted by cloud user can be represented by a Source Unit Vector (*SUV*). Thus the attribute of destination unit vector can be represented by,

$$DUV = f(t, NIC, AT, wc) \quad (2)$$

where, t represents the kind of service needed by the work, Software System as a Service (SAAS), Infrastructure as a Service (IAAS) and Platform-as-a-Service (PAAS). NIC represents the amount of directions given within the job, this is often a count of instruction within the job determined by the processor. Job point in time (AT) indicate clock time of arrival of job within the system and worst case completion time (WC) that is the minimum time needed to complete the work by a process unit. The Cloud service supplier has to allot these N jobs among M variety of processors. The specified value operations ζ as indicated in equation three is decreased.

$$\zeta = w1 * \alpha(DUV \div SUV) + w2 * L \quad (3)$$

where, w1 and w2 square measures are predefined weights. It is troublesome to decide/optimize the weight because one criterion might be is that when larger weight is to be distributed, the network would become slower or down. The logic here is that the user preference or importance is given to a specific issue over the opposite. Here the later approach has been used and also the optimization is then performed on the given set of weights. The weights square measure thought-about as w1 = zero.8 and w2 = zero.2 such that summation is one. Thus the load equalization drawback square measure is complicated and may be thought-about as computationally complex. Such a loop cannot be developed by applied mathematics thus it is difficult to search out the globally best resolution by mistreatment settled polynomial time algorithms or rules. GAs [3] square measure is the foremost and widely used artificial intelligent techniques primarily for effective search and optimization. It is a random looking out algorithmic program supported by the mechanisms of action and biology. GAs has been proved to be very economical and stable in looking for world optimum solutions, especially in complicated and or immense search house. In this paper, GA has been planned as a load equalization technique for cloud computing to search out world optimum processors for a job during a cloud. The arrival of job as linear and rescheduling of jobs is not considered because the resolution is world optimum in nature.

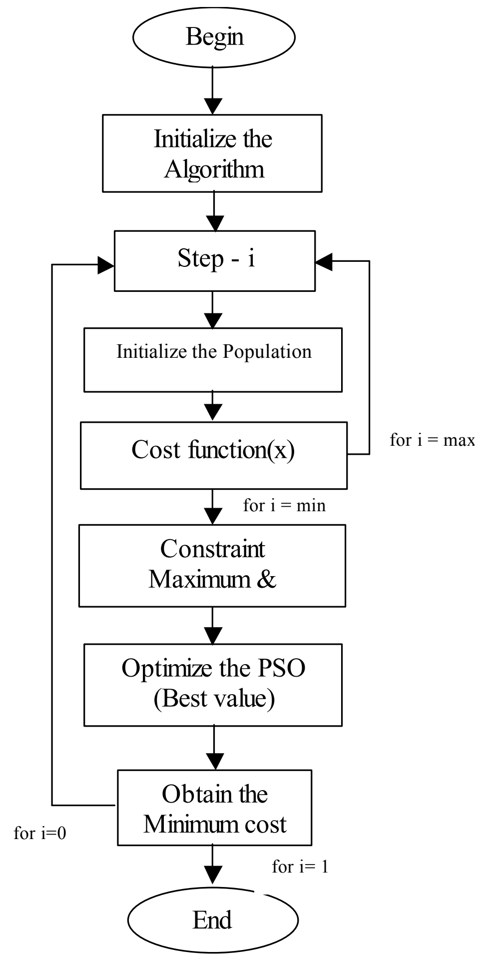

Fig. 2: The Rule for Genetic algorithm used for resource allocation.

Proposed Algorithm: A simple GA consists of 3 operations: Choice, Genetic Operation and Replacement. The advantage of this technique is that it will handle a colossal search house which is applicable to advanced objective operation and may avoid being adventure into native optimum resolution. The rule of GA is used for the load reconciliation in Cloud computing and it is shown in Figure 2 and details of GA area unit are as follows.

Initial Population Generation: GA works on fastened bit string illustration of individual resolution. So, all the potential resolutions within the solution house area unit is encoded into binary strings. From this associate of initial population of 10, 10 several chromosomes area unit is designated haphazardly.

Crossover: The target of this step is to pick most of the days and the most effective fitted chromosome that will be tried by people for crossover. The fitness worth of every individual body is calculated and the utilization of the fitness operates as given in.

This pool of chromosomes undergoes a random single purpose crossover. Whenever relying upon the crossover purpose, the portion lying on one aspect of crossover website is changed with the opposite aspect, so it generates a brand new type of individuals.

Mutation: Currently a really little worth (0.05) is picked up as mutation chance. Relying upon the mutation value, the bits of the chromosomes and area units are toggled from one to zero or zero to one. The output of this is often a brand new pairing pool ready for crossover. This GA method is recurrent until the fittest body (optimal solution) is found or the termination condition (maximum range of iteration) is exceeded.

