DOI: http://dx.doi.org/10.26483/ijarcs.v8i8.4642

ISSN No. 0976-5697

Volume 8, No. 8, September-October 2017

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

INTEGRATED GA AND FIREFLY BASED SCHEDULING TECHNIQUE FOR PARALLEL COMPUTING

Er.Davinderjit Kaur
Department of Computer Engineering and Technology
Guru Nanak Dev University
Amritsar, India

Er.Amit chabbra
Department of Computer Engineering and Technology
Guru Nanak Dev University
Amritsar, India

Abstract— This paper has focused on a new Meta heuristic technique i.e. GA+ FIREFLY Hybrid algorithm for parallel job scheduling problem. It has been observed that in existing literature has introduced genetic algorithm which solve parallel job scheduling problem but the genetic algorithm suffers from local optima problem. Moreover it converges slowly so more time it takes to provide the final results. In order to eliminate this problem further improvement has been required to get the sub optimal solution as well as Firefly algorithm works on global optima. It is flexible, robust. Moreover, it uses few parameters as compared to GA and it can be easily hybridized with GA. This research has proposed the hybridisation of GA and Firefly which has done the work on various parameters like make span, flow time, mean waiting time, normalization function etc. The experimental results will also be drawn in order to find the best decomposition among the available one.

Keywords—Parallel computing; Scheduling;Genetic algorithm;Firefly algorithm.

1.INTRODUCTION

A computing system is described as collection of either homogeneous systems or heterogeneous systems. During an instruction cycle, execution of programmed on computing system may use different number of processors at different instance of time [1]. Emerging concept that can simultaneously execute various tasks on different processors is known as processing. It is an effective way of solving complex and computation intensive issues. System can be categorized into two systems: homogeneous and heterogeneous, depending upon the nature of processors [2]. The core component of system is multiprocessor system (MTS). In MTSs, there are multiple input/output modules, multiple processing elements, and multiple memory modules. computing is a kind of calculation where several computations are finished simultaneously[3][4].

1.1 Scheduling

Scheduling in systems of industries alludes to determine job's arrangement in which they need to be refined completely over work phases, accompanied by finding along with completion time of jobs being processed [4]. The scheduling has gained its importance in present times because of developing demands of consumer for varying markets with worldwide competition, quick advancement of newly formed procedures thus, advances reduced product life cycles [5]. Other than this, the era of reliably good schedules has ended up being to a great degree difficultly in moderate to big shops, ideal scheduling include procedures- immoderate and illogical identification[6]. Scheduling large amount of jobs over distributed systems play a significant role to achieve throughput and high system utilization. The sequence in which jobs are scheduled can have full effect on system performance [7]. The scheduling problem exemplifies numerous varieties for example,, single machine, flow shop etc . Every single of these issue classes is novel, and possesses its requirements plus goals[8].

1.2 Various Algorithms for Scheduling

There are Deterministic and Metaheuristic algorithms are available for solving Scheduling problems in computing.

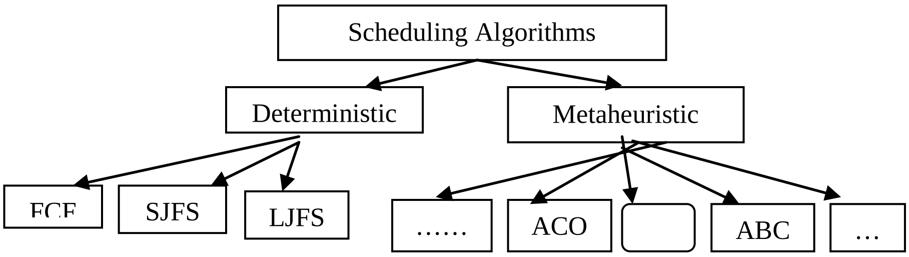


Figure 1. Scheduling algorithm

1.3 Genetic Algorithm

A Genetic Algorithm is a stochastic search heuristic utilized to locate just about exceptional solutions through nature-based techniques[9]. This will begin by developing a basic people with solutions referred to as people today, each protected having a chromosome[10]. To make a new age group, some actions are performed: ranking individuals motivated with a health performance, a ranking-based choice, your cross-over as well as the mutation[11]. The algorithm criteria is enthusiastic with the trust any time many decades, the revolutionary people will improve when compared with the prior ones[12].

