IJBPAS, July, 2015, 4(7): 4336-4348

ISSN: 2277–4998

International Journal of Biology, Pharmacy and Allied Sciences (IJBPAS)

'A Bridge Between Laboratory and Reader'

www.ijbpas.com

NEW METAHEURISTIC SCHEDULING ALGORITHMS FOR GRID SYSTEMS BASED ON GENETIC ANDFIREFLY ALGORITHMS

EBRAHIM BEHROUZIAN NEJAD[1*], HOSSEIN PARVAN[2], HAMID MEHDI[3], AMIN EINIPOUR[4]

1: Department of Computer, Andimeshk Branch, Islamic Azad University, Andimeshk, Iran

2: Department of Computer, Khouzestan Science and Research Branch, Islamic Azad University, Ahvaz, Iran

3: Department of Computer, Andimeshk Branch, Islamic Azad University, Andimeshk, Iran

4: Department of Computer, Andimeshk Branch, Islamic Azad University,Andimeshk, Iran

***Corresponding Author: E Mail: Behrouzian.e@gmail.com**

ABSTRACT

Grid computing system includes a set of programs and resources that are distributed in grid machines. The heterogeneous and dynamic nature of the grid, as well as the differing demands of applications run on the grid, makes grid scheduling complicated, so the deterministic algorithms will not necessarily be efficient to solve these kinds of problems. Hence, various researches have focused on heuristic algorithms such as Genetic Algorithm (GA). The simplicity and parallel nature of genetic algorithm and also searching the problem environment in different ways leads to using it for resolving several optimization problems. But since genetic algorithm is an inherent algorithm which searches the problem space globally and does not have the efficiency required for local searching, combining it with local search algorithms can compensate for this shortcomings. In this research, it has proposed a hybrid scheduling algorithm for solving the independent task scheduling problem in grid which is composed of GA with Firefly algorithm considering factor of time. The results of simulation show that the proposed algorithm can decrease Makes pan of 10% as compared to the best processed method.

Keywords: Computation Grid, Scheduling, Genetic Algorithm, Firefly Algorithm

INTRODUCTION

A computational grid is a large scale, heterogeneous collection of autonomous systems, geographically distributed and interconnected by heterogeneous networks (Singh and Sahu, 2014). Task scheduling problem in grid computing is a NP-complete problem (Izakian *et al*., 2009). Scheduler is used to manage the jobs and resources. Scheduler performs two main functions; First scheduler selects the appropriate computational resource for the job and then assigns the resource to the jobs (Sharma and Mittal, 2013). Generally, there are three methods for scheduling (Pooranian *et al*., 2013):

1. Manual scheduling. The user divides the tasks between different resources.

2. Application-mode scheduling. Applications perform the scheduling, with each application defining the resources, such as MPI programs, required for its execution. A list of machines that have MPI programs is given to the user at runtime.

3. Scheduling that is independent of applications, such as scheduling by a grid broker. This method is much more appropriate for grid scheduling. For task processing and task analysis, applications deliver their requirements to the broker, based on the quality of service required for Their tasks.The main objective of scheduling is to reduce the completion time of an application by properly allocating the jobs to the processors. The first phase of grid task scheduling is resource discovery, which generates a list of potential resources. The second phase includes gathering information about these resources and choosing the best set of resources matching the application's requirements. In the third phase, the task is executed, which involves file staging and cleanup (Pooranian *et al*., 2013). Scheduling algorithms must be designed according to the current challenges in grid environment and they assign tasks to resource to decrease makespan which is generated. Because of the complex issues of scheduling tasks on the grid is deterministic algorithms work best for this offer. This research combines a genetic algorithm (GA) and the firefly algorithm (FA). GAs is weak for local searches and strong for global searches. Conversely, FA is a local search algorithm is therefore strong for local searches and weak for global searches. Combining the benefits of these two algorithms can solve the grid scheduling problem. This paper presents a static scheduling algorithm for scheduling independent tasks in a grid system. Static scheduling means that all necessary data

about tasks, resources, and the number of resources should be specified before execution. The advantage of static scheduling is that no overhead is exerted on the system. In addition to decreasing makespan, our proposed algorithm considers quality of service (QOS) to minimize the number of tasks that miss their deadlines.

