IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received May 18, 2019, accepted June 4, 2019, date of publication June 12, 2019, date of current version June 28, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2922400

Heuristic Search Based Localization in Mobile Computational Grid

DINESH PRASAD SAHU[1], KARAN SINGH[1], MANISHA MANJU[2], DAVID TANIAR[3], LE MINH TUAN[4,5], LE HOANG SON[5], MOHAMED ABDEL-BASSET[6], AND HOANG VIET LONG[7,8]

[1]School of Computer and Systems Sciences, Jawaharlal Nehru University (JNU), New Delhi 110067, India
[2]Department of Computer Science Engineering, G B Pant Government Engineering College, New Delhi 110020, India
[3]Faculty of Information Technology, Monash University, Clayton, VIC 3800, Australia
[4]Hanoi University of Home Affairs, Hanoi 10000, Vietnam
[5]VNU Information Technology Institute, Vietnam National University, Hanoi 10000, Vietnam
[6]Department of Operations Research, Faculty of Computers and Informatics, Zagazig University, Sharqiyah 44519, Egypt
[7]Division of Computational Mathematics and Engineering, Institute for Computational Science, Ton Duc Thang University, Ho Chi Minh City 700000, Vietnam
[8]Faculty of Mathematics and Statistics, Ton Duc Thang University, Ho Chi Minh City 700000, Vietnam

Corresponding author: Hoang Viet Long (hoangvietlong@tdtu.edu.vn)

ABSTRACT In recent years, the number of cell phones in society has increased drastically and they are getting popular due to their computational ability and adaptability. Resource provisioning is important, but still remains NP-hard problem in mobile computational grid (MCG). Once the jobs are assigned to the MCG, the main challenge is how to identify the correct resource according to the job's requirement and use them to execute the sub-jobs. The heuristic methods such as Min-Min, Max-Min, and HEFT can be used to select appropriate resources from the MCG that is assigned for job execution. Since the computational nodes are static and mobile in nature, the performance of such heuristics is not as expected. Such heuristics suffers from low throughput and low speedup. The process of localization is used in a wireless sensor network with good results. The proposed model uses heuristics and localization process for optimizing the quality of service parameter localization, normalized speedup, and throughput in MCG, with the concept of grid nodes available in MCG. The observation shows significant improvement in the quality of service parameter localization, normalized speedup, and throughput in MCG. The proposed model HGLA and MIN-MIN, MAX-MIN, and HEFT are compared with respect to localization, speedup, and throughput. The results reveal that the proposed model shows better performance over MIN-MIN, MAX-MIN, and HEFT.

INDEX TERMS Mobile agent, mobility, resource allocation, speed-up, localization ratio, resource provisioning, MIN-MIN, MAX-MIN, HEFT.

I. INTRODUCTION

People are developing new procedures that can deal with complex significant problems. To handle complex problems in less time, different problem-solving paradigms have been found. Parallel processing, Distributed computing, Cluster computing, Cloud computing and Grid computing are some examples [1], [52]. Till now, we have observed a significant improvement in mobile devices. These mobile devices are becoming more important by their adaptability nature and computational capability. These mobile devices are incorporated in computational grid so that their processing cycle can be properly utilized. Static grid with mobile devices built a mobile grid. An interface is built so that mobile devices and computational grid can communicate with each other and their capability can be used in proper manner [2].

Mobile computational grid (MCG) is a combination is stationary and mobile-computed devices [53]. Latest mobile devices are smart, they suit user's requirements and are able to solve computationally intensive problem [3]. Small size smart compute mobile devices are also included with the very large-scale integration.

One of the important problems in MCG is scheduling of resources in order to solve a computationally intensive

The associate editor coordinating the review of this manuscript and approving it for publication was Muhammad Imran.

2169-3536 © 2019 IEEE. Translations and content mining are permitted for academic research only.
Personal use is also permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

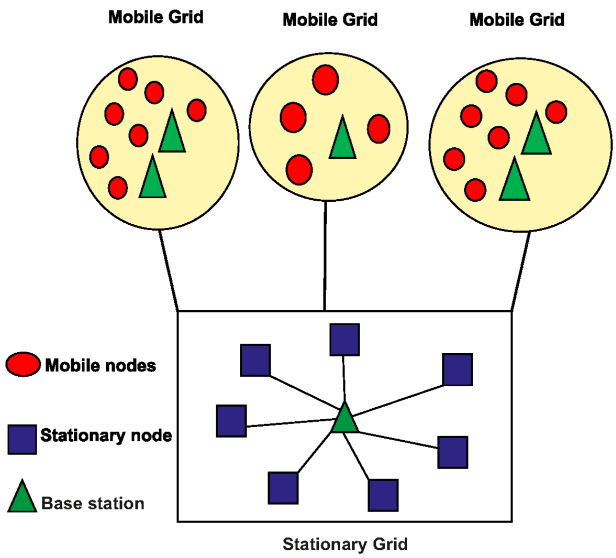

FIGURE 1. Mobile computational grid.

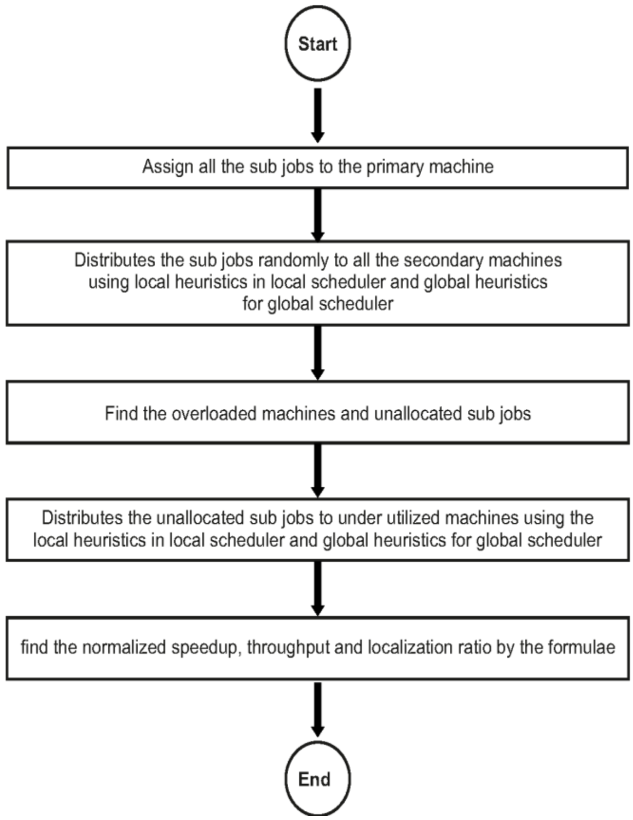

FIGURE 2. HGLA algorithm.

problem [4]. **Scheduling problem in MCG** is NP-hard by nature [5], [6]. Since the static and mobile nodes are there in MCG, the computation done by the mobile nodes must be given back to the job if mobile node has moved from one work station to another. The *process of localization* will help to find the local resource in order to satisfy the request of a job [4]. The *heuristic based search methods* are also found useful to locate the correct resource for the job [7], [8]. Another issue is how the *communication* takes place among the compute nodes [9].

In MCG, assets can be information, documents, system resource, data transmission, storage capacity, different instruments, programming applications, PCs. They all are associated and middleware programming layer is used for administrations to security, work monitoring, resource administration and so on. Since MCG comprises of mobile nodes, it is likely that these gadgets change their area occasionally. As these gadgets change their area, the network connection (topology) of these gadgets continues changing bringing. Heterogeneity of the mobile device and the **dynamic change in topology is the biggest challenge** in this framework to perform the resource scheduling [5], [7].

Many researchers [30], [31], [33] have tried to solve this complex problem and not bale to give the optimal solution. The resources in MCG are available locally or globally or in the both places. The schedulers in MCG are classified as follows.

A. LOCAL SCHEDULER (LS)

The LS is an important service executes on MCG that is part of the shared intelligent agent in order to manage end mobile services. Software or operating system provisioning is performed by host server. The LS is used to schedule the jobs on devices periodically. After the progress of LS program, it is integrated on the device with the assistance of scheduling interface. The local scheduler allocates each task with identity number. The LS runs a script having an ID with range that is changed from the default LS scripts with End-point Management [9], [10].

B. GLOBAL SCHEDULER (GS)

The GS is responsible to select suitable local place and map to the jobs onto the designated place. In this each part of the GS sets priority to the jobs. By using a mapping such priorities are associated with global priorities [9,10].

Localization is a method which will help to *assign the resources to the local schedular.* The resources are either locally or globally available or both in the MCG. The localization method will optimize the quality of services (QoS) parameters such as resource allocation time, etc. [14], [16]. For using the localization process the problem-solving techniques may be used. There are some efforts to solve that problem as below.

