



Hybrid SFLA-GA algorithm for an optimal resource allocation in cloud

S. Kayalvili¹ · M. Selvam²

Received: 25 December 2017 / Revised: 31 January 2018 / Accepted: 2 February 2018 / Published online: 16 February 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract

The cloud computing is a type of computing model which in the recent years has acquired attention due to its varied application and ease of use. It is a convenient and quick way of accessing shared resources at any time and at any place by means of using internet which is realized effectively for sharing of software and hardware resources. For managing the cloud resource and dynamic configuration for all types of underlying hardware resources that are comprised in virtualization technology to provide services to users with virtual machines (VM) as the basic unit, a key role is played by virtualization technology. Optimizing the objective in satisfying the constraint, the purpose of deploying VM is to realize the ideal outcome by altering the layout as well as the placement of all the VM. The allocation of the cloud resources to that of the user based on the request is a problem that is NP Hard. Heuristic methods are utilized for optimizing the resource allocation. The shuffled frog leaping algorithm (SFLA) has the benefit of easier implementation and high speed convergence with the capability of having global optimization and are used widely in various areas. The Genetic Algorithms (GAs) are the iterative stochastic optimization based methods that are based on the natural selection principles and their evolution. For this work, there is a hybrid SFLA-GA used for obtaining the allocation of optimal resources in the cloud computing.

Keywords Cloud computing · Resource allocation · Shuffled frog leaping algorithm (SFLA) · Genetic algorithms (GAs) · Hybrid SFLA-GA

1 Introduction

The Cloud computing has adopted a technology of virtualization which will link a large amount of resources for computing and the storage resources along with the resources of software to form a virtual resource pool and will provide services of cloud for the user or also for the application system. In Cloud computing, the user tasks are carried out by means of using the virtual technology for mapping the physical resources of the VMs and the tasks are submitted using VM based scheduling. There are two parts into which the cloud environment is divided [1]: the

primary level scheduling of the tasks submitted by the user and mapping them to the VM resources in the cloud computing environment. The Secondary level scheduling being a virtual machine (VM) and a process of host mapping which makes a VM in its suitable host to create or Migrate. Research has shown that the resource scheduling in cloud computing is NP problem. There are multiple tasks as well as targets for solving this NP problem. Artificial Intelligence algorithms have been formulated by researchers locally and abroad, for resource scheduling and significant outcomes have been obtained.

An efficient and effective utilization of such resources [2] will be needed for a requirement in the environment of cloud computing. When there is a dynamic fluctuation in the resource availability [3] the meeting of the constraints of quality of service (QoS) with the preservation of a utilization of a level that is satisfactory and the performance of the system which will be a problem that is crucial to be tackled. The dynamic behavior will arouse the challenges among cloud users and the providers and so the allocation of resources becomes the central theme in cloud computing. The meta-heuristic algorithms have a critical place in

✉ S. Kayalvili
s.kayal.cse@gmail.com; kayalvilis@gmail.com
M. Selvam
amm_selvam@yahoo.co.in

¹ Department of Computer Science and Engineering, Velalar College of Engineering and Technology, Erode, India

² Department of Computer Science and Engineering, Excel Engineering College, Komarapalayam, India

the field of research in the last few years owing to the high level of efficiency in solving the issues.

Even though allocation of resources in the environment of cloud computing is globally important the main aim is to identify the attention to this and to find an optimal as well as feasible technique for this. There are several strategies that are efficient in the resource allocation which will be effective in an environment that is constrained in cloud computing. The main challenges in this are thereby distributed in clouds and the concentration is only on the issues that are problematic in a paradigm of cloud. Resource provisioning also includes the fining of the cloud providers, the VM scheduling and utilizing the resources in an efficient manner on the basis of the pricing scheme and the management of resources and workload. The cloud computing has evolved using many techniques in provisioning that are all discussed in the different parameters like price, customer satisfaction, reliability, demand, scalability, consumption of power, and the most important aspect of the QoS of cloud. Various meta-heuristics will be having some novel variations and are suggested for the allocation of resources in many fields. There are many prominent meta-heuristic algorithms which are prominent as well as remarkable in cloud computing like that of the Ant Colony Optimization (ACO), the Cuckoo Search (CS), the Differential Evolution (DE), the Firefly Algorithm (FA), the Harmony Search (HS), the Immune Algorithm (IA), the League Championship Algorithm (LCA), the Memetic Algorithm (MA) and so on.