I. Related Researches

In (Zhang and Zeng, 2013) the authors tried to optimize the convergence speed of a GA with two changing points in the standard GA. After executing the crossover action, if the fitness value of the produced population is less than the average fitness or the best individual of the population, secondary preferential hybridization or mutation is also used after the primary mutation action.

Authors in (Cruz *et al*., 2010) proposed a hybrid genetic-annealing evolutionary algorithm for the independent task scheduling problem. The main purpose of this algorithm was to find the solution that minimizes the total runtime. GAs is weak for local searches, while simulated annealing (SA) is powerful for local searches. The authors combined these two methods to use both their abilities to search the problem space. The GA includes a stochastic population generator, an elitism selection operator, and mutations and crossovers with the help of SA. Based on the fitness function, the selection operator selects the best half of the chromosomes in the population, the crossover is performed, and new children are produced for the next generation.

In (Gharooni fard *et al*., 2010) a GA is presented in which chaotic variables are used instead of random variables for chromosome production. This leads to a distribution of solutions over the entire search space and avoids local minima, so that the best solutions and productions are obtained in a shorter time.

In (Pooranian *et al*., 2013) a combination of two genetic and gravitational emulation local searches (GELS) Algorithms has been proposed to solve the problem of independent task scheduling. Since genetic algorithm works weakly in local searches, combining it with GELS Algorithm has improved this problem. In this algorithm, two factors including time and the number of missed tasks have been considered. On the other hand, GGA takes much time to analyze Tasks and if the tasks are increased enormously, it is unable to perform good results in task deadlines.

Authors in (Yousif *et al*., 2011) proposed a novel metaheuristics method based on Firefly Algorithm (FA) for scheduling jobs on grid computing. The proposed method is to dynamically create an optimal schedule to complete the jobs within minimum makespan.

In (Zhang *et al*., 2008) a heuristic approach based on particle swarm optimization (PSO) algorithm is adapted to solving task scheduling problem in grid environment. Each particle is represented a possible solution, and the position vector is transformed from the continuous variable to the discrete variable. This approach aims to generate an optimal schedule so as to get the minimum completion time while completing the tasks.

In (Pooranian *et al*., 2011) an optimized scheduling hybrid algorithm was presented named GPSO. GPSO is composed of both particle swarm optimization and GELS algorithms. Since PSO is weak on local search, GELS was used for improve it and avoid becoming trapped in a local optimum. GPSO is done independent Task scheduling to decrease Makespan and minimize the missed tasks.

II. Genetic Algorithm

Genetic algorithms (GA) are search methods based on principles of natural selection and genetics (Holland, 1975). GAs encodes the decision variables of a search problem into finite-length strings of alphabets of certain cardinality. The strings which are candidate solutions to the search problem are referred to as chromosomes, the alphabets are referred to as genes and the values of genes are called alleles. A simple GA works as follows (Gharooni fard *et al*., 2010):

1. Initialize the population.
2. Calculate fitness for each individual in the population.
3. Reproduce selected individuals to form a new population.
4. Perform crossover and mutation on the population.
5. Loop to step 2 until some condition is met.

III. Firefly Algorithm

Firefly algorithm (FA) is a metaheuristic algorithm, inspired by the flashing behavior of fireflies (Yousif *et al*., 2011). Firefly optimization algorithm illustrated by (Yang, 2010) can be described as follows:

- The firefly x attracts all other fireflies and is attracted to all other fireflies.
- The less bright firefly is attracted and moved to the brighter one.
- The brightness decreases when the distance between fireflies is increased.
- The brightest firefly moves randomly (no other fireflies can attract it).
- The firefly particles are randomly distributed in the search space.