Guangjie et al. [11] proposed a localization algorithm using mobile anchor node in WSNs. MANAL algorithms are categorized as: localization based on mobility model and localization using path scheduling scheme. They introduced a complete review for the most fascinating and effective developments. The most important problem for MANAL procedure is the determining the movement path through which the

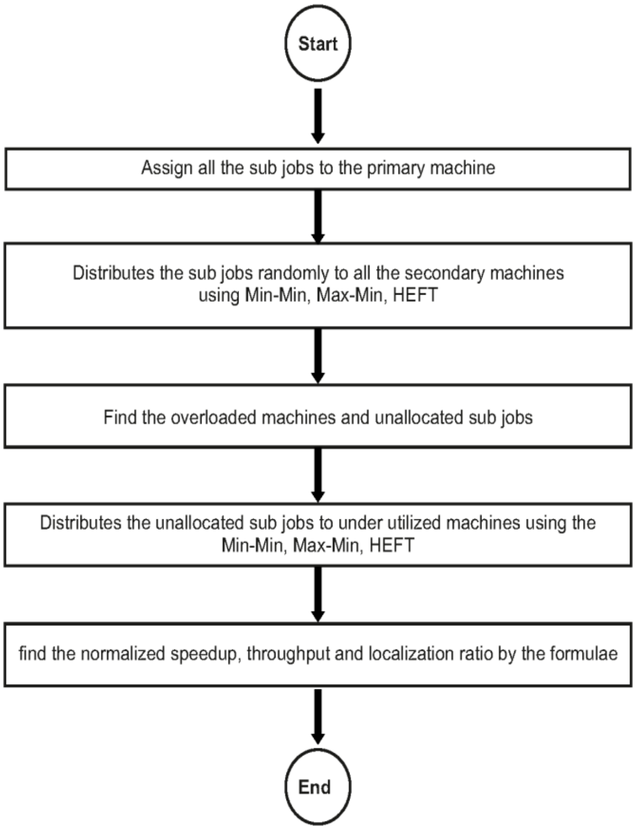

FIGURE 3. **Min-Min, Max-Min and HEFT guided localization in MCG.**

anchor nodes move along for the optimization of localization ratio under some monitoring region. Next important problem in MANAL procedure is to identify the process of localization which is used by other nodes to calculate their locations depending upon the beacon information obtained by moving location aware anchor device. Such procedures either uses moving anchor nodes or reference node and moving anchor nodes to assist the other nodes for localization.

Chenglu et al. [12] gave a system for mobile mapping which is used specially for using indoor in non-GNSS/GPS. By presenting 6-DOF localization, 2D and 3D maps can be generated by the system. Simulation result shows that given EKF-based approach blending 2D laser scanning and IMU data effectively minimizes the fault when there is a movement in system. Firstly, backpack mobile mapping is used on indoor non-GPS scenario. A person which has a backpack system can apply the movement using roll and pitch method; such movements requires 6-DOF to find pose computation for the mapping system. They introduced a procedure for pose identification which uses the 2-D scanner and 6-DOF pose tracking with the help of Extended Kalman Filter.

Fatih et al. [13] used crowd GPS to find the localization of lost object. In order to measure the profit, new parameters were used. The GA and heuristic algorithm are used to obtain the clustering of users. The synthetic and real-world social network dataset is used to perform rigorous simulation. Efficient clustering was obtained of users by preserving their privacy. Localization is determined by GPS crowd with GPS assistance of objects which are lost. The users are clustered and a beacon network is introduced where the lost devices get the beacon from each other using localization. A parameter to measure such benefits is proposed and the users adding larger profit are clustered into one group and this way using active localization can be used to obtain more benefit. GA and heuristic greedy algorithm are applied to create clusters of users. A large number of experiment and simulation has been done on dataset. The observation is that efficient partition is created by a smaller number of communications between users by preserving their privacy.

Park and Roh [14] proposed a global localization method using on place learning and a 2-D range scan where SVM was used for training the data set for recognizing places. The map is divided into local places with the help of spectral clustering. Coarse and fine localization is used for tagging for global localization. Support vector machine is used to train the input data. The support vector machine can result in various decision for inside environment that has multiple local locations the are similar from each other. In order to manage this situation SVM location recognition is mixed with particle filter for global localization. In offline cases, it has localization learning and in online it has coarse to fine.

Wang et al. [15] gave localization which is wireless and free from devices. Location and other activities can be computed by observing the effect on shadow in links around it. Deep learning methods were used for understanding DFLAR. It can get the discriminative feature by signal from wireless network which minimizes the time-consuming properties methods used by users in history and it also determine more realistic discriminative properties. The observation is that this method shows better localization with accuracy 0.85 in indoor environment by considering only 8 nodes. It works better than the traditional methods. It also focuses on localization activity identification and gesture identification which makes wireless network a better infrastructure. The problem with this method is that how to get multi target localization, activity identification and gesture identification and how to improve the accuracy of the system.

Liu and Li [16] proposed methods to handle localization scalability and accuracy of a phone using opportunistic sensing. Location estimation is done by semidefinite programs. The extensive analysis proof the betterment of their approach. It uses the localization of fine grain to solve location aware problems like indoor movement for the blind person, finding virtual reality in games, movement for robots and driving. Adding more anchor nodes will increase the timing of process of management. The model uses the multimodal sensor data to enhance the scalability and efficiency. Liu et al. [18] introduced a method to uniquely merge the two subnetworks under some derived conditions. By going through extensive experiments, they observed that almost all the nodes in the 3D sparse network can be localized by their algorithm. The algorithm also manages the error propagation efficiently. Divide conquer and combine method was used to develop the algorithm.

Fankhauser et al. [19] introduced techniques which includes the drift and uncertainties of the state computation and a noise model for distance sensors grid based maps with lower and upper bounds on confidence were used. A novel method was used to capture the problem of localization drifts for the mobile robots. It computes the elevation map and tries to manage the localization drift. Aron et al. [25] gave a better method of job scheduling using available resources. A resource scheduling method using PSO is developed which does not satisfy all security constraints. Implementation was done using GridSim. Toporkov and Toporkova introduced a hybrid method of heuristics, backfilling and cyclic scheduling [26]. The result of various algorithms and heuristics were analyzed.

In this paper, we consider the resource allocation and provisioning in MCG. The **idea** is as follows. The local schedular uses local heuristics which assign the resources locally [9]. If the resources are available in the near locality then the resources are allotted for the computation. If the resources are not available in locality then we need to consult the global schedular. The global schedular uses the global heuristic to picks the resources globally and then it transfers to the local schedular and finally those resources are distributed by the local schedular [10].

In this paper, we **develop a mathematical framework of our scheduling model**. Two scheduling algorithms namely HGLA and HEFT, Max-Min, Min-Min algorithm are proposed. For the local search of resources, local search heuristics (Min-Min, Max-Min and HEFT) are used. For global search, Genetic algorithm is used. Min-Min, Max-Min, HEFT and GA local search heuristic methods are well used and the explained as baseline for comparison in most of the research paper [10], [21], [25], [26], [37], [39], [41], [42]. The experiment is done via MATLAB and grid sim. The performance of both algorithms is compared. The proposed algorithm HGLA has better performance than the Min-Min, Max-Min and HEFT in terms of throughput and normalized localization ratio. Due to better performance, it can be also used in other paradigms like cloud computing, fog computing, cluster computing, etc. In section 2, mathematical formulation and algorithm of scheduling model are presented. Experimental results are explained in section 3. Finally, conclusion and future scope are explained in section 4.

II. HEURISTIC SEARCH BASED LOCALIZATION IN MOBILE COMPUTATIONAL GRID

A. NETWORK TOPOLOGY AND ASSUMPTIONS

A foundational mathematical background is necessary to develop such resource scheduling model which tries to optimize localization ratio. The basic mathematical analysis of scheduling is given in the literature [1], [2], [5]. Further basic mathematical analysis of localization is given in the literature [3], [7], [9]. By using these mathematical formulations given in this paper is derived. The above stated problem in MCG uses the following assumptions:

- Every machine can perform one sub job at a time.
- Execution time of a sub job of a job is known in advance
- That the offline data is used to analyses the performance of the method.
- The proposed method in limited network and simulation environment.
- The discussed method is analyzed by MATLAB and simulation environment is simulated by JAVA and NS-3.