This work makes use of the genetic algorithm (GA) and the Shift Frog Leap Algorithm (SLFA). The former allows huge variety of problems to be optimized—for instance, Travelling Salesman Problem (TSP) can finally link to circuit design, scheduling, and delivery problems. It is a problem solving technique that can spawn multiple optimal results. This is unaffected by bad parameters and these are discarded by the GA. Despite complex and greater number of parameters, the GA can solve problems by using fitness function. Instead of merely encoding the genes, GA can relate conveniently to the existing real world scenario. The latter (SFLA) on the other hand is a meta-heuristic optimization technique; It is based on observation, imitation and modeling the behavior of a group of frogs while foraging for food source that has the maximum amount of food. Several complex optimization problems can be solved using the SFLA. These include non-linear, multi-modal and non-differentiable problems. SFLA has been successfully used across engineering optimization problems such as Travelling Salesman Problem, bridge deck repairs, job-shop scheduling arrangement and water resource distribution. Fast convergence speed is the best advantage of the SFLA, which combines the advantaged of both the PSO algorithm that is based on the behavior and also the genetic based MA.

The Bio-inspired methods of optimization [4] are widely used in both theory as well as in application and are applied successfully in various problems which cannot solved with the traditional methods like the linear programming method or the gradient method. The selection of features has improved the efficiency of calculation and the accuracy of classification using multiple features as not all the features tend to influence the accuracy of classification. The choice of suitable features will improve the accuracy of prediction and this happens by avoiding the choice of features that are inappropriate. In a conventional method the SFLA will get trapped early as this does not make use of a mutation that is similar to that of the GA which when trapped temporarily can continue to escape the local optimum by means of mutation and the SFLA conventionally used will not be able to do so. The SFLA-GA technique combines the advantages of the SFLA and an exchange of information among the individual members in the group (the frogs) and this will implement a local search by means of using the GA for evaluating the results of processing. By using the SFLA-GA, the feature selection has been accomplished and the accuracy of classification is calculated. The combination of a GA with that of a SFLA will make sure that the progress is made and a global solution found.

2 Related work

The allocation of resources is a challenging aspect in Cloud and it depends on the user request that is heterogeneous and the resource providers' dynamicity. For allocating this efficiently, an Efficient Resource Allocation (ERA) has been proposed by [5]. In this mechanism the Job Centric Cosine Similarity based K-Means (JCC K-Means) has been used for clustering the resources on the basis of the user request and a branch as well as bound based assignment algorithm has been used to efficiently allocate the resources of cloud. This work has been simulated and compared with that of the K-means and also with Random Allocation. A simulation is completed by using Java and Eclipse IDE and the results prove that this work has achieved better results and is efficient in discovery of resource.

Luo et al. [6] had proposed one novel method of integer optimization used for a strategic planning by means of an integration of allocation and the coordination method within a brand and bound paradigm. Owing to the distributed computation of the coordination of resource method along with the evaluation of nodes in that of the brand and the bound paradigm, this method could solve the problem of integer programming. In the branch-and-bound paradigm the solution space size will decrease in a monotonic manner and the global optimal solution to that of the integer programming is got by using this method. Lastly the method is

applied to the optimal charging control and the problem of the electric vehicles that are under the grid conditions that are constrained in case of a study in simulation.

This branch-and-bound (B&B) framework makes use of a tree research based strategy for enumerating the possible solutions in any problem. Identified are three components of the algorithm in B&B which may be specified by the user for a behavior fine tuning. They are the search strategy, branching strategy, and pruning rules. A survey that presented a description of the research advances in designing B&B algorithms was made by Morrison et al. [7] in which directions for further exploration and motivation was made.

A shuffled frog leaping algorithm (SFLA) falls within a local optimum that helps solving problems in multi optimum function and also impacts the accuracy as well as speed of convergence. So, Liu et al. [8] further presented a grouped SFLA in order to solve the problems in continuous optimization that was combined with the traits of the transformation of cloud model and its transformation between that of the qualitative and the quantitative research. This algorithm further divides the domain of definition into different groups and provided every group with one set of frogs and for each region search of frogs the definition domain is a regions search called the memplex and this search will progress as the “elite strategy” for updating the information on location of the elite frogs using an algorithm of cloud model. The method also narrows its search space and will improve this situation effectively and the results of this computer simulation in terms of speed and accuracy could confirm this.

The Resource scheduling in case of that of cloud computing has proved to be the main focus of this research. An analysis of the present situation and an introduction of the shuffled frog leap algorithm has been made by Chen and Huang [4]. The aim is at keeping the SFLA from falling within the local optimum and with a convergence speed that is fast. This improved method of SFLA effectively avoids the algorithm from being inside the local optimum and has reduced global searching and the optimum time. This classic function has proved that the performance of this algorithm within that of this paper has improved the efficiency of this system into that of the tasks of processing and keeps allocation of resources in cloud computing very effective.

