Int. J. of Comp. & Info. Tech., (2017) 5(3): 161-167.

IJOCIT

ISBN: 2345-3877
www.ijocit.org
Volume 5, Issue 3

Original Research

Improvement of Load Balancing Infrastructure in the Cloud Computing Environment by Combining Genetic and Harmonic Search Algorithms

Pari Azimi Sefat [1]
Keyhan KhamForoosh [2, *]

Received: 22 Mar 2017
Accepted: 30 Jul 2017

Abstract
Including challenges in cloud computing systems is the load balancing problem, or resources allocations to system requests. Several reasons have caused this issue as a matter of NP-complete finds including, the heterogeneity and dynamism of the properties of resources and requests in the cloud computing environment. In this paper, an exponential approach called GA-HS is proposed to find a suitable solution for mapping a set of requests to resources in the system, with regards to conditions of the cloud computing system. During the course algorithm that runs as parallel to requests and resources categorized, Various operations, such as the creation of an initial population with the combination of effective features of the genetic algorithm and Harmony Search Algorithm methods, purposefully and with regard to the suitability of the source condition, the process of finding the response speeds up and controls the direction of implementation. The purpose of this algorithm is to apply the appropriate and parameters techniques at each stage in order to balance the load in the resources, for reduce the makespan time, and increase the speedup rate than the other algorithms.

Keywords: Cloud Computing, Load Balancing, Response Time, Resource Management, NGA Algorithm, LAGA Algorithm.

Citation: Azimi Sefat, P., KhamForoosh , K., (2017). Improvement of Load Balancing Infrastructure in the Cloud Computing Environment by Combining Genetic and Harmonic Search Algorithms, *Int. J. of Comp. & Info. Tech. (IJOCIT)*, 5(3): 161-167.

[1] Department of computer engineering , Sanandaj Branch, Islamic Azad University, Sanandaj, Iran
[*] Corresponding Author: khamfroosh@yahoo.com

1. Introduction

The concept of cloud computing refers to the use of processing capacity, memory, storage of computers and shared servers via the internet. In companies and organizations with high volumes of data processed daily, there is a need for high storage space, with high processing speeds. In the 21st century, with the possibility of easy access to the internet and portable devices, users are unwilling to use devices with high processing power, and day by day the tendency to use service-oriented services has increased. By organizations using cloud computing technology, do not need to invest heavily in the context of the preparation and implementation of hardware to provide services as well as the cost of recruiting and training professionals have to support this hardware. In traditional data centers of many of the costs and additional resources due dedicated servers due consideration for the provision of services, is wasted. Cloud computing is a technology used in data centers. These ideas are dying out of the traditional data centers are integrating servers. Current computing platforms are trying Integrated to existing facilities and, as far as possible, implement the services through fewer servers. This is done through virtualization technology. Virtualization technology saves in many areas, including physical location, electricity, cost, cooling and activation [1].

The problem of load balancing in cloud computing is one of the challenges of cloud computing. In this regard, various methods have been proposed. However, using past previous experiences to improve the load balancing algorithm is considered. This can lead to continuous improvement of the algorithm. Due to the presence of cloud computing platforms, the problem of load balancing is important for cloud computing facilities and thus providing an efficient load balancing algorithm in cloud computing environments is necessary. The main purpose of load balancing, optimizing performance and throughput, So that the response time is reduced .Based on the current state of the system, load balancing algorithms are divided into two categories: static and dynamic [2].

Static load balancing algorithms do not depend on the current state of the system and require prior knowledge of the system. With load changes at runtime they can not be adapted. The purpose of this type of algorithms is to minimize the runtime and limit communication overhead and latency. In dynamic load balancing algorithms, the decision to load balances is based on the current state of the system and does not require prior knowledge of the system. So from the point of view of static methods are better. For this reason, they help to improve the overall system efficiency with dynamic load migration. A dynamic strategy usually runs several times and may assign a scheduled task to a new host based on the dynamic state of the system [3].

In this paper, we are going to provide a new method of load balancing using the methods of optimization. Which increases the efficiency of the algorithm in cloud computing. In this method, applying a genetic algorithm [4] and taking into account the response time, each request will be made to improve the load balancing in the computational environment. Our proposed algorithm is a combination of optimal genetic (GA) and Harmonic Search (HS) algorithms.