B. MATHEMATICAL MODELLING

The notations and their description used in this model is mentioned in Table 1. In our model, we consider a machine as primary and others as secondary. The primary machines receive all the jobs and it distributes them to the secondary machines with the help of local and global schedulers. At the very beginning each of the secondary machines declare their computational speed and capability. After this declaration the primary machine distributes the jobs and related small jobs (with the help of local and global schedular) to the secondary computational machine which suits the job requirement and meets the machine capability and machine is also not overloaded. The job $\mathrm{J_K}$ has $\mathrm{n_K}$ sub-jobs. $\mathrm{SJ_{Ik}}$ is the Kth sub-job of $\mathrm{i_{th}}$ job. If $\mathrm{T(SJ_{IK})}$ represents the time of execution of $\mathrm{SJ_{Ik}}$. Then, the following holds true.

$$\mathrm{J}_0 = \sum_{i=0}^{i=n_0} T(S\mathrm{J}_{0\mathrm{i}}) \tag{1}$$

$$\mathrm{J}_1 = \sum_{i=0}^{i=n_1} T(S\mathrm{J}_{1\mathrm{i}}) \tag{2}$$

$$\cdots\cdots\cdots\cdots\cdots\cdots$$

$$\mathrm{J}_n = \sum_{i=0}^{i=n_n} T(S\mathrm{J}_{\mathrm{ni}}) \tag{3}$$

Million instructions per seconds is the speed of mobile compute node processors. Every processor additionally uses particular limit of what is the number of jobs that can be processed by that processor. If we consider j^{th} sub job having size $\mathrm{S_{ij}}$ MIPS, and processing speed of $\mathrm{k^{th}}$ node M is $\mathrm{S_k}$ MIPS then the completion time is determined as [16].

$$\mathrm{T_{ijk}} = \frac{S_{ij}}{S_k} \tag{4}$$

The situation in which a machine has allotted number of sub jobs beyond its capability, then additional sub jobs should be redistributed among those machines which has a smaller number of tasks than its ability. Before redistributing the sub tasks to the underutilized machine, we have to distinguish between the machine which are over-burden and the machines which are underutilized.

If $\mathrm{M_u}$ is the number of underutilized machines the

K_i = total sub jobs allocated on machine $\mathrm{M_i}$ – Power (number of sub jobs) of the processor $\mathrm{M_i}$.

Also, if K_1 represents the total count of unallocated sub jobs (the extra number of sub jobs than the capacity

TABLE 1. Notation their discription and domain value.

Notation	Description	Domain value
J_i	i^{th} job	50•j•14000
SJ_{ik}	k^{th} sub job of j^{th} job	$10 \leq J_{ik} \leq 100$
T_{ijk}	Time to execute j^{th} sub job of i^{th} job on k^{th} machine	10 Sec • T_{ijk} • 25 Sec
S_k	Processing speed of k^{th} machine	101 MIPS • T_{ijk} • 200 MIPS
M_u	Number of underutilized machines	0•M_u•N
K_1	Total number of unallocated jobs	0•K_1 • M
K_2	Total available computational power	0• K_2• M
λ_i	Rate of arrival of jobs on i^{th} machine	1MIPS • λ_i•100 MIPS
μ_i	Mean service rate on i^{th} machine	0 • μ_i•1
U_i	Utilization of i^{th} machine	0• U_i•1
E_{WT}	Expected waiting time	0•E_{WT}<MS
E_{ST}	Expected service time	0• E_{ST}•1
ETC(i, j)	Expected time of completion of i^{th} job on j^{th} machine	1 • ETC(i, j) •MS
δ_{ij}	Boolean variable	$\delta_{ij} \epsilon \{0,1\}$
MS	Makespan	MS
T'_{ij}	Time for completion j^{th} subtask of task i	1Sec •T'_{ij}• MS
K'_i	Number of jobs that can be allotted on i^{th} machine	$1 \leq K'_i \leq$ capacity of machine
R	Largest radio range	1•R•INF
d_{ij}	Shortest distance between nodes S_i and S_j	1•d_{ij}• dimeter of graph
Th_{max}	Maximum throughput	$0 \leq Th_{max} \leq 1$
N_{max}^{task}	Maximum Number of tasks	1•N_{max}^{task}•M
e	Exponential coefficients	e=2.71
t	Ratio of min and max	0•t•1
Z_1	Max of coordinate	0•Z_1•100
Lr	Localization Ratio	0•L_r•1
N	Total number of machines	100•N•1200
M	Total number of jobs	50•M•14000
$T(SJ_{IK})$	$T(SJ_{IK})$ represent the time of execution of SJ_{IK}	10 sec •$T(SJ_{IK})$•25 sec

of machine) then,

$$K_1 = \sum_{i=0}^{i=n} K_i \quad \forall K_i > 0 \tag{5}$$

Total computational power available is

$$K_2 = \sum_{i=0}^{i=n} K'_i \quad \forall K_i < 0 \text{ and } K_i^{'} = |-K_i| \tag{6}$$

By above consideration, three following cases are possible
1. $K_1 = K_2$ redistribution of jobs is done smoothly.
2. $K_1 > K_2$ in this case few jobs are not properly distributed.
3. $K_1 < K_2$ successful allotment of jobs is done in this case.

Utilization of the machine M_i is the ratio of arrival rate λ_i and mean service rate μ_i

$$U_i = \frac{\lambda_i}{\mu_i} \quad \forall i \in N \tag{7}$$

Using the theorem of queuing theory, the average number of jobs arrived at machine M_i is defined as follows:

$$Average\ number\ of\ jobs = \lambda_i \star \mu_i \quad \forall i \in N \tag{8}$$

Expected waiting time of a job at j^{th} machine M_j is

$$E_{WT} = \frac{\lambda_j}{\mu_j(\mu_j - \lambda_j)} \tag{9}$$

Expected service time at the machine is

$$E_{ST} = \frac{1}{\mu_j} \tag{10}$$

Therefore, total time (ETC(i; j)) at the j^{th} node of i^{th} task is given as follows.

$$ETC(i, j) = \sum_{i=1}^{r} [(E_{WT} + E_{ST}) \times \delta_{ji} \times NOI_i] \tag{11}$$

NOI_i is the number of instructions in i^{th} job and δ_{ji} is a Boolean variable with

$$\delta_{ji} = \begin{cases} 0 & \text{if } i^{th} \text{ job is assigned to } j^{th} \text{ machine} \\ 1 & \text{if } i^{th} \text{ job is not assigned to } j^{th} \text{ machine} \end{cases} \tag{12}$$

The total execution time i.e. the makespan of the schedule is given as,

$$MS = \max_{1 \leq 1 \leq R} ETC(I, J) \tag{13}$$

Speedup is the ratio of serial completion time and the makespan of the schedule:

$$Speedup = \frac{Serial\ Execution\ Time}{MS} \tag{14}$$

The objective function is to optimize

$$Normalized\ Speedup = \frac{Speedup}{No\ of\ processor} \tag{15}$$

$$Th_{max} = Normalized\ Sepeedup * N_{tasks}^{max} \tag{16}$$

The unweighted graph is used to define the local connectivity information issued by the radio. The nodes are the MCG compute nodes and the edges can be the radio links. The edges in the shortest path between two nodes is considered as the hop count between two sensor nodes. Then, the distance between S_i and S_j, d_{ij} is less than $R*h_{ij}$, where R is the largest radio range. It is expected that a good estimate can be found if we have the knowledge of local, the average number of neighbors per node.

$$Z_1 = Max\,(x, y) \tag{17}$$

$$t = \frac{Min(x, y)}{Z_1} \tag{18}$$

$$nlocal = lacalization\ parameter = r\ and\ (0, 1) \tag{19}$$

$$Lr = (R \star e - nlocal - Z_1)/e^{1 - nlocal \star \pi \left(cos(t - t\sqrt{1 - t^2})\right)} \tag{20}$$

In this paper, the **objectives are to maximize speed up, throughput and minimize localization ratio**. The proposed model is compared with the state-of-the-art MIN-MIN, MAX-MIN and HEFT by considering the quality of service parameters speed-up, throughput and localization ratio.

C. THE PROPOSED ALGORITHMS

This section proposes two algorithms namely heuristic guided localization in MCG and Min-Min, Max-Min and HEFT guided localization in MCG. The general methodology of both the algorithms are given as follows:

All the jobs having fixed number of sub-jobs submitted by the user is assigned to the primary machine. Primary machines use the identify the secondary machines and assign them the sub-jobs in random fashion. Speed and capacity of each secondary machine is known in advance. The execution time of each sub-job of a job also known in advance. If a certain machine has got number of jobs than its capacity then those sub-jobs are returned back to primary machine. The primary machine finds the machines which are underutilized i.e. the machines which got a smaller number of sub-jobs than their capacity. Rest of the sub-jobs are assigned to the underutilized machines for their execution. Once the sub-jobs are done on the secondary machine its computation is given back to the primary machine where the integration of sub-jobs are done and the total waiting time, turnaround time, makespan of the schedule, speedup and normalized speedup is computed.

In the first algorithm called the "**heuristic guided localization in MCG**", the primary machine uses heuristic based the localization process with the help of local schedular to identify the secondary machine to allocate the sub-jobs. On the contrary, the second algorithm called "**Min-Min, Max-Min and HEFT guided localization in MCG**", the primary machine uses the Min-Min, Max-Min, and HEFT to identify the secondary machine for sub-job execution.