1.4 The Firefly algorithm

The particular **firefly protocol** is usually a meta heuristic consist of by simply Xin-She Yang in addition to motivated from the blinking behavior involving fireflies[13]. The key intent to get a firefly's expensive is to work as a proof procedure to bring in various other fireflies. Xin-She Yang made this kind of firefly protocol by simply accepting:

1. All fireflies tend to be unisexual, making sure that any kind of specific firefly will be enthusiastic about all other fireflies[14] [15];
2. Attraction is definitely proportional with their brightness, in addition to for just about any not one but two fireflies, the lesser bright become attracted by means of (and hence move towards) the richer one particular; however, this intensity) lessen because their common difference increases[16];
3. When there is not any fireflies brighter when compared to a presented firefly, it can move randomly[17][18].

Algorithm of Firefly

```
Objective function f(X), x=(x1,………xd)
Generate initial  population of fireflies  xi(i=1,2,….n)
Light intensity Ii at xi is determined by f(xi)
Define light absorption coefficient y
While (t<max generation)
For i=1:n all n fireflies
For j=1:n all n fireflies
If (I3>Ii), move firefly i towards j in d-dimensions
End if
Attractiveness varies with distance r via exp[-yr]
Evaluated new solution and update light intensity
End for  j
End for i
Rank the fireflies and find the current best
End while
Post process result and visualization.
```