Nguyen [9] had further proposed a Hybrid SFL-Bees Algorithm which is a combination of the strengths of SFLA along with the Bees Algorithms (BA). The SFLA was able to identify solutions that are optimal very quickly owing to directive searching and information exchange and the BA will have a higher random that can make it escape the local optima. So the Hybrid SFL-Bees Algorithm can find the solutions fast like the SFLA and escape from local optima like the BA. The Numerical simulations were done on two different benchmark functions which are the Griewangk’s

function (F8) and the Schwefel’s function (F7), and their comparative results proved the effectiveness of this algorithm that could outperform the SFLA and also the BA.

Under Infrastructure as Service, cloud users can use the cloud computing resources on a pay-per-use basis. Based on the request of the user, the resources are allocated virtually. As the challenges under this heterogeneous environment are exacerbating, it becomes a challenge to assign the resources optimally so that better QoS is obtained in a less amount of time. There are several experiments that are being performed for the optimal allocation of the virtual resources in the cloud environment. There are several algorithms that can improvise the efficacy of the allocation of resources, taking into account the QoS parameter. This work provides a new technique for optimally allocating resources to multiple subtasks need a given number of resources to be scheduled at a given cost. This work addressed the problem of optimal resource allocation using the SFLA-GA algorithms.

3 Methodology

The section has explained in detail the branch and bound algorithm, the genetic algorithm, the SFLA and finally the hybrid SFLA-GA.

3.1 Branch and bound algorithm (B&B)

The branch-and-bound [6] is that design paradigm used in problems of combinatorial optimization for integer or the discrete variables. The disregarding of the time of in which the algorithm has been designed in accordance to the paradigm of branch and bound will find a global optimum for a problem of combinatorial optimization. So, a method of resource allocation and its coordination has been proposed.

The B&B has encapsulated a whole family of such algorithms and the procedure has implicitly enumerated the possible solutions faced in the problem and stores the partial solutions which are sub problems. The nodes that are not explored in a tree will generate children and this is done by partitioning this into smaller regions solved recursively (through branching), with the rules being used to prune the regions of search space that are suboptimal. As soon as the whole tree is explored an ideal solution will be found and the search will be returned. But this framework has three components that are not specified. They are the search strategy (which refers to the order in which the sub problems of the tree get explored), a branching strategy (based on the partition of solution space for producing new sub problems in a tree) and finally pruning rules (the rules which prevent the exploration of the regions that are sub-optimal regions). The algorithm of the Branch and Bound Algorithm is shown in Fig. 1.

Algorithm: Branch-and-Bound(X, f)

- 1 Set $L = \{X\}$ and initialize \hat{x}
- 2 **while** $L \neq \emptyset$
- 3 Select a sub problem S from L to explore
- 4 **if** a solution $\hat{x}' \in \{x \in S \mid f(x) < f(\hat{x})\}$ can be found : Set $\hat{x} = \hat{x}'$
- 5 **if** S cannot be pruned:
- 6 Partition S into S_1, S_2, \dots, S_r
- 7 Insert S_1, S_2, \dots, S_r into L
- 8 Remove S from L
- 9 Return \hat{x}

Fig. 1 Algorithm for Branch and Bound

Relating to the pseudo code [7] a search strategy will affect the order wherein the nodes are chosen for being explored. Its branching strategy will affect the actual number of children and the manner in which this sub -problem has been partitioned, finally the pruning rules for determining if the S is gauged. Most of the times there is an initial solution that is found through a heuristic procedure. The Algorithm has been guaranteed for terminating the X which is finite and its procedure of partitioning will create some child sub problems S_i which are some proper subsets in S for every sub problem S . The set X (and all the sub problems) are given and a given element x , a membership in one certain sub problem is efficiently checked and the sub problem partition is computed will the members of the X . There are two important factors to this which are the branching factor b for the tree, that denotes the maximum children generated in a node in the tree and a search depth d for the tree that is the longest path from the root to the leaf. So this algorithm will operate in the $O(Mb^d)$ worst-case running time, in which M denotes a bound on the time required to explore its sub problem and the pruning rules in this can improve the performance of the algorithm.

Observed are two phases in this algorithm which is the search phase, where the algorithm does not yet have an optimal solution x^* . The next is verification phase, where an incumbent solution will be optimal and has some unexplored sub problems in this tree which are yet to be pruned. Any incumbent solution is not proven as optimal until such time there are no more sub problems that are left. The distinction between the search and verification phases are not known until termination.

3.2 Genetic algorithm (GA)

The Genetic algorithms are those techniques that are on the basis of the mechanics of such natural selection and their natural genetics. They are different from the normal procedures in many ways [10, 11]. For instance, these genetic

algorithms will encode parameters within the solution space used as the strings, and will work these strings, and make use of the objective function, for evaluating the solution quality in heuristics for guiding their search on problem specific knowledge and their insights.