2. Importance of Load Balancing

By balancing the load, we can balance the load by dynamically transferring the volume of local task from a machine to a remote machine or a less-used machine. This maximizes user satisfaction . Minimizes response time, increases the exploitation of resources, reduces the number of rejections and reduces system performance.

The factors responsible for load balancing are:

- **Limited energy consumption:** Load balancing can reduce the amount of energy consumed by avoiding excessive interaction of nodes or virtual machines with respect to workload.
- **Reducing carbon release:** consuming energy and releasing carbon are two sides of the same coin. Both are directly related to each other. Load balancing helps reduce energy consumption, which automatically reduces carbon releasing, and therefore reaches green computing [5].

3. Review of Past Work

Cheng et al. [6] dealt with multiple task-oriented virtual resource integration and optimal scheduling from the perspective of cloud manufacturing enterprises. They focused on the issue of scheduling tasks as many as possible onto a fixed amount of resources to obtain a higher profit for an enterprise under the constraint of delivery deadlines. Jian et al. [7] dealt with the scheduling of a batch of workshop production tasks with the same characteristic and production process. The research issue is that, given the production time and production cost of each task in a production process, how to schedule tasks to minimize the total cost and time. Different from the

aforementioned two works where only the same type of tasks was tackled, Li et al. [8] addressed the scheduling of multiple heterogeneous tasks at the subtask level. Also, the transportation of components or products between subtasks is taken into account. To achieve the optimization objectives, all of the three works above considered resource occupancy and time division sharing. Lartigau et al. [9] discussed scheduling methodology for production services in cloud manufacturing, and proposed a framework for scheduling methodology at the cloud platform level. In their approach, a couple of important concepts in production service, such as batch of a task, quantity of resources that were usually ignored or not fully considered in many previous works were taken into account.

Tao et al. [10] proposed a parallel method for service composition optimal-selection in a cloud manufacturing system. Lartigau et al. [11] dealt with the issue of cloud manufacturing service composition taking geo-perspective transportation and execution time into account. Jin et al. [12] considered correlations among cloud services in cloud manufacturing. Task scheduling in cloud computing has been intensively studied. Kumar et al. [13] discussed the scheduling of independent tasks in cloud computing using an improved genetic algorithm, which incorporates Min-Min and Max-Min algorithms. Wu et al. [14] proposed a QoS-driven task scheduling algorithm in cloud computing. In their model, they first computed the priority of tasks according to task attributes such as user privilege, task urgency, latency time and task workload, and then scheduled them according to their priority. The objective is to reduce the completion time and latency time, as well as achieve a better load balancing.

Throttled algorithm in this algorithm, the user initially sends a request to the balancer to select the virtual machine that has the least load, and when the machine is found, the operation that the user comes from gives the machine. At first the process begins with keeping a list of virtual machines and scrolls through the queue. First there is a list of virtual machines. The algorithm chooses the machine that has the least load and is also available and gives the request. If the machine is not found, Balancing load and returns value of one and the request is queued [16, 15].

Round Robin algorithm the name of the algorithm indicates that it acts as a turn-by-turn. When the data center controller receives a request from the customer aware load balancer that allocate a new car in the request. The balancer carries a Machine randomly from the list of virtual machines and the ID sends it to the data center controller. In this method, the next request is processed in a circular order [17].

Monitoring Active algorithm this algorithm maintains the virtual machine index table and the number of requests assigned to them.at first, the number of Requests assigned to the virtual machine is zero. When receive requesting the allocation of a new virtual machine from the data center controller, the balancer checks the table's load and identifies the virtual machine that loading less than the rest. If its value is greater than one in the table, it will be selected as the first virtual machine. The Load Balancer returns the Virtual Machine ID to the Data Center Controller. The data center controller sends the request to the virtual machine identified by the same ID. At this stage, the data center controller ‹to load balancer warns for next allocation [15, 16].

Central Load Balancer (CLB) algorithm in this technique, any request from the user goes toward the controller. The data center controller responds to the load balancer that there is a request for allocation. The balancer has a table load that its includes virtual machine identifier, status and priority. The balancer scans the table's load and finds the virtual machine with the highest priority, then examines the status of the virtual machines, which, if available, load the load balancer, returns the virtual machine ID, and if the virtual machine is busy, the machine chooses less priority. The controller requests to the data center load balancing a machine that it has an identifier. In this algorithm, the priority of each virtual machine is calculated based on the CPU and memory speed. According to the comparison made in [18], this algorithm has a better response time than the Round Robin, Active Monitoring, and Throttled algorithms.