Various data structures are used in order to implement the above proposed algorithms. R_{list} is an array list which includes all the sub-jobs which needs to be redistributed. NMc is the total number of machines available for its execution. Alloc is the two dimensional matrix, if Alloc[i, j] = k it implies j^{th} sub job of i^{th} job executed to k machine. A one-dimensional array containing the count of sub jobs, a machine can complete at a time is CMc. List[i] is a data structure which contains subjobs assigned machine M. SMc is an array used to store the speed of all the machines. Number of jobs and sub-jobs are represented by Mjobs and Msjobs respectively. Tmp is a temporary variable used to store the number of jobs which are overloaded to a specific machine.

Steps of **Heuristic guided localization algorithm in MCG** (HGLA) are given in the following section:

Step 1: Initially all the jobs containing sub-jobs are assigned to primary machine PM.

Step 2: Rest of the R_{List} is initialized to NULL, total turnaround time and total waiting time are initialized to 0.

Step 3: Distribute the sub-jobs randomly to all the secondary machines using local heuristics in local schedular and global heuristic for global schedular.

Step 4: If number of sub-jobs on a secondary machine is larger than its capacity then put the overloaded jobs to the R_{List}.

Step 5: Find the machines which are underutilized.

Step 6: Distributes the sub-jobs from RList to the underutilized machines using local heuristics in local schedular and global heuristic for global schedular.

Step 7: Find the normalized speedup, throughput and localization ratio by the formulae given in equation (11), (12) and (16).

The pseudo code of the HGLA algorithm is given as follows:

Algorithm 1 HGLA Algorithm

```
Output: Localization ratio in MCG.
1. Start
2. All the jobs are initially given to PM
3. T = W = 0
4. RList ← NULL
5. While RList ≠ NULL
      For each i ← 1 to NMc do
          Alloc[i, j] ← random.math    × NMc
          construct a list List[i] for subjobs
          overloaded to machine M[i]
      For each i← 0 to NMc do
          If List[i]. Size = CMc[i] do
             For each j ← CMc[i] to EList[i] do
                 Add the subjobs to the RList[i]

6.    For each i ← 0 to NMc do
          If List[i]. Size ←CMc[i] do
          include the primary subjobs to the
          rest_of_List[i] to List[i].
          T ←T + Turnaround time of subjob that are
          already completed.
          W ← W + The total time consumed
          to complete the pending task.
          Apply random heuristic with normal
          distribution to compute localization ratio,
          speed up and throughput using the
          formulae defined above.

7. Stop
```

Steps of **Min-Min, Max-Min and HEFT guided in MCG** are given in the following section:

Step 1: Initially all the jobs containing sub-jobs are assigned to primary machine PM.

Step 2: Rest of the R_{List} is initialized to NULL, total turnaround time and total waiting time are initialized to 0.

Step 3: Distribute the sub-jobs randomly to all the secondary machines using Min-Min, Max-Min and HEFT.

Step 4: If number of sub-jobs on a secondary machine is larger than its capacity then put the overloaded jobs to the R_{List}.

Step 5: Find the machines which are underutilized.

Step 6: Distributes the sub-jobs from R_{List} to the underutilized machines using Min-Min, Max-Min and HEFT.

Step 7: Find the normalized speedup, throughput and localization ratio by the formulae given in equation (11), (12) and (16).

The pseudo code for "**Min-Min, Max-Min and HEFT guided localization in MCG**" is given as follows:

Algorithm 2 HEFT, Max-Min, Min-Min Algorithm

```
Output: Localization ratio in MCG
1.  Start
2   All the jobs are initially given to PM
3.  T ← 0
4.  W ← 0
5.  RList ← NULL
6.  While RList ≠ NULL do
        For i ← 0 to Nsjob[i] do
            include all these subjobs to List A
        If A.Size ≠ NULL do
7.              For i ← 0 to NMc do
                    Tmp[i] ← 0
        For i ← 0 to NMc do
            If Njobs allotted to machine M[i] >
            CMc_M[i]
                Tmp[i] ← Tmp[i]
                + Machine_Jobs[i]
8.      Find the best job in RList by Min-min, Max- Min,
        HEFT and allot that job to machine M
9.      T ← T + Time to complete jobs already in Queue
10.     W ← W + Time to complete pending tasks.
11.     Find the Localization ratio.
12. Stop
```

The proposed model HGLA (Heuristic Guided Localization Algorithm) is compared with the MIN-MIN, MAX-MIN, and HEFT by considering the quality of service parameters speed-up and localization, MAX-MIN, and HEFT. Next segment talks about the conducted experiments and observed results.

D. RESEARCH GAP AND CONTRIBUTION OF PROPOSED STUDY

The following research gap has been identified from the literature:

1. Most of the simple algorithm for resource provisioning algorithms suffers from the job starvation problems.

2. All the scheduling algorithms discussed in literature survey only focused on the single site. They do not say anything about the multisite job distribution.

3. The Grid usually under the consideration is static. What happens if it also includes some mobile nodes?

4. The nature of the grid in all resource provisioning algorithms is homogeneous. If it has heterogeneous collection of resources then how to manage and perform the resource scheduling?

5.All the resource scheduling doesn't consider the case when a system is over loaded then how to perform the load balancing of job.

6. Mostly algorithms are job independent algorithms.

The proposed model has following contributions to the research field:

1. HGLA algorithm eliminate the problem of starvation problem by uniformly distributing the sub-jobs to compute nodes.

2. HGLA algorithm considers the job distribution in multiple sites among heterogenous resources.

3. The nature of the grid under consideration in static and mobile both.

4.The redistribution of job is done in such a way so that load balancing is also performed.

III. RESULT AND DISCUSSION

The programming of proposed model is done on Eclipse with the coordination with Gridsim [39]. The analysis of performance is discussed of the model. The size of jobs and sub-jobs determine the convergence of solution. The parameters used for simulation in the experimentation are Input parameter values given Table 2. Experiments are accompanied 40 times with 1TB secondary memory and 16 GB RAM and mean is evaluated for every observation.

A. EXPERIMENT 1: EXPERIMENT OF LOCALIZATION WITH MOBILE AGENT AND WITHOUT MOBILE AGENT

Total nodes under consideration is 100 along with parameter given in Table 2. Parameter localization of location position is given in Figure 4. it is viewed that after a location is changed in rapid way for some tasks but in the case of HGLA the CMG has better location than the MIN-MIN, MAX-MIN and HEFT.

B. EXPERIMENT 2: EXPERIMENT OF NORMALIZED SPEEDUP AND COMPARISON WITH THE STATE OF ARTS

1200 nodes are considered to find the performance of the model. HGLA, MIN-MIN, MAX-MIN, and HEFT are compared in Figure 5 with respect to the parameter normalized speedup. The normalized speedup increases in a rapid way but in the case of HGLA, the CMG has better-normalized speedup than the MIN-MIN, MAX-MIN and HEFT. The values of normalized speedup are written in the corresponding bar in the given Figure 5. The value of normalized speedup should be closer to 1 in the case of the speed up because this is normalized speed up between 0 and 1.

C. EXPERIMENT 3: EXPERIMENT OF LOCALIZATION RATIO AND COMPARISON WITH THE STATE OF ARTS

1200 nodes are considered as the input to observe the localization ratio. HGLA, MIN-MIN, HEFT and MAX-MIN are

TABLE 2. Parameters, there values and simulation environment.

Parameter	Value
Number of nodes	100 - 1200
Tasks count	50 - 14000
No of sub tasks in a task	10 - 100
Rate of arrival	1 - 100 MIPS
Speed of execution	101 - 200 MIPS
Task size	2000 - 5000 MI
Subtask Size	20 - 100MI
Memory size of a node	4 – 32 GB
Service rate	0-1
Size of Cache memory	2 – 128 KB
No of registers	16 - 256
No of bus	2 - 64
Speed of Local schedular	2 – 2048 tasks
Speed of Global schedular	1 – 512 tasks
Simulation languages	Java, MATLAB
Tool	Eclipse, Gridsim

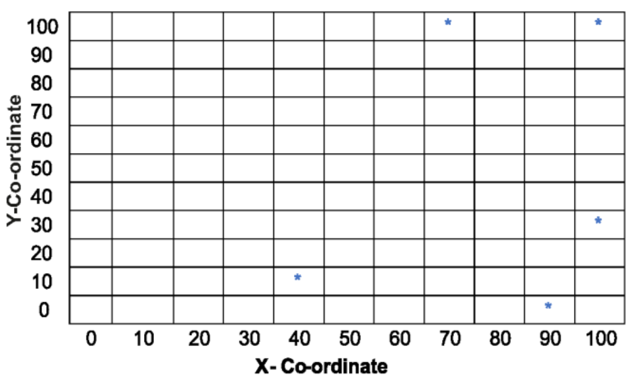

FIGURE 4. Localization ratio observation.

compared in figure 6 with respect to the parameter localization ratio. Figure 6 reveals that after a certain iteration and count of jobs the localization ratio decreases in a rapid way. In the case of HGLA for CMG, has better localization ratio than the MIN-MIN, MAX-MIN and HEFT which is the state of the art. The values of localization ratio are written in the corresponding bar in the given Figure 6. The value of localization ratio should be closer to 0.