2. RELATED WORK

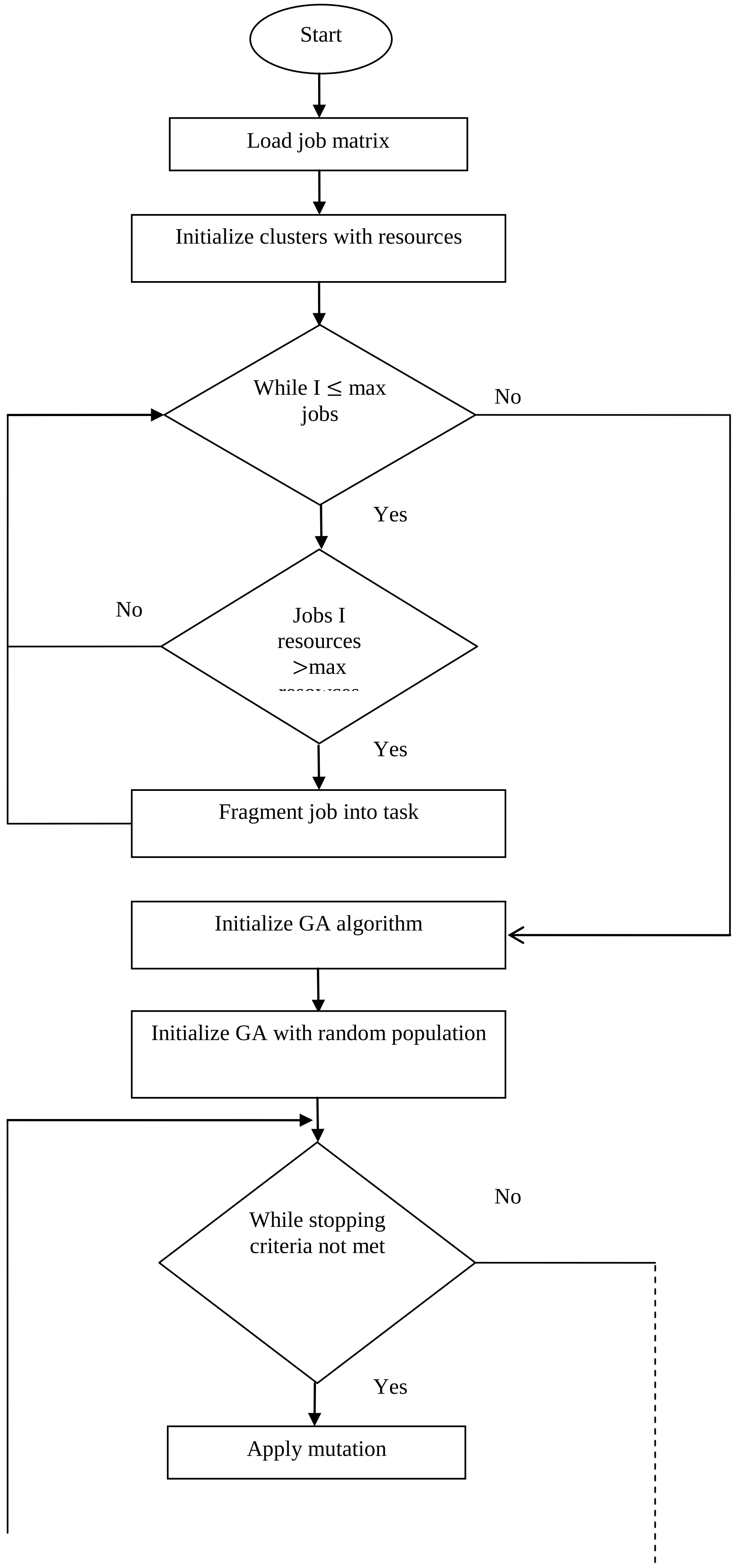
Masdari M, et al. [1] have tackled the particular independent batch scheduling within the computational grid like a bi objective global reduction issue with makespan and power usage as primary requirements and also applied Dynamic voltage Scaling Frequency technique towards the administration of accumulative energy used by the particular grid resources as well as create three genetic algorithm just as power conscious grid schedulers that have been empirically examined within three grid capacity circumstances within fixed and variable modes. The actual research effects demonstrated the actual operation regarding planned procedure inside the minimization regarding electric power use by means of total method plus in adjustable weight balancing within the options in grid groupings that are sufficient to take care of essential Top quality grade. **Farahnakian, F., et al. [2]** currently have provided Multi Objective Synthetic bee colony (MOABC) Seo Algorithm criteria which will run optimization procedures both equally rendering period and also electric power use to solve employment scheduling problems in Grid.MOABC is actually a swarm Algorithm which is motivated by bees actions as well as its comparison is done with another swarm algorithm which is motivated by fireflies actions. NSGA-II Algorithm is compared with above two algorithms to jujge their multi objective functions. Additionally , the most effective algorithm, MOABC is compared with one of the most well -known algorithm MOHEFT regarding workflow scheduling as well as with WMS or DBC i.e. real Grid Schedulers .The outcome acquired indicates that MOABC is great approach out of all algorithm discussed. **Alkhanak, et al. [3]** get unveiled crossbreed technique called FUGE which usually relies upon Fuzzy concept along with innate algorithm criteria which usually was created to execute highest handling with a lot having in to account delivery period and cost.It alter the conventional genetic algorithm and also fuzzy technique in order to create new fuzzy based GA to be able to enhance the efficiency such as makespan. this algorithm allocates tasks to sources through taking in to account Virtual machine Computing rate, storage, bandwidth of VM and job size. **Arsuaga-Ríos, et al. [4]** supplied your replication in strategy to reach various aims ,just like reducing the going some time and strength cost.The primary algorithms employed are actually genetical protocol and also ant Colony optimization, with a brand new variable fusion technique offered to get maximum perfect method quickly. **Javanmardi, et al.[5]** have symbolized a new hybrid occupation Booking Strategy making use of innate Algorithm criteria using fluffy idea which regularly considers force balancing with program furthermore decreases entire managing some time to cost.It modified conventional genetic algorithm as well as minimize iteration of developing population using fuzzy theory.

```
1. Define objective function f(a), where a = (a1,...... ad)
 2. Produce an underlying population of fireflies
3. Formulate the light intensity L
4. Specify the absorption coefficient β
 5. While (t< Max_Gen)
6. For i=1 to n (all n fireflies)
 7. For j=1 to n (all n fireflies)
  8.  If ( Lj > Li ), move firefly i towards  firefly j
  9.   End if
10.  Examine new solutions and update light intensity;
11.   End for j
 12. End for i
13.   Rank the fireflies and find the current best
14.   End while
```