The GA [12] is that algorithm that is probabilistic and also imitates the natural evolution and its progression. The process of biological evolution for the chromosomes is an idea of the GA and has been based on the idea of survival of the fittest in which there are better solutions which are obtained by means of a recombination of this. In this algorithm, there is a population of strings that are called chromosomes have encrypted the applicant solutions for the problems in optimization. Allocation of sufficient and suitable resources for the VM is an issue that is quite challenging and the Virtualization technology on which the facilities of cloud computing are based will be the sharing of resources between that of the VMs and so there is an aggregation by the resource manager of all accessible resources and there is also an enhancement of resource utilization. There is an advantage in Genetic Algorithm (GA) for resolving issues in resource allocation and also a new model has been recommended. There is an objective of discovering a trade-off solution among the time of task completion and the energy consumption in computing data which permit exploring the solution space to search efficiently. This is both scalable and also energy efficient based on the model for capturing the network topology of the data center and the consumption of device power. The GA pseudo code is:

```

Input : Populationsize, Problemsize, Pcrossover, Pmutation
Output : Sbest
Population ← InitializePopulation(Populationsize, Problemsize);
EvaluatePopulation(Population);
Sbest ← GetBestSolution(Population);
while : StopCriterian () do
  Parents ← SelectParents (Population, Populationsize);
  Children ←  $\emptyset$ ;
  for each parent1, parent2 ∈ parents do
    Child1, Child2 ← Crossover(parent1, parent2, Pcrossover);
    Children ← Mutate(Child1, Pmutation);
    Children ← Mutate(Child2, Pmutation);
  End
  EvaluatePopulation (Children);
  Sbest ← GetBestSolution(Children);
  Population ← Replace(Population, Children);
End
Return Sbest;

```

3.3 Shuffled frog leaping algorithm (SFLA)

An SFLA [9] is that method of meta-heuristic optimization which mimics the memetic evolution of a group of frogs that seek location. This algorithm consists of elements of the local as well as the global exchange of information. An SFLA has a population of the possible solutions which have been defined by that of a virtual frogs. These frogs are grouped into subsets called memeplexes each having an individual frog that gets influenced by the other frogs and their ideas. The memeplex is used for performing individual search by using a particle swarm optimization-like method. For ensuring its exploration globally, there are some evolution steps in memeplex where the virtual frogs get shuffled into newer memeplexes using a technique like the shuffled and complex evolution algorithm.

Additionally, provision of chances of a generation of an improved information the random and the virtual frogs get generated and also substituted within the population in case the local search is not able to find solutions which are better. Such local searches and their process of shuffling will continue till the convergence criteria gets satisfied. The SFLA flowchart has been illustrated in Fig. 2.

3.4 Hybrid shuffled frog leaping algorithm-genetic algorithm (SFLA-GA)

This SFLA is an evolutionary algorithm that is quite similar to that of the MA and the PSO usually used for searching for a global solution [13], i.e., using problems of real numbers. In case of a solution space the frogs in an SFLA indicate a solution, and every frog is grouped into a different memeplex. Each such memeplex further represents another small part of this solution space. There is an independent local search that is done in each such memeplex and its steps will have the shuffling executed where the frogs will interchange messages and this will feature speedy searches that are similar to that of the other algorithms of optimization which may be able to get trapped within that of the local optima and also prevent this from identifying a global optimal solution. To ensure such traps are avoided there is a need to combine the GA and this is used as a substitute for the updating of the equation of SFLA and the GA has to be optimally combined to preempt the trapping in the local optima. This can be effectively done when the GA conducts a local search. For this algorithm every memeplex will independently evolve for making local searches in various regions of the solution space after which such memeplexes get shuffled and also re-divided into newer memeplexes for making global searches by means of exchanging information with one another.