The proposed algorithm is as given below:

Step 1: Initialize a population

Step 2: Initialize the random variable.

Step 3: Set the Iteration and parameters

Step 4: Evaluate the fitness value of each population.

Step 5: Find the Initial Population cost

Step 6: To find out maximum or minimum number of iteration to identity whether the solution is exceeded or optimized.

Step 6(a): Consider the chromosome with lowest fitness twice and eliminate the chromosome with highest fitness value to construct the mating pool.

Step 6(b): Perform single point crossover by randomly selecting the crossover point to form new offspring

Step 6(c): Mutate new offspring with a mutation probability of (0.05) and new population and use this population for next round of iteration.

Step 6(d): Test for the end condition.

RESULTS AND DISCUSSIONS

In resource allocation problem, the hybrid genetic algorithm is such as Particle Swarm Optimization (GA-PSO) is used to obtain the best values such as cost,

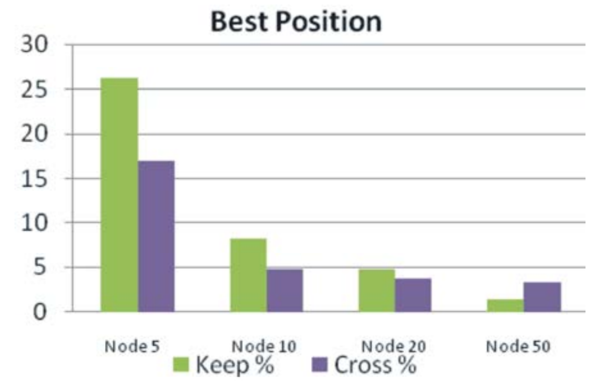

Fig. 3: Comparison of Best Position.

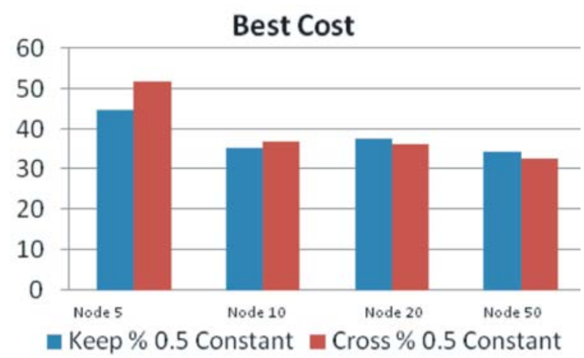

Fig. 4: Comparison of Best Cost Value.

Table 1: Best cost and Position to obtain the Keep percentage is 0.9

No of Iteration	pop size	Best Cost	Best Position	Time
100	5	44.7878	26.199	7.871978
100	10	35.2665	8.1954	7.902401
100	20	37.5725	4.7237	7.8423076
100	50	34.4549	1.40568	7.9041924

Table 2: Best cost and Position to obtain the Crossover percentage is 0.9

No of Iteration	pop size	Best Cost	Best Position	Time
100	5	51.74808	17.02698	7.8965458
100	10	37.10024	4.83382	7.920454
100	20	36.24924	3.79142	7.8930798
100	50	32.6863	3.40274	7.8967516

position and the overall computational time for resource allocation shown below in the Table 1 and Table 2. In Table 1 & 2 the numbers of nodes are varied to obtain the best cost, position and computing time with the same iteration.

Figure 3 shows the comparison between the best position in keep percentage vs cross percentage and Figure 4 shows the best cost of the hybrid algorithm and finally Figure 5 contains the best timing analysis compared with different population. The simulated output is shown in the Figure 6.

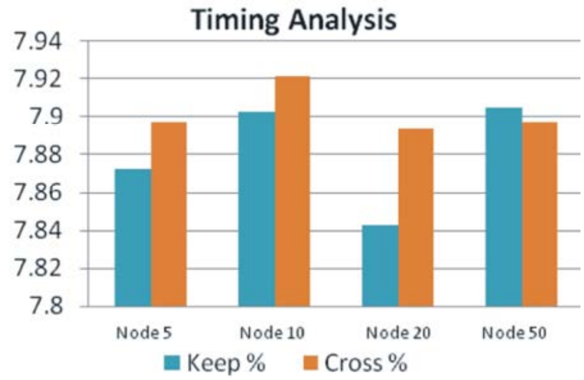

Fig. 5: Comparison of Computation Time.