HSGA algorithm in [19] This is a hybrid heuristic algorithm to find an an appropriate Schedule for workflow graph is based on the genetic algorithm. Initially, the HSGA algorithm makes tasks prioritizing in complex graphs according to their effect on each other and using the graph topology. This method is effective and effective for program to reduce completion time. Then, the Round Robin and best-fit methods combine a prototype population to get a good solution. Applying some proper operation, including mutation, is used to control an optimal solution in the algorithm. Multi-genetic genetic algorithms (NGA) are better at convergence faster because each subset population develops independently of the other. The NGA function is heavily influenced by the correct selection of parameters, such as connection topology, migration method, migration rate, and population size, shortening the number of generations to find optimal or near optimal solutions. The NGA algorithm is proposed for heterogeneous

multiprocessor systems also based on the GA algorithm focuses on the time of completion of the program and the time delay of communication. The NGA algorithm is based on the gradual integration of two phases. The first phase is proportional that the system is equipped with knowledge of all tasks and scheduling a pair of jobs on a processor, while these pairs of work are independent of each other [20]. LAGA uses two methods of genetic algorithms and learning automata algorithm to simultaneously search for state space. Learning automata and genetic algorithms are both common search tools used to solve many Np-hard issues. In the genetic algorithm, it is necessary to determine the correct rate of composition and mutation to produce the appropriate population. Also, it's very important to know how population size to reach the answer is, or more generations continue algorithms not to converge quickly and get an accurate answer, very important [21].

4. Proposed Algorithm (GA_HS)

In this paper, to solve the problem of load balancing cloud computing systems from combined tow genetic and harmony algorithms is used. Genetic algorithm works poorly despite the high abilities in sustainability and local. The purpose of the proposed algorithm combining genetic algorithm with local search capabilities general search algorithm is harmony. In cases where there are no resources available and there is no mechanism for identifying these resources first, we must deal with a specific algorithm for obtaining resources and then we can allocate these resources based on the genetic algorithm activities that can be reached in a load-balancing environment. Therefore, in order to create a balancing mechanism between data and resources, resources must first be identified first And they got information from each location for this reason, we can first use the Harmonic Search Algorithm, which is an innovative algorithm for obtaining the shortest path in a graph a complex graph of available resources to make a simple graph then, using the genetic algorithm, we can allocate these resources to some kind of activity that can be used for the best function of the system. However, in complex environments, it should be noted that the use of complex graphs in solving this problem can be very problematic and the resources available to us will be blocked away.

To this end, efforts have been made its designed for this purpose which must first be obtained by using other algorithms such as Harmony Source Resource Algorithm then obtained using the shortest route among the resources available, using the genetic algorithm we assign the best source to the activities so that the best results can be achieved [19].

The proposed algorithm (GA_HS) includes the following steps:

1. 1-First, a primary population of chromosomes is produced.
2. Calculate the fitness of each member of the population.
3. 3- Selecting parents who have a better fit than the rest of the population.
4. Applying a double-acting exchange operator for two parent chromosomes and producing two children.
5. 5- Selecting a child as a parent and applying the mutation operator and generating a new child.
6. Delivery of the new chromosome along with several primary population chromosomes to the Harmonic Search Algorithm.
7. Calculating the fitness of delivered chromosomes by Harmony Search Algorithm.
8. Creating a Matrix and Generating a New Solution or Harmony and Updating Memory by Harmony Search Algorithm.
9. 9 - Choosing the best chromosome by Harmony Search Algorithm and Delivering to the Genetic Algorithm.
10. If the condition for stopping the algorithm is reached, otherwise we will return to step 3.

At the end, the best chromosome is obtained as a solution to the problem.

5. Evaluation and Comparison

In this section we will simulate and evaluate our proposed algorithm.In order to evaluate the simulation software MATLAB have used.One of the most important MATLAB capabilities is the hardware connection that is expanding day by day. The advantage of using this software is high processing power and having various mathematical functions for working with data and analyzing and analyzing them. Then get the parameters in order to evaluate the efficiency of the proposed algorithm, compare it with NGA, LAGA, and HSGA algorithms.