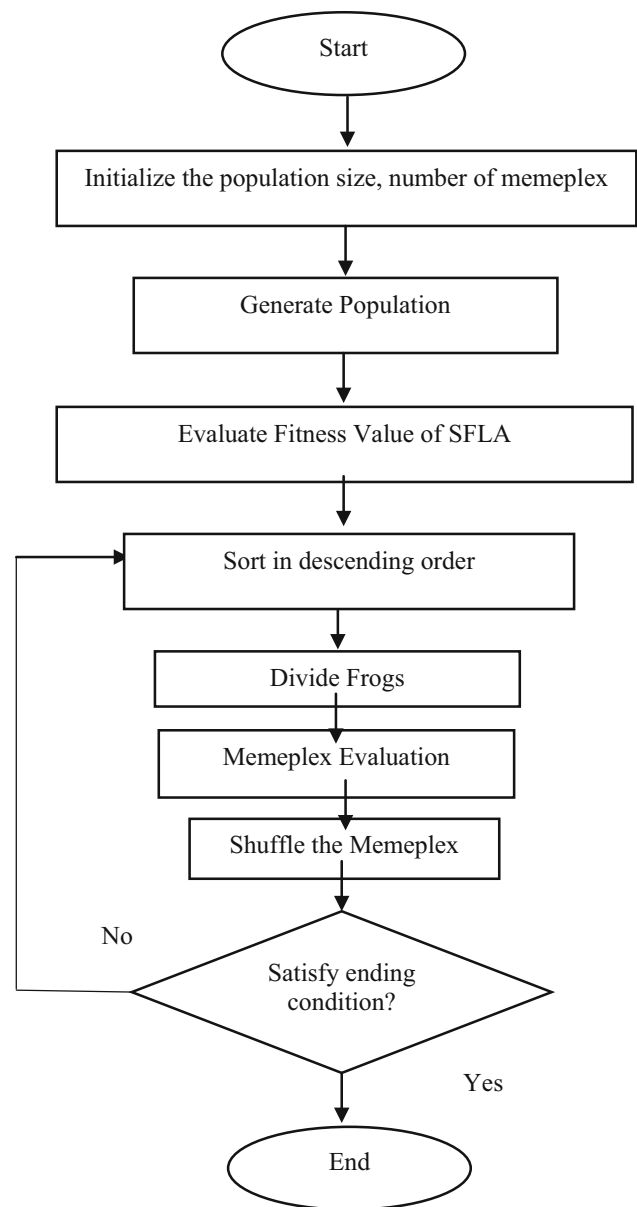


Fig. 2 Flowchart of the SFLA

The code models of the GA chromosome, and SFLA frogs are the codes used in the real-number mode. For conducting the resource allocation by means of using optimal algorithms, a code model for the optimal algorithms was made and this modified the position of every solution and also represented this as its binary string. The string length that was equal to its resources numbers. A bit value `_1_` indicates a chosen resource and a bit value of `_0_` a non-selected resource. There was also a modification of every solution (the chromosomes and the frogs) which had to be represented as its binary string in order to ensure the SFLA and the GA be combined within the SFLA-GA, and its expressions that are easily interchanged at the time of selection. While conducting the evolution of