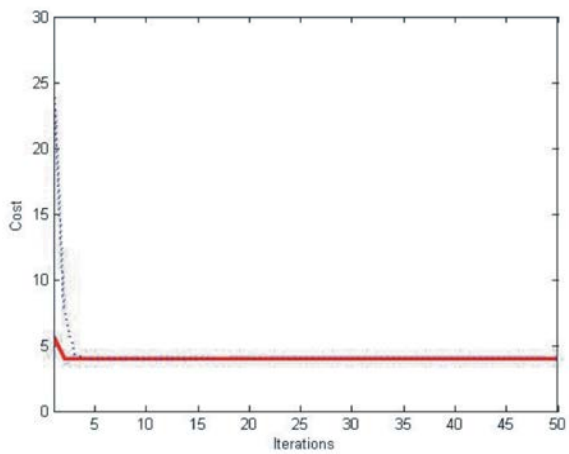

Fig. 6: Computation Iteration to obtain the Best values of Cost and Position.

CONCLUSIONS

In this paper, the two most applied computational process algorithms (GA & PSO) is used to optimize and minimize Cost, Best Position and the Total process time and it is shown in the table 1 & 2. The input parameters are constant and randomly applied to obtain the results and the number of nodes increases and the computation time is almost constant.

REFERENCES

1. Senthil Kumar, A.M. and M. Venkatesan, 2015. An Efficient Multiple Object Resource Allocation Using Hybrid GA-ACO Algorithm, Australian Journal of Basic and Applied Sciences, 9(31): 53-59.
2. Kousik Dasgupta and Brototi Mandal, 2013. A Genetic Algorithm (GA) based Load Balancing Strategy for Cloud Computing" Procedia Technology.
3. Shekhar Singh and Mala Kalra, 2014. Scheduling of Independent Tasks in Cloud Computing Using Modified Genetic Algorithm" IEEE Sixth International Conference on Computational Intelligence and Communication Networks.
4. Dikaiakos, M.D., G. Pallis, D. Katsa, P. Mehra and A. Vakali, 2009. Cloud Computing: Distributed Internet Computing for IT and Scientific Research, in Proc. of IEEE Journal of Internet Computing, 13(5): 10-13.
5. Vouk, A., 2008. Cloud computing- issues, research and implementations, in Proc. of Information Technology Interfaces, pp: 31-40.
6. Armstrong, T.R. and D. Hensgen, 1998. The relative performance of various mapping algorithms is independent of sizable variances in runtime predictions, in Proc. of 7th IEEE Heterogeneous Computing Workshop (HCW 98), pp: 79-87.
7. Yang Xu, Lei Wu, Liying Guo, Zheng Chen, Lai Yang and Zhongzhi Shi, 2008. An Intelligent Load Balancing Algorithm Towards Efficient Cloud Computing, in Proc. of AI for Data Center Management and Cloud Computing: Papers, from the 2011 AAAI Workshop (WS-11-08), pp: 27-32.
8. Ratan Mishra and Anant Jaiswal, 2012. Ant colony Optimization: A Solution of Load balancing in Cloud, in International Journal of Web & Semantic Technology (IJWesT), 3(2): 33-50.
9. Brototi Mondal, Kousik Dasgupta and Paramartha Dutta, 2012. Load Balancing in Cloud Computing using Stochastic Hill Climbing-A Soft Computing Approach, in Proc. of C3IT-2012, Elsevier, Procedia Technology, 4: 783-789.
10. Florin, P., V. Cristea, N. Bessis and S. Sotiriadis, 2002. Reputation Guided Genetic Scheduling Algorithm for Independent Tasks in Inter-clouds Environments. 27th International Conference on Advanced Information Networking and Applications Workshops, IEEE.
11. Meihong, W. and Z. Wenhua, 2010. A comparison of four popular heuristics for task scheduling problem in computational grid., 6th International Conference on Wireless Communications Networking and Mobile Computing, IEEE.
12. Dianati, M., I. Song and M. Treiber, 2002. An introduction to genetic algorithms and evolution strategies. Technical report, University of Waterloo, Ontario, N2L 3G1, Canada.
13. Joshi, K.D. and A.A. Pandya, 2003. Genetic algorithms and their applications–traveling sales person and antenna design, pp: 1-5.

14. Ernst, A., H. Hiang and M. Krishnamoorthy, 2001. Mathematical programming approaches for solving task allocation problems, Proceedings of the 16th National Conference of Australian Society of Operations Research.
15. Glover, F., 1989. Tabu search—Part I" ORSA J. Comput., 1: 190-206.
16. Hochbaum, D.S., 1995. A nonlinear knapsack problem, Oper. Res. Lett., 17: 103-110.
17. Morales, D., F. Almeida, F. Garcia, J.L. Roda and C. Rodriguez, 2000. Design of parallel algorithms for the single resource allocation problem, European J. Oper. Res., 126: 166-174.