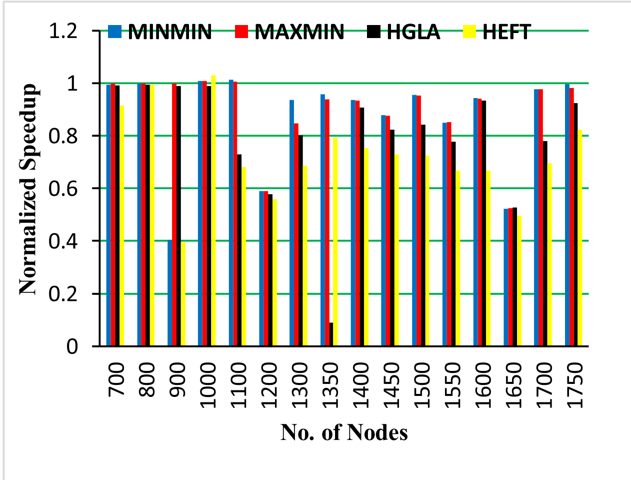

FIGURE 5. Normalized speedup-based comparison between HGLA, MIN-MIN, HEFT and MAX-MIN.

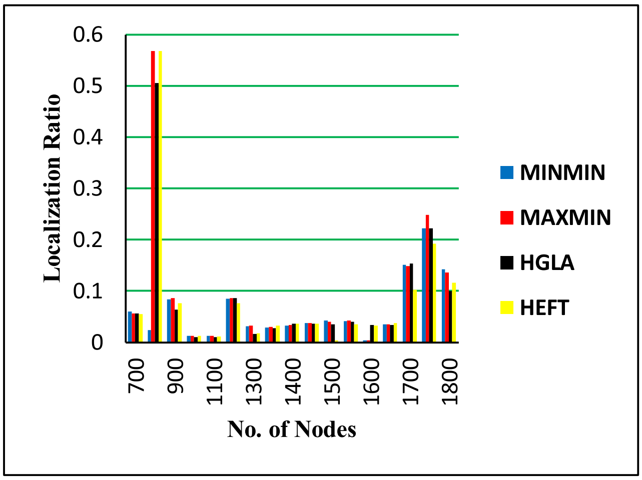

FIGURE 6. Localization Ratio based comparison between HGLA, MIN-MIN, HEFT and MAX-MIN.

D. EXPERIMENT 4: EXPERIMENT OF NORMALIZED SPEEDUP AND COMPARISON WITH THE STATE OF ARTS

1200 nodes are considered to find the performance of the model. HGLA, MIN-MIN, MAX-MIN, and HEFT are compared in Figure 7 with respect to the normalized speedup. The normalized speedup increases in a rapid way but in the case of HGLA, the CMG has better-normalized speedup than the MIN-MIN, MAX-MIN and HEFT. The values of normalized speedup are written in the corresponding bar in the given Figure 7. The value of normalized speedup should be closer to 1 in the case of the speed up because this is normalized speed up between 0 and 1.

E. EXPERIMENT 5: EXPERIMENT OF LOCALIZATION RATIO AND COMPARISON WITH THE STATE OF ARTS

HGLA, MIN-MIN, HEFT and MAX-MIN are compared in Figure 8 with respect to the parameter localization ratio using 700 to 1800 nodes. Figure 8 reveals that after a

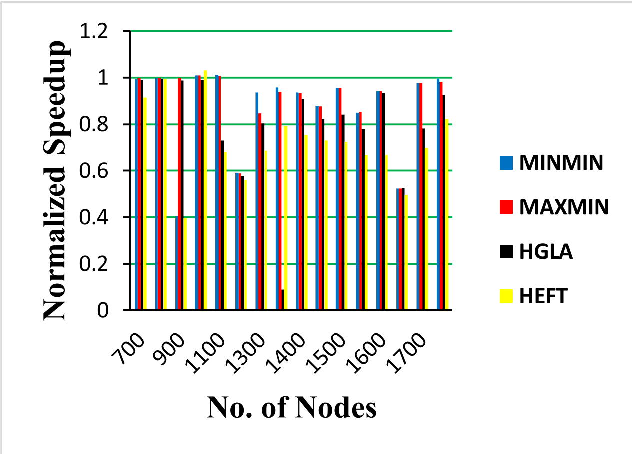

FIGURE 7. Normalized based speedup comparison between HGLA, Min-Min, HEFT and Max-Min.

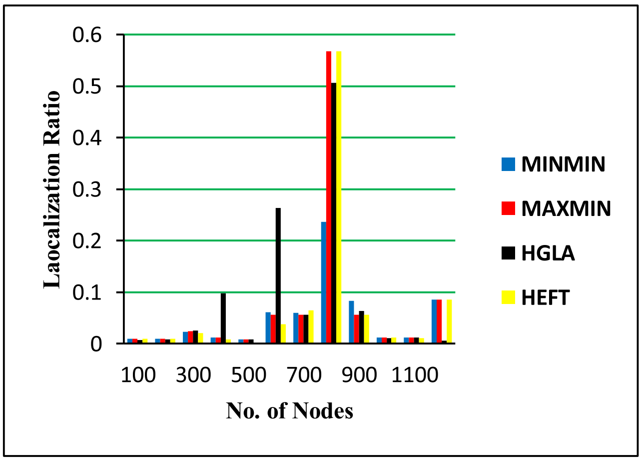

FIGURE 8. Localization Ratio based comparison between HGLA, MIN-MIN, HEFT and MAX-MIN.

certain some iteration and count of jobs the localization ratio decreases in a rapid way and in the case of HGLA the CMG has better localization ratio than the MIN-MIN, MAX-MIN and HEFT which is the state of the art. The values of localization ratio are written in the corresponding bar in the figure. The value of localization ratio should be closer to 0. It is analysed that if number of nodes increases then localization ratio initially increases after certain interval it start decreasing then again increasing. The localization ratio is minimum in nodes between 1000–1100 which is optimized design in case of localization ratio as one objective and considered as one of the quality service parameters in the CMG.

F. EXPERIMENT 6: EXPERIMENT OF THROUGHPUT AND COMPARISON WITH THE STATE OF ART WHEN NUMBER OF NODES ARE FROM 100 TO 1200

HGLA, MIN-MIN, HEFT and MAX-MIN are compared in Figure 9 with respect to the parameter throughput using 100 to 1200 nodes. Figure 9 reveals that after a certain few

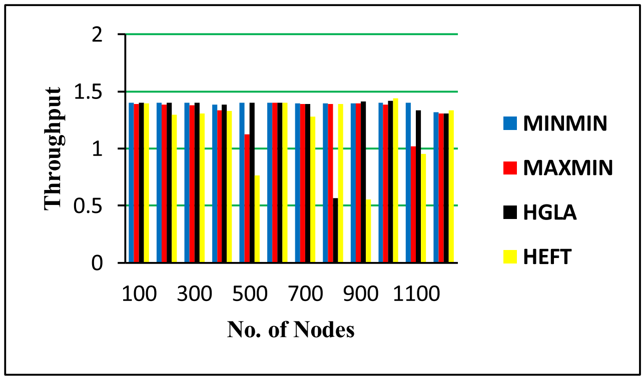

FIGURE 9. Throughput based comparison between HGLA, MIN-MIN, HEFT and MAX-MIN.

iteration and number of jobs the throughput increases in a rapid way but in the case of HGLA the CMG has better throughput than the MIN-MIN, MAX-MIN and HEFT which is the state of the art. The values throughput is written in the corresponding bar in the figure. The value of throughput should be closer to 1.4 thousand per tasks. From figure 9, it is analysed that if number of nodes increases then throughput initially increases after certain interval it start decreasing then again increasing. The throughput is maximum in nodes between 1000–1100 which is optimized design in case of throughput as one objective and considered as one of the quality service parameters in the CMG.

G. EXPERIMENT 7: EXPERIMENT OF THROUGHPUT AND COMPARISON WITH THE STATE OF ART WHEN NUMBER OF NODES ARE FROM 100 TO 1200

HGLA, MIN-MIN, HEFT and MAX-MIN are compared in figure 10 with respect to the parameter throughput using 100 to 1200 nodes. Figure 10 reveals that after a certain number of iteration and number of jobs the throughput increases in a rapid way but in the case of HGLA the CMG has better throughput than the MIN-MIN, MAX-MIN and HEFT which is the state of the art. The values throughput is written in the corresponding bar in the given figure. The value of throughput should be closer to 1.4 thousand per tasks. From figure 10 it is analysed that if number of nodes increases then throughput initially increases after certain interval it start decreasing then again increasing. The throughput maximum in nodes 800 which is optimized design in case of throughput as one objective and considered as one of the quality service parameters in the CMG.