3.METHODOLOGY

3.1 PSEUDO CODE FOR HYBRID GA+FIREFLY ALGORITHM

```
1. Pick an initial solution using random distribution.
2.  Evaluate the fitness function of the individuals
 3. Repeat
4.  The best individuals should be selected
5. Generate new individuals by applying crossover and mutation operators
 6. The fitness function for new individuals should be examined
 7. The worst individuals are replaced with the best ones
8. Until a stopping criteria is met
9. Apply firefly based optimization to attain more efficient results.
10.Objective function $f(X), x = (x1, … … … xd)^T$
11.Generate initial population of fireflies $x_i(i = 1,2, … . n)$
12.Light intensity Ii at xi is determined by $f(xi)$
13.Define light absorption coefficient$y$
While $(t < maxgeneration)$
14. For $i = 1: nallnfireflies$
$Forj = 1: n$ all$n$ fireflies
$If\ (I_j > I_i)$, move firefly$i$towards $j$ in d-dimensions
End if
Attractiveness varies with makespan$rviaexp[-$□$r]$
Evaluated new solution and update light intensity
End for  $j$
End for$i$
15. Rank the fireflies and find the current best
End while
16. Return final schedule with minimum make span
```


Start
Load job matrix
Initialize clusters with resources
While I ≤ max jobs
No
Yes
No
Jobs I resources >max resowces
Yes
Fragment job into task
Initialize GA algorithm
Initialize GA with random population
No
While stopping criteria not met
Yes
Apply mutation