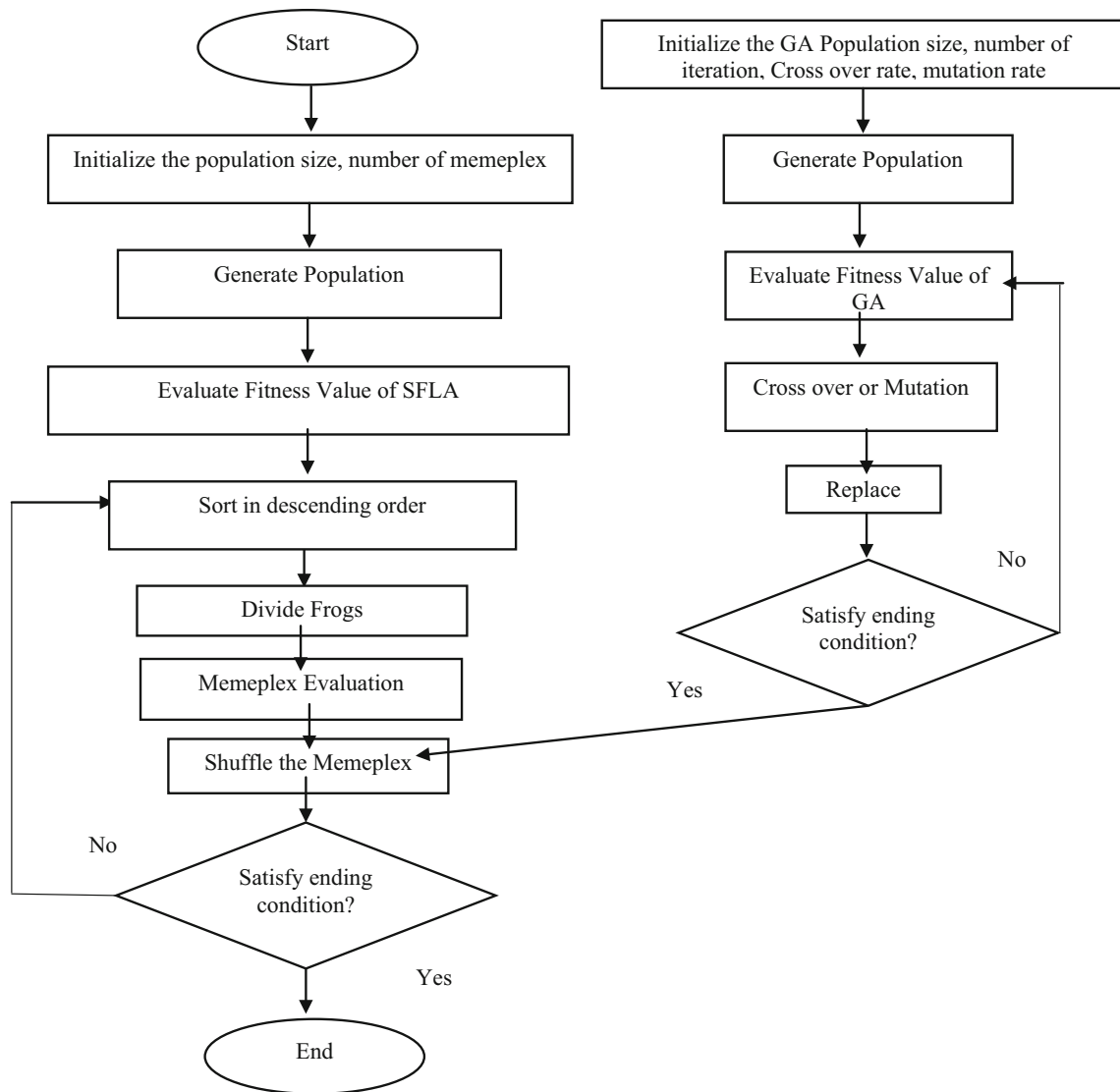


Fig. 3 Flowchart of the Hybrid SFLA-GA algorithm

memplexes for the SFLA the frog features will tend to represent the GA chromosomes, and also the GA that will be used for conducting any local search. The Flow chart of this SFLA-GA algorithm is shown in Fig. 3.

4 Results and discussion

Tables 1, 2, 3, 4 and 5 and Figs. 4, 5, 6, 7 and 8 indicate the expected time taken for completion in seconds for that of the QoS at the time the buyers outnumber the sellers, and for the QoS when the trust is very high, for the Increase in that of the Pricing from that of an average pricing % and also the Decrease in the Pricing from that of the average pricing %.

Table 1 QoS when sellers outnumber buyers

Number of jobs	Branch and bound	SFLA-GA
500	67.32	59.35
1000	139.22	125.22
1500	224.63	196.93
2000	287.12	255.69
2500	367.44	325.71
3000	431.42	388.16

From Table 1 and Fig. 4, it is observed that the SFLA-GA performs better for QoS when sellers outnumber buyers by 10.59% than branch and bound at number of jobs 1000. Also, SFLA-GA performs better for QoS when sellers outnumber buyers by 11.58% than branch and

Table 2 QoS when buyer outnumber sellers

Number of jobs	Branch and bound	SFLA-GA
500	71.3	62.02
1000	146.44	129.69
1500	236.78	207.28
2000	306.17	277.11
2500	393.05	343.91
3000	461.05	403.29

Table 3 QoS when trust is high

Number of jobs	Branch and bound	SFLA-GA
500	65.57	57.73
1000	135.09	119.22
1500	219.85	197.82
2000	281.21	246.32
2500	356.94	310.09
3000	419.39	374.72

Table 4 Increase in pricing from average pricing %

Number of jobs	Branch and bound	SFLA-GA
500	11.88	13.34
1000	10.89	12.06
1500	14.21	15.83
2000	14.67	16.16
2500	15.22	17.26
3000	15.46	17.34

Table 5 Decrease in pricing from average pricing %

Number of jobs	Branch and bound	SFLA-GA
500	11.02	12.27
1000	10.89	12.36
1500	11.46	13.05
2000	12.78	14.48
2500	14.22	16.18
3000	13.06	15.97

bound at number of jobs 2000. Again SFLA-GA performs better for QoS when sellers outnumber buyers by 10.56% than branch and bound at number of jobs 3000.

From Table 2 and Fig. 5, it is observed that the SFLA-GA performs better for QoS when buyers outnumber sellers by 12.13, 9.96 and 13.37% than branch and bound at number of jobs 1000, 2000 and 3000 respectively.