5.1. Set up simulation parameters

Noted as to simulate the proposed algorithm with other algorithms in MATLAB software we use. Experimental space is a cloud computing system with different processing powers, which are distributed randomly in the environment. In other words, each processor has its own MIPS size, which can vary with its other processor. Also, the number of requests (tasks) varies from 40 to 100 jobs; the estimated time to perform any task can also vary with the task. in order to achieve more accurate results, simulation is performed several times and we compute the average results obtained from simulation implementation [19].

Table 1: Simulation Parameters

Parameter	Value
Number of tasks in application	20 - 100
Task lengths	12-72 (× 105 MI)
The number of resources	30
Resource speeds (MIPS)	500 - 1000
Bandwidth between resources	10-100 (mbps)

Table 2: Parameters used in the GA algorithm structure

Parameter	Value
Population size	20
Crossover rate	0.5
Mutation rate	0.5
Elitism selection rate	0.75
Number of iteration	20

Table 3: Parameters used in the structure of the HS algorithm

Parameter	Value
harmony memory size	10
new harmony memory size	100
harmony memory consideration rate	0.75
pitch adjustment rate	0.05
fret width	0.1
Number of iteration	20

5.2. Simulation Results

5.1.1. Evaluation of Makespan

Figure 1 shows the results of the evaluation and comparison of the proposed algorithm with the LAGA, NGA, and HSGA algorithms in terms of the Makespan parameter the same time number of tasks (tasks) increases simultaneously. As shown in Fig. 1, the proposed algorithm has performs better with the number of requests, compared to the LAGA, NGA, and HSGA algorithms, shorter Makespan time ,too. In these algorithms (NGA, LAGA, HSGA) calculates the time of calculation of resources in each node.And selects a resource with a minimum failure rate in the jump action the time to complete the timing failure rate is focused [20].

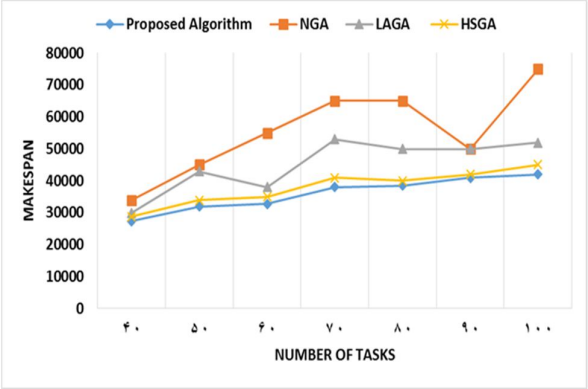

Figure 1: Comparison diagram Makespan algorithm for increasing the number of tasks

5.1.2. Evaluation of Speedup

The diagram of Figure 2 also compares the evaluation of the proposed method with these two algorithms in terms of the Speedup parameter while increasing the number of tasks . As noted earlier, increasing the acceleration factor is improved system performance. According to Figure 2 the acceleration (Speedup) compared the proposed method compared to the algorithms more and the proposed method is far better results in terms of this parameter is also produced. To minimize processing time, although it affects communication costs for sending and receiving requests and data among resources, these algorithms do not consider it.

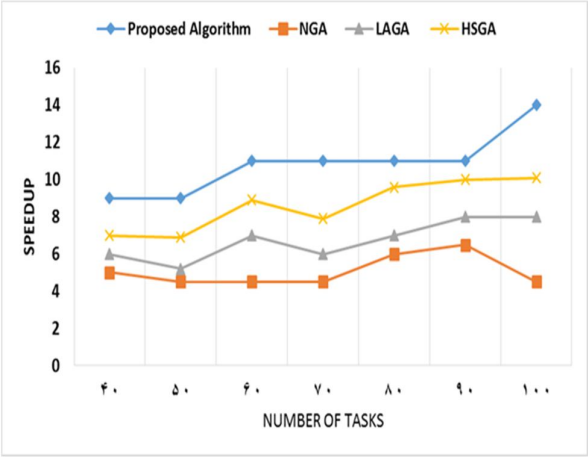

Figure 2: Comparison diagram of the Speedup algorithm for increasing the number of tasks

5.1.3. Evaluation of Load Balance

Figure 3 shows the comparison of the evaluation of the proposed method with these two algorithms in terms of the load balancing parameter while increasing the number of tasks . As shown in Figure 3, in all three methods, there is a relative load balancing, and none of these three methods have been able to create a complete load balancing between virtual machines. However, the proposed method provides better balance relative to the other two methods. the reason this event is that in proposed method load to ratio is distributed demand completion time, as well as the cost of communication between virtual machines. So the load applied by the proposed method to each virtual machine is not the same. But at the same tim eless load difference is seen. Which is because the simulation environment is heterogeneous this load balance is appropriate.