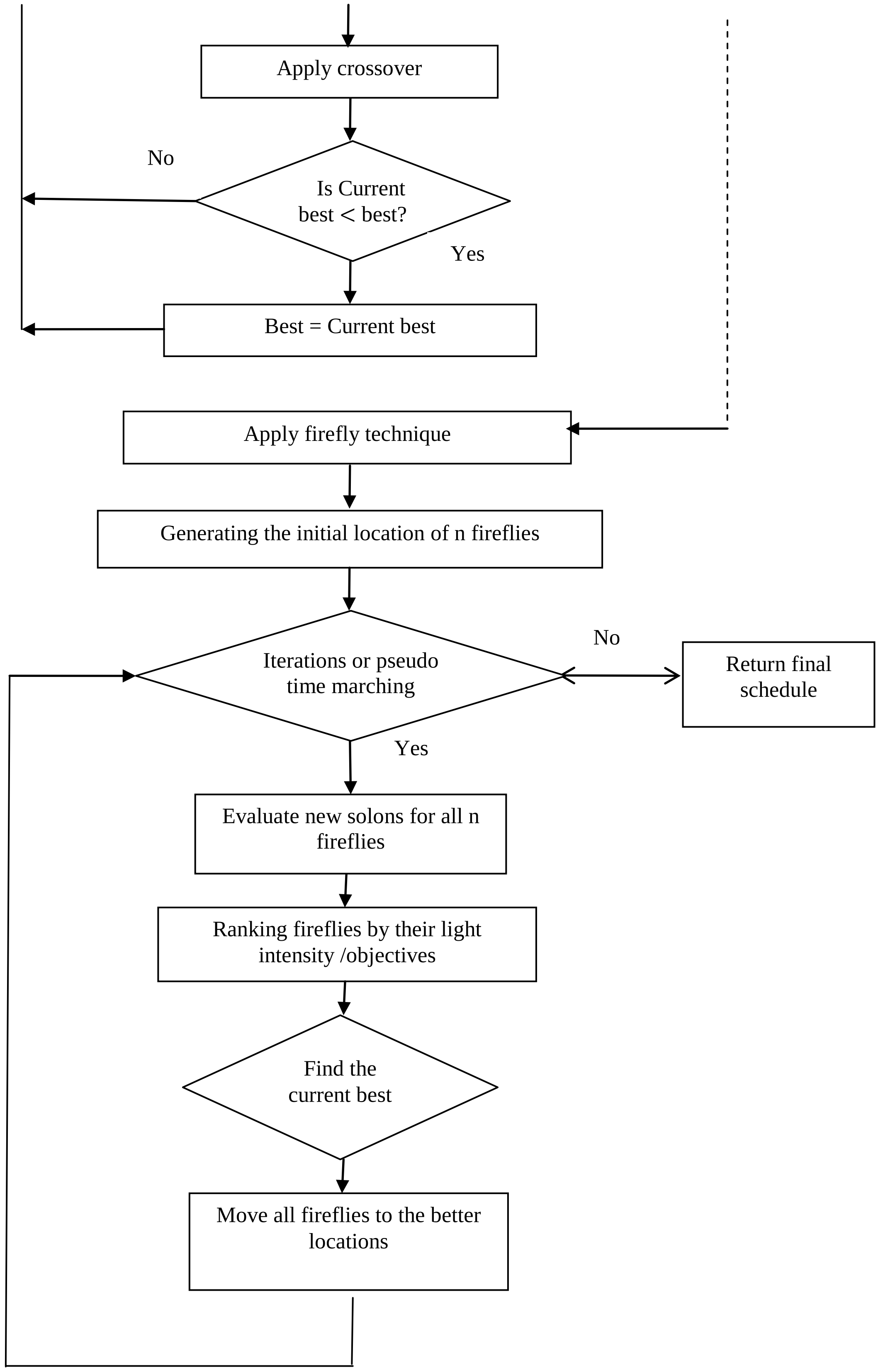

Apply crossover
No
Is Current best < best?
Yes
Best = Current best
Apply firefly technique
Generating the initial location of n fireflies
Iterations or pseudo time marching
No
Return final schedule
Yes
Evaluate new solons for all n fireflies
Ranking fireflies by their light intensity /objectives
Find the current best
Move all fireflies to the better locations

Fig2: flowchart of integrated GA+Firefly algorithm

4.Experimentation and results

A. PERFORMANCE ANYLYSIS

This proposed method is implemented by using MATLAB tool u2013a.The algorithm results are concluded by using various performance parameters Root Mean Square Error (RMSE), Peak Signal to Noise Ratio (PSNR) and structural similarity index metric (SSIM).

1. Makespan

The makespan between any two fireflies *i*and *j* at **x*i*** and **x*j***, respectively, is the Cartesianmakespan

$$r_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\| = \sqrt{\sum_{k=1}^{d} \boldsymbol{x}_{i,}k - \boldsymbol{x}_{i,}k)^2},$$

$$xi = xi + \beta_0 e^{-\gamma r^2} ij\,(x_j - x_i) + \alpha\,(rand - \frac{1}{2})$$

Where the second term is due to the attraction while the third term is randomization with α being the randomization parameter. rand is a random number generator uniformly distributed in [0, 1].

2. Waiting time

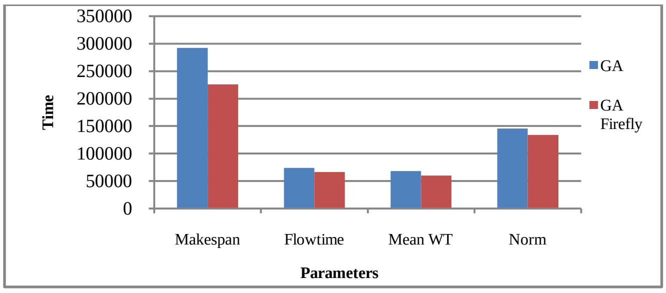
A waiting period is the period of time between when an action is requested or mandated and when it occurs. The flow price usually is an ordinary rate. Movement time: Just how much of time a flow model usually spends in business course of action out of beginning to end, also referred to as the complete processing time. If there is a few course in the course of action, the pass time is actually similar to the actual length of the longest path.

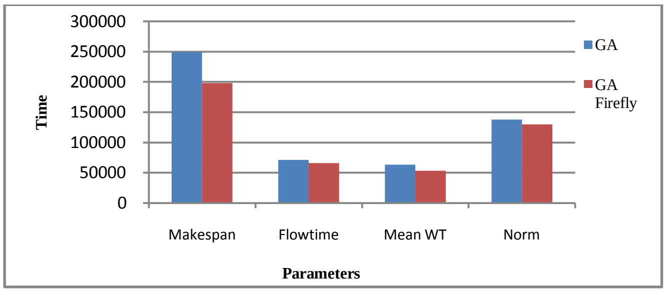
3. Flow time

The flow rate usually is a standard rate. Movement occasion: The quantity of time a flow system usually spends in a business method coming from beginning to end, also called the entire processing time. Should there be several route throughout the method, the pass time will be comparable to the actual entire top path.