IV. Scheduling Problem Description

The scheduling problem for independent tasks is an NP-hard problem that consists of N tasks and M machines. Each task should be considered to be processed by each of the M machines, so that the makespan is minimized. However, this only considers one of the QOS parameters, the time constraint and ignores the cost. Each task can be executed on only one resource and is not stopped before its execution is complete. Since our proposed scheduling algorithms are static, we assume that the expected execution time for each task i on each resource j have already been determined. According to equation (1), if a possible solution to the scheduling problem, task i is allocated to resource j, the string corresponding to element i, is equal j.

$$solution_i = j \quad (1)$$

Fitness function is calculated to equation (2).

$$\text{Cost} = \text{Min}\{\text{Makespan} = \text{Max}(\text{RunningTime}[i, j \quad (2)$$

In equation (2), RunningTime[i,j] is equal to the time that task i is ended on resource j and is calculated to equation (3). The purpose of scheduling is that each task should be sent to each resource so that finally makespan is minimized.

$$\text{RunningTime}[i, j] = \frac{\text{JobCycle}_i}{\text{Processor Speed}_j} \quad 1 \le i \le N \quad (3)$$

In equation (3), $JobCycle_i$ parameter indicates the length (number of instructions) of task i and Processor $Speed_j$, processor speed determines the resource j. The speed of each resource is expressed in the form of MIPS (Million Instructions per Second), and the length of each job in the number of instructions.In each iteration of the population update, the generated solutions are evaluated. Based on this assessment, fitness function given in equation (2) is calculated.

V. The Parameters of the Proposed Algorithms

As the genetic algorithm is largely dependent on how the chromosomes, the proposed scheduling algorithms, a simple method has been used to represent chromosomes. That natural numbers is used to encode the chromosomes.In this algorithm, a solution to the problem is represented by a chromosome. Length of chromosomes is considered equal to size of the number of input independent tasks. The values inside the genes are random numbers between 1 to M. M is the resource numbers. Algorithm finished when the iteration time of the algorithm reaches its maximum. Figure.1 shows an example of the chromosome representation. For example, in the figure, task T_3 executes on resource R_1.

A. Initial Population

The first step in the proposed algorithms, initial solutions is generated. In these algorithms, the initial population is created randomly. That is produce a random number between 1 and M, to the task being considered executed on it. In other words, each task is randomly allocated to one of the resources.

B. Selection Operator

Before using mutation and crossover operators, it is the selection step. Here elitist operator has been used of chromosome selection.

C. Crossover Operator

The proposed algorithms are used to uniform crossover operator. In uniform crossover, a value of the first parent's gene is assigned to the first offspring and the value of the second parent's gene is to the second offspring.

D. Mutation Operator

The mutation operator preserves diversification the search. Two points on each chromosome is randomly selected and then changed to a random number between 1 and M.

VI. How to Use a Firefly Algorithm (FA) in Scheduling Problem

One of the key issues in applying FA successfully to job scheduling is how to encode a schedule to a search solution, i.e. finding a suitable mapping between problem solution and firefly.In the proposed methods each firefly represents a candidate solution of the grid scheduling problem in a vector form, with n elements; where n is the number of jobs to be scheduled and each element is an integer value between 1 to M, that are produced randomly.Where M is the total number of resources. Firefly i specifies the resource to which the job number i is allocated. **Figure 2** illustrates how 4 tasks are allocated to 4 resources.