H. EXPERIMENT 8: EXPERIMENT OF THROUGHPUT AND COMPARISON WITH THE STATE OF ART WHEN NUMBER OF NODES ARE FROM 700 TO 1800

HGLA, MIN-MIN, HEFT and MAX-MIN are compared in Figure 11 with respect to the parameter throughput. Figure 11 reveals that after some iteration and number of jobs the throughput increases in a rapid way but in the case of HGLA the CMG has better throughput than the MIN-MIN,

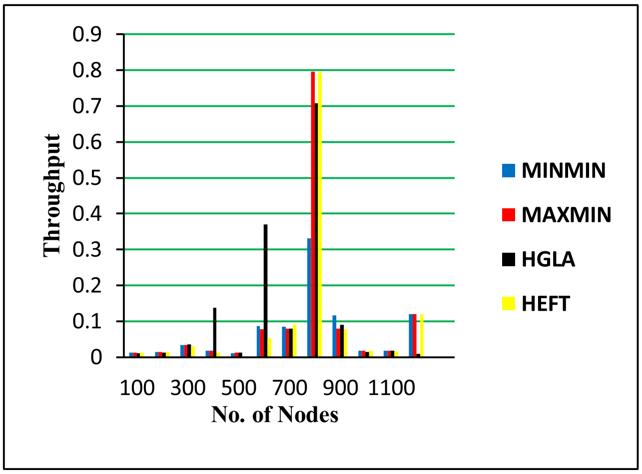

FIGURE 10. **Throughput based comparison between HGLA, MIN-MIN, HEFT and MAX-MIN.**

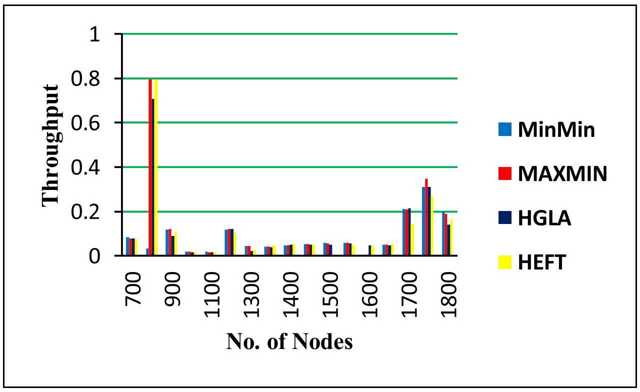

FIGURE 11. **Throughput based comparison between HGLA, MIN-MIN, HEFT and MAX-MIN.**

MAX-MIN and HEFT which is the state of the art. The values throughput is written in the corresponding bar in the given Figure 11. The value of throughput should be closer to 1.4 thousand per tasks. From Figure 11, it is analysed that if number of nodes increases then throughput initially increases after certain interval it start decreasing then again increasing. The throughput maximum in nodes 800 which is optimized design in case of throughput as one objective and considered as one of the quality service parameters in the CMG.

I. EXPERIMENT 9: EXPERIMENT OF THROUGHPUT AND COMPARISON WITH THE STATE OF ART WHEN NUMBER OF NODES ARE FROM 700 TO 1750

HGLA, MIN-MIN, HEFT and MAX-MIN are compared in Figure 12 with respect to the parameter throughput using 700 to 1750 nodes. Figure 12 reveals that after some iteration and number of jobs the throughput increases in a rapid way but in the case of HGLA the CMG has better throughput than the MIN-MIN, MAX-MIN and HEFT which is the state of the arts. The values throughput is written in the corresponding bar in the given Figure 12. The value of throughput should be closer to 1.4 thousand per tasks. From Figure 12, it is analysed that if number of nodes increases then throughput initially increases after certain interval it start decreasing then again increasing. The throughput maximum in nodes between 800-1100 which is optimized design in case of throughput as one objective and considered as one of the quality service parameters in the CMG.

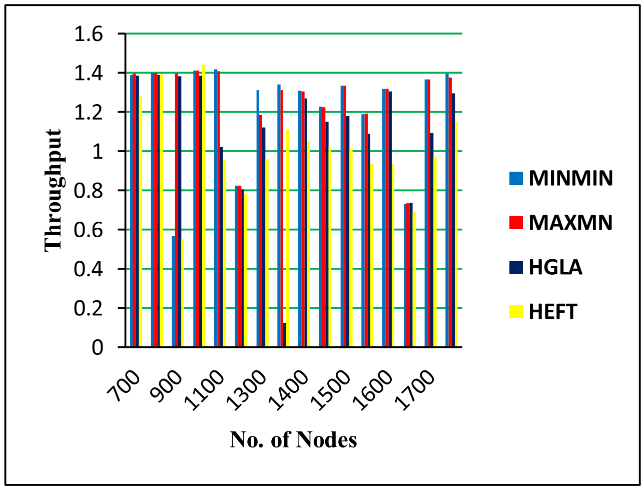

FIGURE 12. **Throughput based comparison between HGLA, MIN-MIN, HEFT and MAX-MIN.**

Finally, we can conclude in terms of result analysis we had taken three QoS parameters normalized speedup, localization ratio and throughput. The redistribution of jobs has been done carefully so that the no compute nodes is overloaded. HGLA, MIN-MIN, HEFT and MAX-MIN are compared in figures 2–9 with respect to the above 3 parameters.

The normalized speedup, localization ratio and throughput are optimum in nodes between 800–1100 which is optimized design. Initially, 100–1100 nodes are considered for the observation and it is found that in most of the cases the proposed HGLA algorithm gives better normalized throughput, speedup and localization ratio. Next, 700–1800 nodes are considered, and once again the HGLA produces good performance in comparison to the Min-Min, Max-Min and HEFT algorithm. Overall proposed algorithm outperforms the local heuristics in most of the cases.

IV. CONCLUSION AND FUTURE DIRECTIONS

In this paper, we have proposed a novel approach heuristic search-based localization in computational grid. We have introduced algorithms for localization based on heuristics and localization using the traditional methods like MIN-MIN, MAX-MIN, and HEFT. Simulation is performed for different input sizes to judge the performance of the discussed model.

It has been found that in almost all the inputs, it performs good and able to justify the optimization of quality of service parameters. The execution time increase if we scale up the number of job and keeps the number of compute processors fixed. The proposed model is compared with the MIN-MIN, MAX-MIN and HEFT by considering the quality of service parameters speed-up, throughput and localization ratio. The results reveal that the proposed model gives better performance than MIN-MIN, MAX-MIN and HEFT.

There are few limitations of the proposed HGLA model, if the node performing the task becomes unavailable due to some faults. The basic assumption of HGLA is that jobs, sub-jobs and their execution time are known in advance. If the jobs are coming online and their execution time is not known in advance then how to manage these scenarios. The network under consideration is limited, if it is unlimited then how it will affect the scheduling process?

Even though significant progress has been made in scheduling in mobile computational grid environments, not much of progress has been made in terms of energy savings in the grid system. At times the efficient resources are found to be over utilized and therefore lot of heat is generated, which in turn leads to a huge amount of cost being spent on cooling. Therefore, this energy consumption, which is a huge wastage, can lead to the decreased cost performance of the large potential of good balanced schedules in grid. Therefore, this now becomes a mandatory issue in the grid system. Therefore, for future works this research can be extended to energy aware conscious scheduling for more complicated experiments with additional objective

In the future, other quality of services parameters like load balancing, security, reliability, availability, mobility, fault tolerance etc. can be addressed for their optimization using heuristic localization method. This concept can also be used in current research area like cloud computing, fog computing cluster computing etc.

CONFLICTS OF INTEREST

The authors declare that they have no competing interests.

REFERENCES

[1] B. Allcock, J. Bester, J. Bresnahan, A. L. Chervenak, I. Foster, C. Kesselman, S. Meder, V. Nefedova, D. Quesnel, and S. Tuecke, "Data management and transfer in high-performance computational grid environments," *Parallel Comput.*, vol. 28, no. 5, pp. 749–771, May 2002.

[2] F. Berman, G. Fox, and A. H. G. Computing, *Making the Global Infrastructure a Reality*. Hoboken, NJ, USA: Wiley, 2003.

[3] C. Kesselman and I. Foster, *The Grid: Blueprint for a New Computing Infrastructure*. Burlington, MA, USA: Morgan Kaufmann Publishers, 1998.