Table 1. 1: Comparison of results GA and GA+Firefly for 100 jobs

Processors	Makespan		Flowtime		Mean WT		Norm	
	GA	GA+Firefly	GA	GA+Firefly	GA	GA+Firefly	GA	GA+Firefly
96	292093.3	225815	74233.2	66319	68192.3	60103	145794.4	134036
112	249402.1	197862	71390.1	66073	63516	53298	137829.1	130137
128	213003.3	152282	69716.4	45226	61457.5	39794	136738	123766





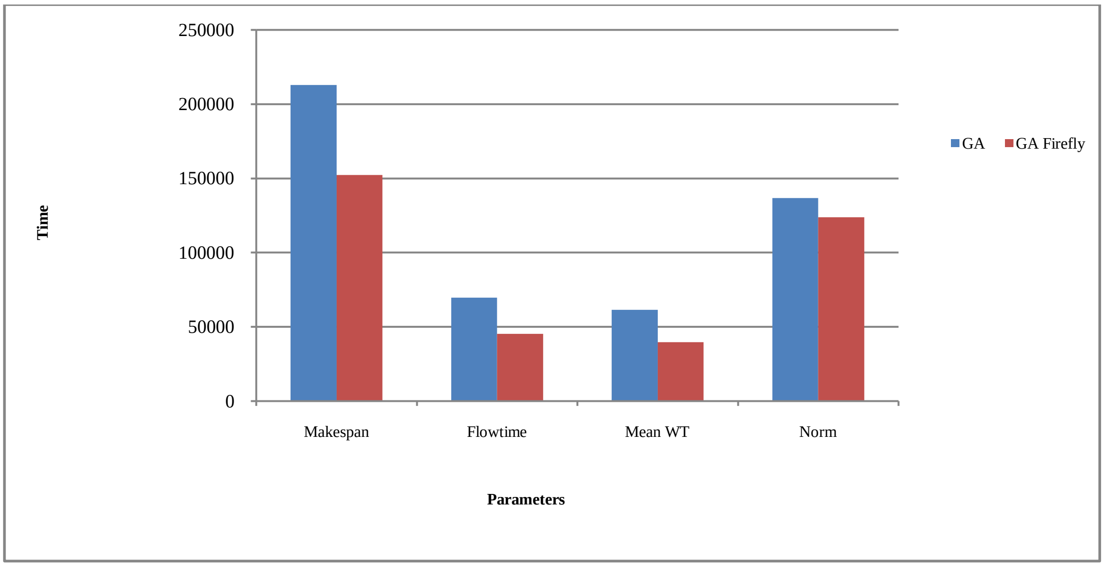
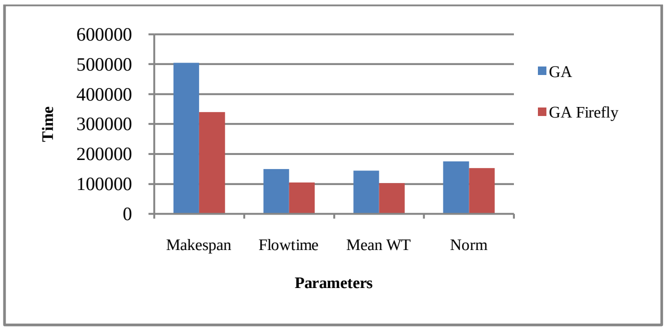
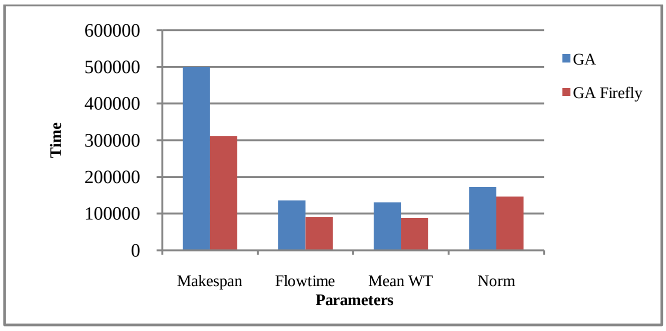