Thus, we should have M machine and N job, to convert continuous optimization problem to the discrete optimization problem. Round off the real optimum values to its nearest integer number. With this operation, a continuous optimization problem becomes a discrete optimization problem. After determining the fitness using the fitness function corresponding to each firefly, firefly i brightness rate of equation (4) is achieved.

$$Brighness_i = \frac{1}{\cos t_i} \quad (4)$$

In the above equation, $Cost_i$ denotes the fitness function for each firefly is calculated using equation (2) and $Brighness_i$, expresses brightness firefly i. To improve the convergence speed of the proposed algorithm, the algorithm compares the attractiveness of the new firefly position with old one. If the new position produces higher attractiveness value, the firefly is moved to the new position; otherwise the firefly will remain in the current position.Calculate the distance r between each two fireflies X_i and X_j based on the equation (5) (Yang, 2010).

$$r_{i,j} = \| X_i - X_j \| = \sqrt{\sum_{k=1}^{d} (X_{i,k} - X_{j,k})^2}$$
(5)

In equation (5), $X_{i,k}$ is the kth component of the spatial coordinate Xi of ith firefly. d parameter is total number of independent task input. The movement of a firefly i is attracted to another more attractive (brighter) firefly j is determined by equation (6) (Yang, 2010).

$$X_i = X_i + \beta . e^{-\eta r_{i,j}} (X_j - X_i) + \alpha .(rand - 0.5)$$
(6)

Where the second term is due to the attraction.The third term is randomization with α being the randomization parameter; rand is a random number generator uniformly distributed in [0,1]. β is attractiveness rate of a firefly. The parameter η now characterizes the variation of the attractiveness, and its value is crucially important in determining the speed of the convergence and how FA algorithm behaves.

VII. The Proposed Method (FA-GA)

In the proposed algorithm, the genetic algorithm parameters (crossover and mutations) to improve the firefly algorithm are applied. In the proposed method, the original algorithm is an algorithm firefly. Firefly in a local search algorithm has good performance, but in global search does a poor job. However, the genetic algorithm has good performance of global searching. The use of genetic algorithm parameters in firefly algorithm, lead to poor performance firefly algorithm to solve the global search and quick convergence and finding the optimal solution is improved. Apply the crossover operator makes the solution space will be searched as well. Mutation operator is improved solutions. Also cause the optimal solution in iterative by using a local search algorithm to find a firefly. This process is repeated for each generation of the population. The flowchart of the proposed method, in **Figure.3** is shown.

VIII. The Proposed Method (GA-FA)

In the proposed algorithm, the genetic algorithm as the basic algorithm is implemented in this way. After finishing the genetic algorithm, the best chromosome of the GA as the current solution is firefly algorithm. Finally, find the optimal solution by using an iterative local search by firefly algorithm. The flowchart of the proposed method, in **Figure 4** is shown.

IX. Experimental Results

The FA-GA and GA-FA algorithms was implemented using matlab software running under the Win 7 operating system on a 2.66GHZ CPU with 4GB RAM. In our proposed algorithm, we assumed that the crossover rate CR = 0.50 and the mutation rate MR = 0.01. Also we assumes that the attractiveness rate β = 3 and absorption coefficient η = 0.001. The results of simulations comparing FA-GA and GA-FA with the FA and GA algorithms are shown in **Figures 5 and 6**. Here, we have tested our works on various tasks; Generations and different fitness function orderly. The diagram in **Figure 5** shows a number of scheduled tasks between 1000 and 5000 allocated to 20 resources using the comparison algorithms. As the figure shows, when the number of tasks increases, the makespan increases as well. The diagram shows that our proposed algorithm (FA-GA) produces a smaller makespan than the other algorithms.

In **Figure 6**, the effect of heterogeneous task lengths on the algorithms is investigated. As the figure shows, when the heterogeneous of task lengths increases, the makespan increases as well. The diagram shows that our proposed algorithm (FA-GA) produces a smaller makespan than the other algorithms.

T_1	T_2	T_3	T_4	. . . .	T_N
R_4	R_3	R_1	R_2	. . . .	R_2

Figure 1; Chromosome representation

	T_1	T_2	T_3	T_4
Firefly 1	R_1	R_2	R_3	R_4
Firefly 2	R_1	R_3	R_4	R_1

Figure 2: Fireflies representation.