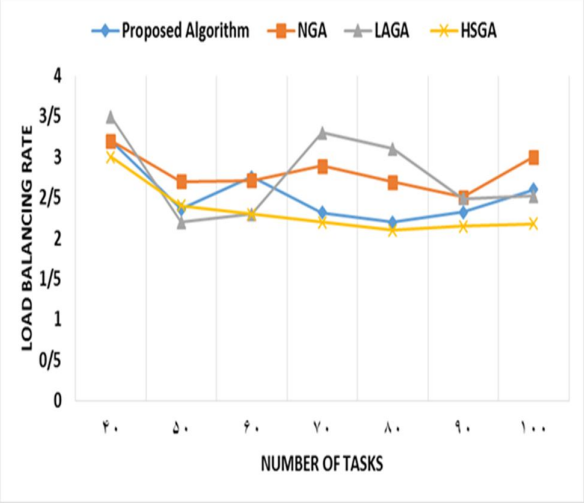

Figure 3: shows the comparison of the loading rate of the algorithm with increasing number of tasks

6. Conclusion

The newest approach in recent years has attracted great interest in cloud computing. Access to the computing environment is pervasive and independent of location and time. Cloud computing is a new generation of distributed systems whose goal is to provide service based on cost. Cost-saving, high availability and easy scalability include the benefits of cloud computing .Which are realized with the help of the technologies used. In this regard, load balancing is an issue of importance in cloud computing. In this study to create load balances in cloud computing, with combination of two genetic and harmonic search algorithms was used. So that it has worked well on issues where search space has a complex and complex set of solutions. On the other hand, the proposed method is a kind of quasi-static method, the method is a combination of dynamic and static methods, as well as the mechanism for selecting the node is the minimum overhead at the time of running the algorithm so and gathering information is available of node statically in initial implementation.

The main objective of load balancing is to achieve proper mapping of tasks on existing nodes (processors and computers) in the system, so that each node performs a fair amount of work. until the overall system reaches its minimum.In this research, we investigate various load balancing methods to reduce Makespan in cloud computing environment. Because the reduction of Makespan is one of the most important of the service level agreements and increase customer satisfaction. And also speedup, which is an acceleration factor acceleration in an algorithm known as a system efficiency improvement parameter. As well as load balancing rate to optimize the use of resources, maximize throughput, minimize response time and avoidance of overhead. So, we proposed an algorithm that mapped requests to some virtual machines that Makespan has dropped. In fact, by improving load balancing we can be achieved to degree of confidence in the field of cloud computing efficiency, thereby increasing customer satisfaction from the cloud-based service received. There is currently no algorithm that can consider all its aspects with acceptable performance. The proposed Makespan time algorithm is highly optimized for LAGA, NGA, and HSGA algorithms, and its speedup time is much higher than the LAGA, NGA, and HSGA algorithms. And the load balancing rate is better than the LAGA and NGA and HSGA algorithms. The proposed algorithm and all comparisons performed in the MATLAB environment are simulated.The results of this simulation are obtained with the number of different requests.

7. Future Works

In the future, we plan to extend this kind of load balancing for the flow of work with affiliated tasks. Also in the future, we have to improve this algorithm taking into account other factors, the quality of our planning service. It is also possible to examine the impact of population diversity and the avoidance of convergence of solutions in the genetic algorithm on the problem-solving quality of the proposed

algorithm, as well as the consideration of the maintenance cost factor. Also, we can use the combination of the proposed algorithm with other demographic algorithms and the combination of the proposed algorithm with a clustering algorithm as a future work. Combining algorithms in problem solving can produce better results.

References

[1] Stanoevska-Slabeva, K., Wozniak, T., & Ristol, S. (2009). Grid and cloud computing: a business perspective on technology and applications. *Springer Publishing Company, Incorporated*.