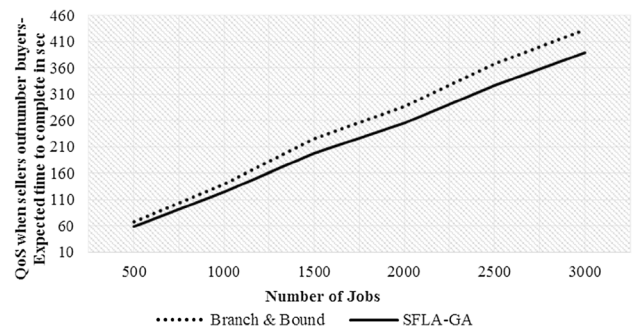


Fig. 4 QoS when sellers outnumber buyers

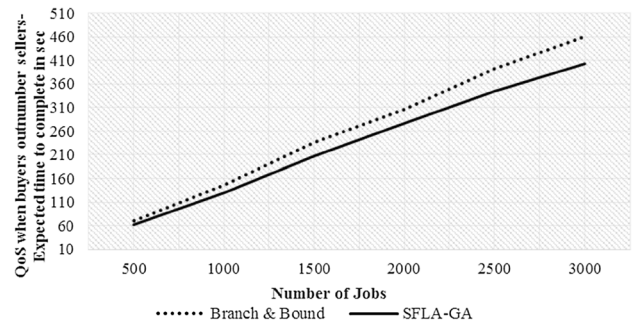


Fig. 5 QoS when buyer outnumber sellers

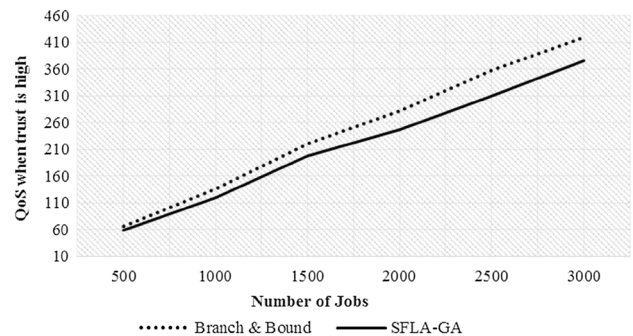


Fig. 6 QoS when trust is high

From Table 3 and Fig. 6, it is observed that the SFLA-GA performs better for QoS when trust is high by 12.48% than branch and bound at number of jobs 1000. Also, SFLA-GA performs better for QoS when trust is high by 13.23% than branch and bound at number of jobs 2000. Again SFLA-GA performs better for QoS when trust is high by 11.25% than branch and bound at number of jobs 3000.

From Table 4 and Fig. 7 it is observed that the SFLA-GA performs better by increasing the increase in Price from average pricing percentage by 10.19% than branch and bound at number of jobs 1000. Also, SFLA-GA

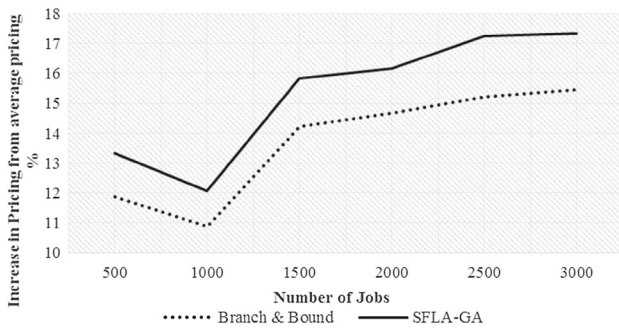


Fig. 7 Increase in Pricing from average pricing %

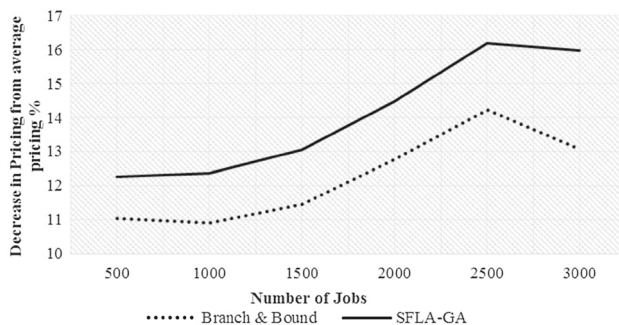


Fig. 8 Decrease in Pricing from average pricing %

performs better increasing the increase in Price from average pricing percentage by 9.67% than branch and bound at number of jobs 2000. Again SFLA-GA performs better by increasing the increase in Price from average pricing percentage by 11.46% than branch and bound at number of jobs 3000.

From Table 5 and Fig. 8 it is observed that the SFLA-GA performs better by increasing the decrease in Price from average pricing percentage by 12.65% than branch and bound at number of jobs 1000. Also SFLA-GA performs better increasing the decrease in Price from average pricing percentage by 12.47% than branch and bound at number of jobs 2000. Again SFLA-GA performs better by increasing the decrease in Price from average pricing percentage by 20.05% than branch and bound at number of jobs 3000.

5 Conclusion

The Cloud users will be able to consume all the resources and also pay as per the use. The main and crucial aim here is to find a technique of resource allocation which is both feasible and optimal in a given service. This B&B framework will be a widely and fundamentally used method in the production of the exact solutions to that of the problems of NP-hard optimization and this hybrid SFLA-GA has

been proposed for the obtaining of an allocation of optimal resource. It has been proven by outcomes that at the time when the sellers are greater than buyers, the SFLA-GA performs much better for the QoS by 10.59% compared to the branch and bound when there are about 1000 jobs. Also, when the sellers exceed the buyers, the SFLA-GA performed better by decreasing the expected time of completion in seconds for QoS by about 11.58% compared to the branch and bound when there are 2000 jobs. Once more, when the sellers are more than the buyers, the SFLA-GA performed better by decreasing the expected completion time in seconds for the QoS by 10.56% compared to the branch and bound when there are 3000 jobs.