[4] I. Foster, C. Kesselman, and S. Tuecke, "The anatomy of the grid: Enabling scalable virtual organizations," *Int. J. High Perform. Comput. Appl.*, vol. 15, no. 3, pp. 200–222, Aug. 2001.

[5] M. Agents, "The future of the Internet, david kotz, robert," Tech. Rep. 15, 1999.

[6] F. Passerini and A. M. Tonello, "Smart grid monitoring using power line modems: Effect of anomalies on signal propagation," *IEEE Access*, vol. 7, pp. 27302–27312, 2019.

[7] M. T. Younis and S. Yang, "Hybrid meta-heuristic algorithms for independent job scheduling in grid computing," *Appl. Soft Comput.*, vol. 72, pp. 498–517, Nov. 2018.

[8] K. Wang, H. Li, S. Maharjan, Y. Zhang, and S. Guo, "Green energy scheduling for demand side management in the smart grid," *IEEE Trans. Green Commun. Netw.*, vol. 2, no. 2, pp. 596–611, Jun. 2018.

[9] I. U. Khan, X. Ma, C. J. Taylor, N. Javaid, and K. Gamage, "Heuristic algorithm based dynamic scheduling model of home appliances in smart grid," in *Proc. 24th Int. Conf. Automat. Comput.*, Sep. 2018, pp. 1–6.

[10] S. Memeti, S. Pllana, A. Binotto, J. Kołodziej, and I. Brandic, "A review of machine learning and meta-heuristic methods for scheduling parallel computing systems," in *Proc. Int. Conf. Learn. Optim. Algorithms, Theory Appl.*, May 2018, p. 5.

[11] G. Han, J. Jiang, C. Zhang, T. Q. Duong, M. Guizani, and G. Karagiannidis, "A survey on mobile anchor node assisted localization in wireless sensor networks," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 2220–2243, 3rd Quart., 2016.

[12] C. Wen, S. Pan, C. Wang, and J. Li, "An indoor backpack system for 2-D and 3-D mapping of building interiors," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 7, pp. 992–996, Jul. 2016.

[13] F. Yucel and E. Bulut, "Clustered crowd GPS for privacy valuing active localization," *IEEE Access*, vol. 6, pp. 23213–23221, 2018.

[14] S. Park and K. S. Roh, "Coarse-to-fine localization for a mobile robot based on place learning with a 2-D range scan," *IEEE Trans. Robot.*, vol. 32, no. 3, pp. 528–544, Jun. 2016.

[15] J. Wang, X. Zhang, Q. Gao, H. Yue, and H. Wang, "Device-free wireless localization and activity recognition: A deep learning approach," *IEEE Trans. Veh. Technol.*, vol. 66, no. 7, pp. 6258–6267, Jul. 2017.

[16] K. Liu and X. Li, "Enhancing localization scalability and accuracy via opportunistic sensing," *IEEE/ACM Trans. Netw.*, vol. 26, no. 3, pp. 1517–1530, Jun. 2018.

[17] J. Jung, S.-M. Lee, and H. Myung, "Indoor mobile robot localization and mapping based on ambient magnetic fields and aiding radio sources," *IEEE Trans. Instrum. Meas.*, vol. 64, no. 7, pp. 1922–1934, Jul. 2015.

[18] A. Y. Hata, F. T. Ramos, and D. F. Wolf, "Monte Carlo localization on Gaussian process occupancy maps for urban environments," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 99, pp. 2893–2902, Sep. 2018.

[19] P. Fankhauser, M. Bloesch, and M. Hutter, "Probabilistic terrain mapping for mobile robots with uncertain localization," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3019–3026, Oct. 2018.

[20] X. Liu, J. Yin, S. Zhang, B. Ding, S. Guo, and K. Wang, "Range-based localization for sparse 3-D sensor networks," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 753–764, Feb. 2019.

[21] S. Prakash and D. P. Vidyarthi, "Observations on effect of IPC in GA based scheduling on computational grid," *Int. J. Grid High Perform. Comput. (IJGHPC)*, vol. 4, no. 1, pp. 67–80, Jan. 2012.

[22] T. Zhang, X. Ji, Z. Zhuang, and W. Xu, "JamCatcher: A mobile jammer localization scheme for advanced metering infrastructure in smart grid," *Sensors*, vol. 19, no. 4, p. 909, Feb. 2019.

[23] N. Minar, K. H. Kramer, and P. Maes, "Cooperating mobile agents for mapping networks," in *Proc. 1st Hungarian Nat. Conf. Agent Based Comput.*, 1998, pp. 1–12.

[24] D. P. Dora, S. Kumar, O. Kaiwartya, and S. Prakash, "Secured time stable geocast (S-TSG) routing for VANETs," in *Proc. 3rd Int. Conf. Adv. Comput., Netw. Informat.* New Delhi, India: Springer, 2016, pp. 161–167.

[25] R. Aron, I. Chana, and A. Abraham, "A hyper-heuristic approach for resource provisioning-based scheduling in grid environment," *J. Supercomput.*, vol. 71, no. 4, pp. 1427–1450, Apr. 2015.

[26] V. Toporkov, A. Toporkova, A. Tselishchev, D. Yemelyanov, and P. Potekhin, "Heuristic strategies for preference-based scheduling in virtual organizations of utility grids," *J. Ambient Intell. Humanized Comput.*, vol. 6, no. 6, pp. 733–740, Dec. 2015.

[27] K. Li, "Job scheduling for grid computing on metacomputers," in *Proc. 19th IEEE Int. Parallel Distrib. Process. Symp.*, Apr. 2005, p. 8.

[28] S. Prakash and D. P. Vidyarthi, "A hybrid GABFO scheduling for optimal makespan in computational grid," *Int. J. Appl. Evol. Comput.*, vol. 5, no. 3, pp. 57–83, Jul. 2014.

[29] D. Fox, W. Burgard, and S. Thrun, "Active Markov localization for mobile robots," *Robot. Auto. Syst.*, vol. 25, nos. 3–4, pp. 195–207, Nov. 1998.

[30] A. C. Arpaci-Dusseau, D. E. Culler, and A. M. Mainwaring, "Scheduling with implicit information in distributed systems," in *Proc. ACM SIGMETRICS Joint Int. Conf. Meas. Modeling Comput. Syst.*, vol. 26, Jun. 1998, pp. 233–243.

[31] I. Foster and C. Kesselman, *The Grid 2: Blueprint for a New Computing Infrastructure*. Amsterdam, The Netherlands: Elsevier, 2003.

[32] D. C. Marinescu, "Reflections on qualitative attributes of mobile agents for computational, data, and service grids," in *Proc. 1st IEEE/ACM Int. Symp. Cluster Comput. Grid*, May 2001, pp. 442–449.

[33] R. Rajachandrasekar, R. Nagarajan, G. Sridhar, and G. Sumathi, "Job submission to grid using mobile device interface," in *Proc. World Congr. Nature Biologically Inspired Comput. (NaBIC)*, Dec. 2009, pp. 170–174.