[2] Ajit, M., & Vidya, G. (2013, July). VM level load balancing in cloud environment. In Computing, Communications and Networking Technologies (ICCCNT), 2013 Fourth International Conference on pp. 1-5.

[3] Kansal, N. J., & Chana, I. (2012). Existing load balancing techniques in cloud computing: A systematic review. *Journal of Information Systems and Communication*, 3(1), 87.

[4] Saima, A. Gulzar. et al. (2016). A hybrid genetic algorithm for optimization of scheduling workflow applications in heterogeneous computing systems. *Journal of Parallel and Distributed Computing*. 80-90.

[5] Kansal, N. J., & Chana, I. (2012). Existing load balancing techniques in cloud computing: a systematic review. *Journal of Information Systems and Communication*, 3(1), 87.

[6] Cheng, Z., Zhan, D., Zhao, X., & Wan, H. (2014). Multi task oriented virtual resource integration and optimal scheduling in cloud manufacturing. *Journal of Applied Mathematics*.

[7] Jian, C. F., & Wang, Y. (2014). Batch task scheduling-oriented optimization modelling and simulation in cloud manufacturing. *International Journal of Simulation Modelling*, 13(1), 93-101.

[8] Li, W., Zhu, C., Yang, L. T., Shu, L., Ngai, E. C. H., & Ma, Y. (2017). Subtask scheduling for distributed robots in cloud manufacturing. *IEEE Systems Journal*, 11(2), 941-950.

[9] Lartigau, J., Nie, L., Xu, X., Zhan, D., & Mou, T. (2012). Scheduling methodology for production services in cloud manufacturing. In Service Sciences International Joint Conference ((IJCSS), pp. 34-39.

[10] Tao, F., LaiLi, Y., Xu, L., & Zhang, L. (2013). FC-PACO-RM: a parallel method for service composition optimal-selection in cloud manufacturing system. *IEEE Transactions on Industrial Informatics*, 9(4), 2023-2033.

[11] Lartigau, J., Xu, X., Nie, L., & Zhan, D. (2015). Cloud manufacturing service composition based on QoS with geo-perspective transportation using an improved Artificial Bee Colony optimisation algorithm. *International Journal of Production Research*, 53(14), 4380-4404.

[12] Jin, H., Yao, X., & Chen, Y. (2015). Correlation-aware QoS modeling and manufacturing cloud service composition. *Journal of Intelligent Manufacturing*, 1-14.

[13] Kumar, P., & Verma, A. (2012, August). Scheduling using improved genetic algorithm in cloud computing for independent tasks. In Proceedings of the International Conference on Advances in Computing, Communications and Informatics (pp. 137-142). ACM.

[14] Wu, X., Deng, M., Zhang, R., Zeng, B., & Zhou, S. (2013). A task scheduling algorithm based on QoS-driven in cloud computing. *Procedia Computer Science*, 17, 1162-1169.

[15] Wickremasinghe, B., & Buyya, R. (2009). Cloud Analyst: A CloudSim-based tool for modelling and analysis of large scale cloud computing environments. *MEDC project report*, 22(6), 433-659.

[16] Kansal, N. J., & Chana, I. (2012). Cloud load balancing techniques: A step towards green computing. *International Journal of Computer Science Issues*, 9(1), 238-246.

[17] Ahmed, T., & Singh, Y. (2012). Analytic study of load balancing techniques using tool cloud analyst. *International Journal of Engineering Research and Applications*, 2(2), 1027-1030

[18] Soni, G., & Kalra, M. (2014). A novel approach for load balancing in cloud data center. In Advance Computing Conference (IACC), *2014 IEEE International*, pp. 807-812.

[19] Delavar, A. G., & Aryan, Y. (2014). HSGA: a hybrid heuristic algorithm for workflow scheduling in cloud systems. *Cluster computing*, 17(1), 129-137.

[20] Kojima, K., Ishigame, M., Chakraborty, G., Hatsuo, H., & Makino, S. (2008). Asynchronous parallel distributed genetic algorithm with elite migration. *International journal of computational Intelligence*, 4(2), 105-111

[21] Wang, X., Yeo, C. S., Buyya, R., & Su, J. (2011). Optimizing the makespan and reliability for workflow applications with reputation and a look-ahead genetic algorithm. *Future Generation Computer Systems*, 27(8), 1124-1134.


IJOCIT
WWW.IJOCIT.ORG