References

1. Kan, N., Jin-dong, W., Heng-wei, Z., Na, W.: Cloud resource scheduling method based on estimation of distribution shuffled frog leaping algorithm. In: 3rd International Conference on Cyberspace Technology (CCT), pp. 1–6 (2015)
2. Anuradha, V.P., Sumathi, D.: A survey on resource allocation strategies in cloud computing. In: International Conference on Information Communication and Embedded Systems (ICICES), Chennai, pp. 1–7 (2014)
3. Jayanthi, S.: Literature review: dynamic resource allocation mechanism in cloud computing environment. In: International Conference on Electronics, Communication and Computational Engineering (ICECCE), Hosur, pp. 279–281 (2014)
4. Chen, X., Huang, W.: Research of improved shuffled frog leaping algorithm in cloud computing resources. *Int. J. Grid Distrib. Comput.* **9**(3), 71–82 (2016)
5. Moorthy, R.S.: An efficient resource allocation (ERA) mechanism in IaaS cloud. In: 2015 IEEE International Conference on Advances in Computing, Communications and Informatics (ICACCI), August 2015, pp. 412–417 (2015)
6. Luo, R., Bourdais, R., van den Boom, T.J., De Schutter, B.: Integration of resource allocation coordination and branch-and-bound. In: 2015 IEEE 54th Annual Conference on Decision and Control (CDC), December 2015, pp. 4272–4277 (2015)
7. Morrison, D.R., Jacobson, S.H., Sauppe, J.J., Sewell, E.C.: Branch-and-bound algorithms: a survey of recent advances in searching, branching, and pruning. *Discret. Optim.* **19**, 79–102 (2016)
8. Liu, H., Yi, F., Yang, H.: Adaptive grouping cloud model shuffled frog leaping algorithm for solving continuous optimization problems. *Comput. Intell. Neurosci.* (2016). <https://doi.org/10.1155/2016/5675349>
9. Nguyen, D.H.: A Hybrid SFL-Bees Algorithm. *Int. J. Comput. Appl.* **128**(5), 13–18 (2015)
10. Brownlee, J.: *Clever Algorithms: Nature-Inspired Programming Recipes*, pp. 97, 176, 243, 275. Springer, Berlin (2011)
11. Wang, P.C., Korfhage, W.: Process scheduling using genetic algorithms. In: IEEE 1995 Proceedings of Seventh IEEE Symposium on Parallel and Distributed Processing, October 1995, pp. 638–641 (1995)
12. Portaluri, G., Giordano, S., Kliazovich, D., Dorronsoro, B.: A power efficient genetic algorithm for resource allocation in cloud computing data centers. In: 2014 IEEE 3rd International Conference on Cloud Networking (CloudNet), October 2014, pp. 58–63 (2014)

13. Eusuff, M., Lansey, K., Pasha, F.: Shuffled frog-leaping algorithm: a mimetic meta-heuristic for discrete optimization. *Eng. Optim.* **38**(2), 129–154 (2006)



S. Kayalvili is working as Selection Grade Assistant Professor at Velalar College of Engineering and Technology, Erode, Tamil Nadu. She completed her P.G. degree in 2009 securing seventh rank in Anna University. Her areas of interest include Grid and Cloud Computing, Big Data and Security. She has secured Silver and Bronze medals for conducting Infosys Campus Connect Foundation Programme to students. She has got 10 years of experience

in teaching and published 17 papers in various international and national journals and conferences.



M. Selvam is working as a Professor and Head of Departments of Computer Science and Engineering and Information Technology, Excel Engineering College, Komarapalayam, Namakkal (D.T.), Tamilnadu. He completed his Ph.D. in February 2010. His areas of interest are Natural Language Processing, Enterprise Computing, Grid and Cloud Computing, Pervasive Computing, Distributed Systems and Computer Networks. He is the recipient of

Best faculty Awards for the years 2001 and 2005 in Kongu

Engineering College, Erode, Best Educationist Award from International Institute of Education and Management, New Delhi and Rashtria Vidhya Gourav Gold Medal Award from Indian Solidarity Council, New Delhi in July 2013. He was the rank holder in Bharathiyar University in his PG programme in Computer Science and Engineering. He has conducted many industrial programmes of Infosys, Wipro and IBM. He has got 22 years of experience in Teaching and 5 years of experience in industry. He has published 47 papers in various International and national Journals and conferences.