[34] G. Sumathi and N. Gopalan, "Status monitoring system for heterogeneous grid environment," in *Proc. Int. Conf. Adv. Comput. Commun.*, Dec. 2006, pp. 102–106.
[35] O. F. Rana and K. Stout, "What is scalability in multi-agent systems," in *Proc. 4th Int. Conf. Auto. Agents*, Jun. 2000, pp. 56–63.
[36] S. Prakash and D. P. Vidyarthi, "Load balancing in computational grid using genetic algorithm," *Adv. Comput.*, vol. 1, no. 1, pp. 8–17, 2011.
[37] J. M. Schopf and B. Nitzberg, "Grids: The top ten questions," *Sci. Program.*, vol. 10, no. 2, pp. 103–111, Jul. 2002.
[38] M. Baker, R. Buyya, and D. Laforenza, "The grid: International efforts in global computing," in *Proc. Int. Conf. Adv. Infrastruct. Electron. Bus., Sci., Educ. Internet*, Jul. 2000, pp. 1–11.
[39] F. Dong and S. G. Akl, "Scheduling algorithms for grid computing: State of the art and open problems," Tech. Rep., 2006.
[40] N. Fernando, S. W. Loke, and W. Rahayu, "Mobile cloud computing: A survey," *Future Generat. Comput. Syst.*, vol. 29, no. 1, pp. 84–106, 2013.
[41] P. Kokkinos, K. Christodoulopoulos, A. Kretsis, and E. Varvarigos, "Data consolidation: A task scheduling and data migration technique for grid networks," in *Proc. 8th IEEE Int. Symp. Cluster Comput. Grid (CCGRID)*, May 2008, pp. 722–727.
[42] G. Juve and E. Deelman, "Resource provisioning options for large-scale scientific workflows," in *Proc. IEEE 4th Int. Conf. eSci.*, Dec. 2008, pp. 608–613.
[43] S. Prakash and D. P. Vidyarthi, "A novel scheduling model for computational grid using quantum genetic algorithm," *J. Supercomput.*, vol. 65, no. 2, pp. 742–770, Aug. 2013.
[44] S. Prakash and D. P. Vidyarthi, "Maximizing availability for task scheduling in computational grid using genetic algorithm," *Concurrency Comput., Pract. Exper.*, vol. 27, no. 1, pp. 193–210, Jan. 2015.
[45] L. H. Son, S. Jha, R. Kumar, J. M. Chatterjee, and M. Khari, "Collaborative handshaking approaches between Internet of computing and Internet of Things towards a Smart World: A review from 2009–2017," *Telecommun. Syst.*, vol. 70, no. 4, pp. 617–634, Apr. 2019. doi: 10.1007/s11235-018-0481-x.
[46] Y. H. Robinson, E. G. Julie, K. Saravanan, R. Kumar, and L. H. Son, "FD-AOMDV: Fault-tolerant disjoint ad-hoc on-demand multipath distance vector routing algorithm in mobile ad-hoc networks," *J. Ambient Intell. Humanized Comput.*, Nov. 2019, pp. 1–18. doi: 10.1007/s12652-018-1126-3.
[47] R. Kapoor, R. Gupta, L. H. Son, S. Jha, and R. Kumar, "Adaptive technique with cross correlation for lowering signal-to-noise ratio wall in sensor networks," *Wireless Pers. Commun.*, vol. 105, no. 3, pp. 787–802, Apr. 2019. doi: 10.1007/s11277-019-06121-7.
[48] N. T. Tam, D. T. Hai, L. H. Son, and L. T. Vinh, "Improving lifetime and network connections of 3D wireless sensor networks based on fuzzy clustering and particle swarm optimization," *Wireless Netw.*, vol. 24, no. 5, pp. 1477–1490, 2018.
[49] K. Saravanan, E. Anusuya, R. Kumar, and L. H. Son, "Real-time water quality monitoring using Internet of Things in SCADA," *Environ. Monitor. Assessment*, vol. 190, pp. 556–572, Sep. 2018.
[50] L. H. Son and P. H. Thong, "Soft computing methods for WiMax network planning on 3D geographical information systems," *J. Comput. Syst. Sci.*, vol. 83, no. 1, pp. 159–179, 2017.
[51] D. T. Hai, L. H. Son, and L. T. Vinh, "Novel fuzzy clustering scheme for 3D wireless sensor networks," *Appl. Soft Comput.*, vol. 54, pp. 141–149, May 2017.
[52] C. Leung, D. Taniar, J. W. Rahayu, and S. Goel, *High-Performance Parallel Database Processing and Grid Databases*. Hoboken, NJ, USA: Wiley, 2008.
[53] A. B. Waluyo, B. Srinivasan, and D. Taniar, "Research in mobile database query optimization and processing," *Mobile Inf. Syst.*, vol. 1, no. 4, pp. 225–252, Nov. 2005.

DINESH PRASAD SAHU received the master's degree in computer application from Jawaharlal Nehru University (JNU), New Delhi, India, and the M.Tech. degree in computer science and engineering from the School of Computer and Systems Sciences, JNU, in 2011, where he is currently pursuing the Ph.D. degree in computer science from the School of Computer and Systems Sciences. His primary research interests are in distributed, grid, soft, and cloud computing.

KARAN SINGH received the degree in computer science and engineering from the Kamala Nehru Institute of Technology, Sultanpur, India, and the M.Tech. and Ph.D. degrees in computer science and engineering from the Motilal Nehru National Institute of Technology, India. He was with the Gautam Buddha University, India. He is currently with the School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi. His primary research interests include computer network, network security, multicast communication, the IoT, and body area network. He is the supervisor of many research scholars. He is an Editorial Board Member of the *Journal of Communications and Network* (CN), USA. He has published over 70 research papers in refereed journals and good conferences. He has organized the workshops, conference sessions, and trainings. He was the General Chair of the international conference (Qshine 2013) at Gautam Buddha University, India. Recently, he has organized a conference ICCCS 2018 at the Dronacharya College of Engineering, Gurgaon, and a special session at the 2nd ICGCET 2018, Denmark. He is a Reviewer of Springer, Taylor & Francis, Elsevier Journals, and the IEEE transactions.

MANISHA MANJU received the degree in computer science and engineering from the Kamla Nehru Institute of Technology (KNIT), Sultanpur, India, the M.Tech. degree in computer science and engineering from NIT Jalandhar, India, and the Ph.D. degree in computer science and engineering from Gautam Buddha University, India. She was with Gautam Buddha University, India. She is currently with the Department of Computer Engineering, G B Pant Engineering College, New Delhi. Her primary research interests are in computer networks, network security, multicast communication, and object-oriented programming. She is a Life Member of the CSI, India.

DAVID TANIAR received the bachelor's, master's, and Ph.D. degrees all in computer science, specializing in databases. His research areas are in big data processing, data warehousing, and mobile and spatial query processing. He has published a book on the *High-Performance Parallel Database Processing* (Wiley, 2008). He has also published over 200 journal papers. He is a regular keynote speaker at an international conference, delivering lectures and speeches on big data. He is a founding Editor-in-Chief of the *International Journal of Web and Grid Services*, and the *International Journal of Data Warehousing and Mining*. He is currently an Associate Professor with Monash University, Australia.

LE MINH TUAN received the bachelor's degree in mathematics and informatics from Thang Long University and the master's degree in computer science from Le Qui Don Technical University, Vietnam, in 2004 and 2008, respectively. He is currently pursuing the Ph.D. degree with the Vietnam National University, Hanoi. He is a Lecturer with the Hanoi University of Home Affairs, Hanoi, Vietnam. His research interests include data mining, machine learning, artificial intelligence, and fuzzy logic.

LE HOANG SON received the Ph.D. degree in mathematics–informatics from the VNU University of Science, Vietnam National University (VNU), in conjunction with the Politecnico di Milano University, Italy, in 2013.

He has been promoted to Associate Professor in information technology at VNU, since 2017. He was a Senior Researcher and the Vice Director of the Center for High-Performance Computing, VNU University of Science, Vietnam National University, from 2007 to 2018. Since 2018, he has been a Senior Researcher with the Department of Multimedia and Virtual Reality, VNU Information Technology Institute. His major field of interests include artificial intelligence, data mining, soft computing, fuzzy computing, fuzzy recommender systems, and geographic information systems. He is a member of the Vietnam Journalists Association, the International Association of Computer Science and Information Technology, the Vietnam Society for Applications of Mathematics, and the Key Laboratory of Geotechnical Engineering and Artificial Intelligence with the University of Transport Technology. He is an Associate Editor of the *Journal of Intelligent and Fuzzy Systems* (SCIE), IEEE ACCESS (SCIE), the *Data Technologies and Applications* (SCIE), the *International Journal of Data Warehousing and Mining* (SCIE), the *Neutrosophic Sets and Systems* (ESCI), the *Vietnam Research and Development on Information and Communication Technology*, the *VNU Journal of Science: Computer Science and Communication Engineering*, and the *Frontiers in Artificial Intelligence*. He serves as the Editorial Board of the *Applied Soft Computing* (SCIE), PLOS ONE (SCIE), the *International Journal of Ambient Computing and Intelligence* (ESCI), and the *Vietnam Journal of Computer Science and Cybernetics*. He is the Guest Editor of several special issues at the *International Journal of Uncertainty, Fuzziness, and Knowledge-Based Systems* (SCIE).

MOHAMED ABDEL-BASSET received the B.Sc., M.Sc., and Ph.D. degrees in information systems and technology from the Faculty of Computers and Informatics, Zagazig University, Egypt. His current research interests include optimization, operations research, data mining, computational intelligence, applied statistics, decision support systems, robust optimization, engineering optimization, multi-objective optimization, swarm intelligence, evolutionary algorithms, and artificial neural networks. He is working on the application of multi-objective and robust meta-heuristic optimization techniques. He is also an/a Editor/Reviewer in different international journals and conferences. He has published over 150 articles in international journals and conference proceedings. He holds the Program Chair in many conferences in the fields of decision-making analysis, big data, optimization, complexity, and the Internet of Things, as well as editorial collaboration in some journals of high impact.

HOANG VIET LONG received the Ph.D. diploma degree in computer science from the Hanoi University of Science and Technology, in 2011, where he defended his thesis in the fuzzy and soft computing field. He is currently the Head of the Faculty of Information Technology, People's Police University of Technology and Logistics, Bac Ninh, Vietnam. He is currently a Researcher with the Institute for Computational Science, Ton Duc Thang University, Ho Chi Minh City, Vietnam. He has been promoted to Associate Professor in information technology, since 2017. Recently, he has been concerning in cybersecurity, machine learning, bitcoin, and blockchain, and has published over 20 papers in ISI-covered journal.