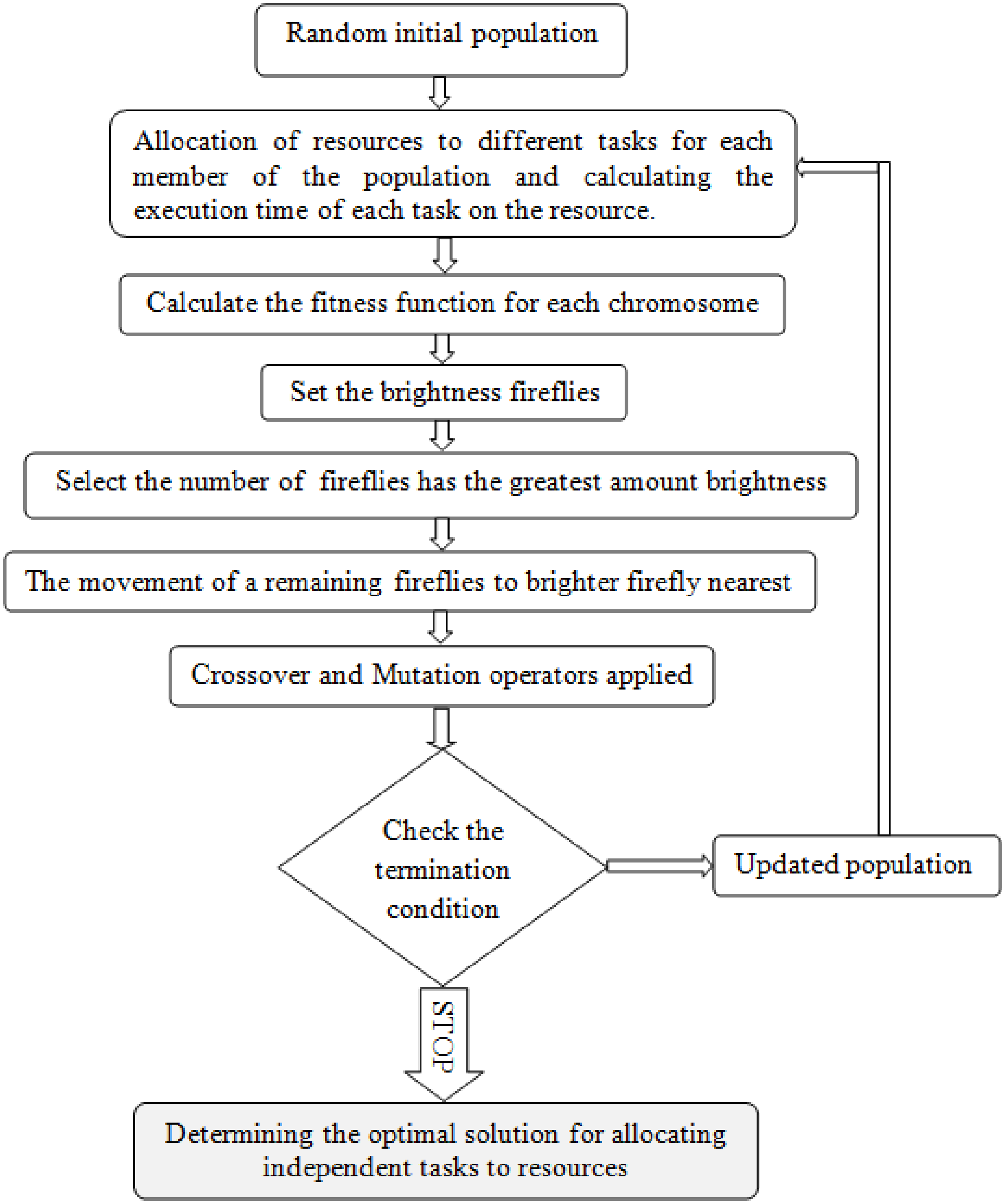

Figure 3: FA-GA flowchart

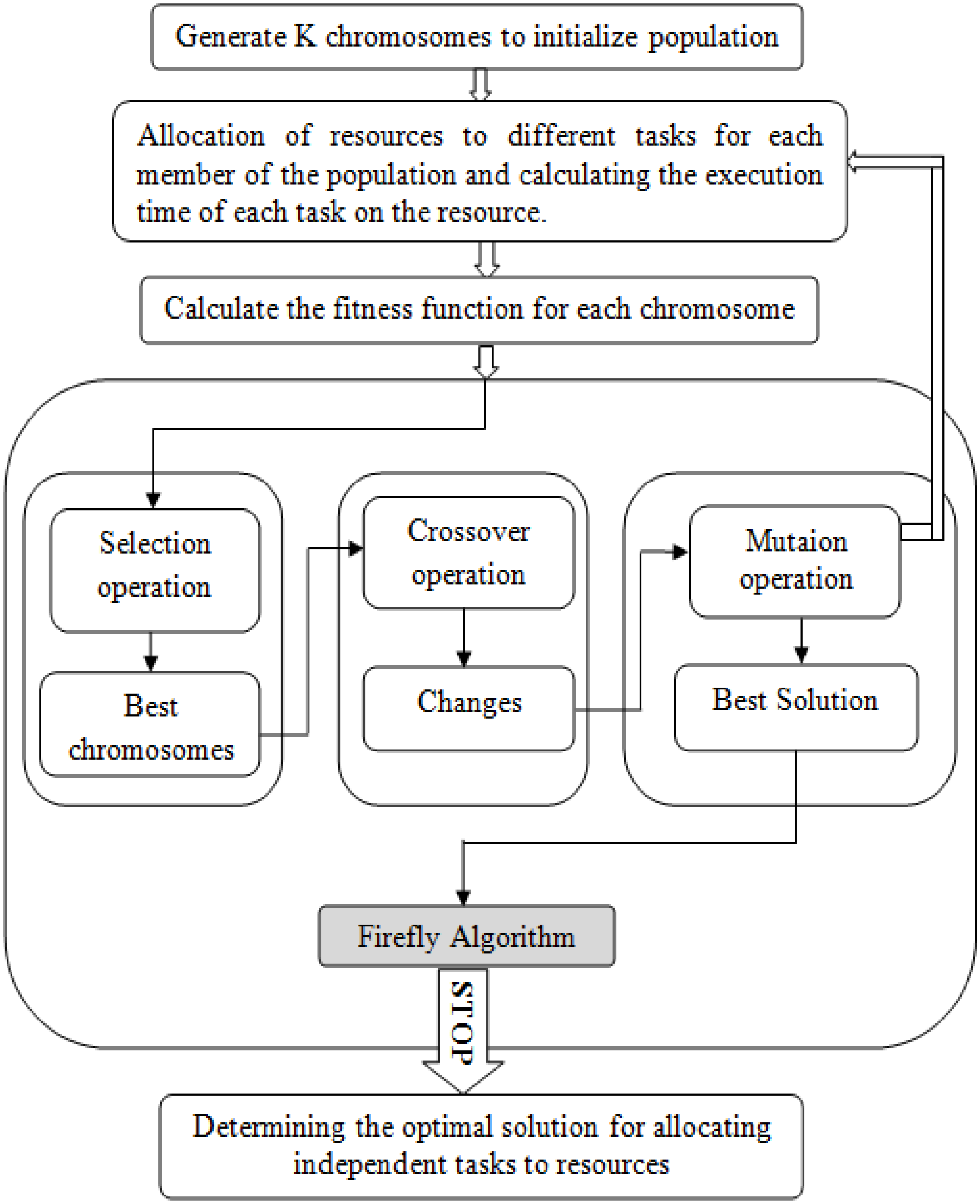

Figure 4: GA-FA flowchart

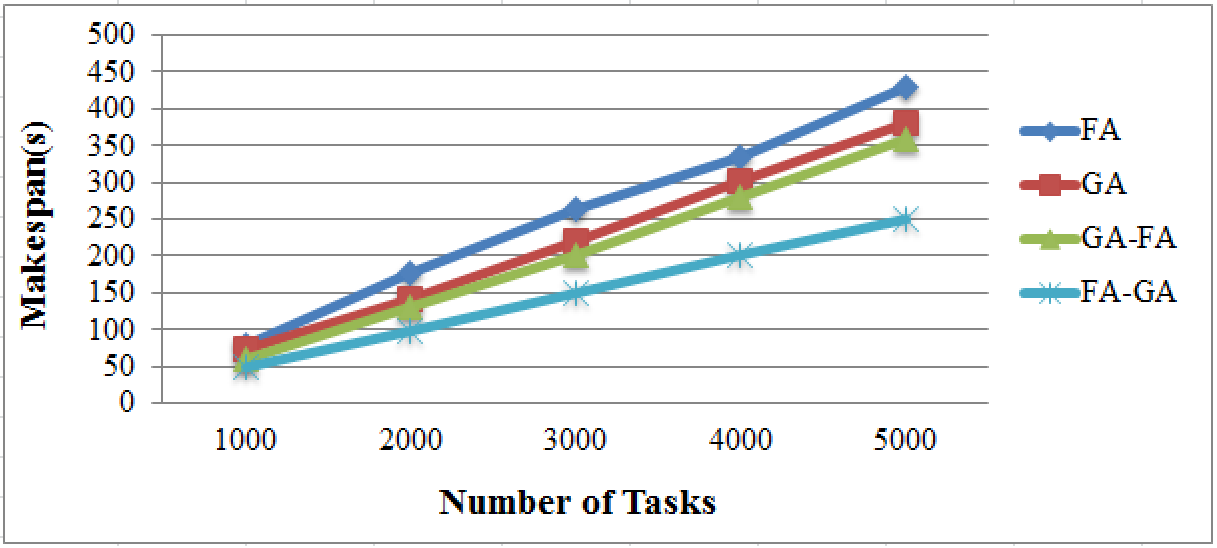

Figure 5: Compares of makespans

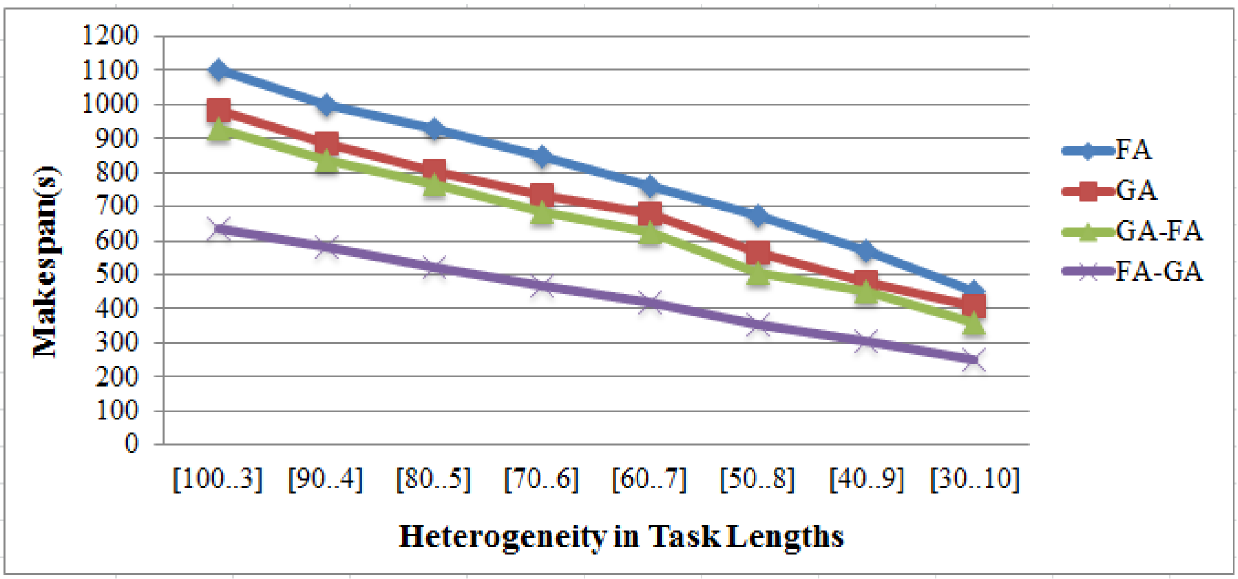

Figure 6: Makespan comparison of heterogeneity task lengths

X. CONCLUSION

This paper presented two algorithms for solving the grid task scheduling problem through a combination of a genetic algorithm (GA), which is a global search algorithm, and the firefly algorithm (FA), which searches locally. The algorithm aims at minimizing makespan. Firefly algorithm because it has features such as high convergence speed, being insensitive to the initial value, flexibility and fault tolerance, is one of the most heuristic ways. By combining the advantages of the firefly and genetic algorithms, both the convergence velocity and the GA identification of an optimal response are improved. The proposed algorithms were compared to the other algorithms and the simulation conclusions showed that the FA-GA produced less makespan comparing the other algorithms.

REFERENCES

[1] Cruz CM, Rodriguez LA, Avila ME, Juarez PF, Cruz RM, Rivera LR (2010). Genetic-annealing algorithm in grid environment for scheduling problems. *Security-Enriched Urban Computing and Smart Grid Communications in Computer and Information Science*, Vol(78), pp 1-9.

[2] Gharooni fard G, Moein Darbari F, Deldari H, Morvaridi A (2010). Scheduling of scientific workflows using a chaos-genetic algorithm. *Procedia Computer Science*, Vol(1), No(1), pp 1445- 1454.

[3] Holland J (1975). Adaptation in natural and artificial systems. MIT Press Cambridge, US.

[4] Izakian H, Tork Ladani B, Zamanifar K, Abraham A (2009), A novel particle swarm optimization approach for grid job scheduling, *Communications in Computer and Information Science*, Vol(31), pp 100-109.

[5] Pooranian Z, Harounabadi A, Shojafar M, Mirabedini J (2011). Hybrid pso for independent task scheduling in grid computing to decrease makespan. *International Conferences on Future Information Technology, Singapore*, Vol(13), pp 327-331.

[6] Pooranian Z, Shojafar M, Tavoli R, Singhal M, Abraham A (2013). A hybrid meta-heuristics algorithm for job scheduling on computational grids. *Informatica*, June, Vol(37), No(2), pp 157-164.

[7] Sharma D, Mittal P (2013). Job scheduling algorithm for computational grid in grid computing environment. *International journal of advanced research in computer science and software engineering*, May, Vol(3), No(5), pp 735-743.

[8] Singh PK, Sahu N (2014). Task scheduling in grid computing environment using compact genetic algorithm. *International journal of science engineering and technology research (IJSETR)*, January, Vol (3), No(1), pp 107-110).

[9] Yousif A, Abdullah AH, Mohd Nor S, Abdelaziz AA (2011). Scheduling jobs on grid computing using firefly algorithm. *Journal of theoretical and applied information technology*, November, Vol(33), No(2), pp 155-164.

[10] Zhang X, Zeng W (2010). Grid workflow scheduling based on improved genetic algorithm. *International conference on computer design and application (ICCDA 2010)*, pp 270-273.

[11] Zhang L, Chen Y, Sun R, Jing S, Yang B (2008). A task scheduling algorithm based on pso for grid computing. *International journal of computational intelligence research*, Vol(4), No(1), pp 37–43.