Table 1.2: Comparison of results GA and GA+Firefly on 300 jobs

Processors	Makespan		Flowtime		Mean WT		Norm	
	GA	GA+Firefly	GA	GA+Firefly	GA	GA+Firefly	GA	GA+Firefly
96	504538.6	339582	149301.7	105335	144190.4	102342	175600.8	153152
112	500032.7	310949	136024	90800	131091.4	87710	172925.3	146529
128	439991.5	254643	120632.8	78925	116095.8	75924	162104.1	141142





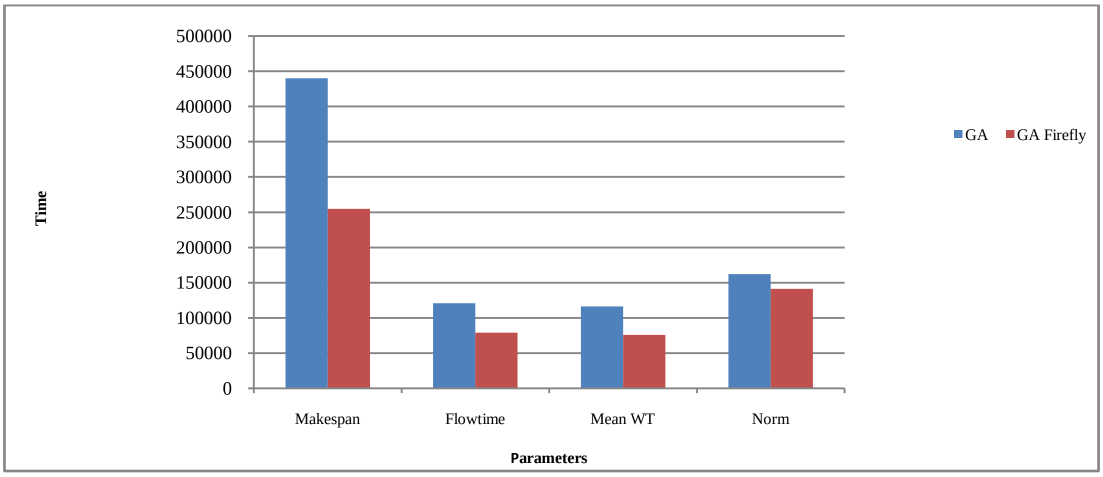
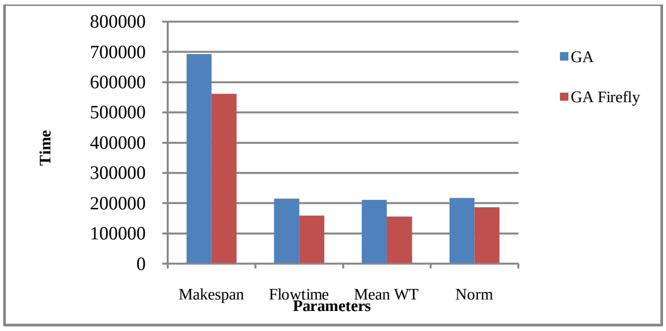
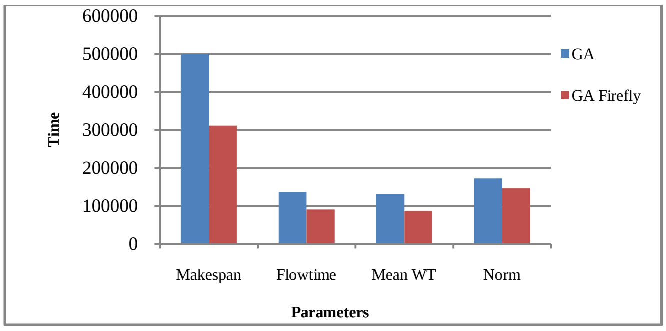


Table 1.3: Comparison of results GA and GA+Firefly on 500 jobs

Processors	Makespan		Flowtime		Mean WT		Norm	
	GA	GA+Firefly	GA	GA+Firefly	GA	GA+Firefly	GA	GA+Firefly
96	693482.2	562067	215531	158536	211020.2	155523	217542.9	186051
112	686663.3	509474	194775.6	157696	190866.2	154710	206916.2	184345
128	679961.6	408572	192612.5	125424	188177.4	122552	205713.4	170230





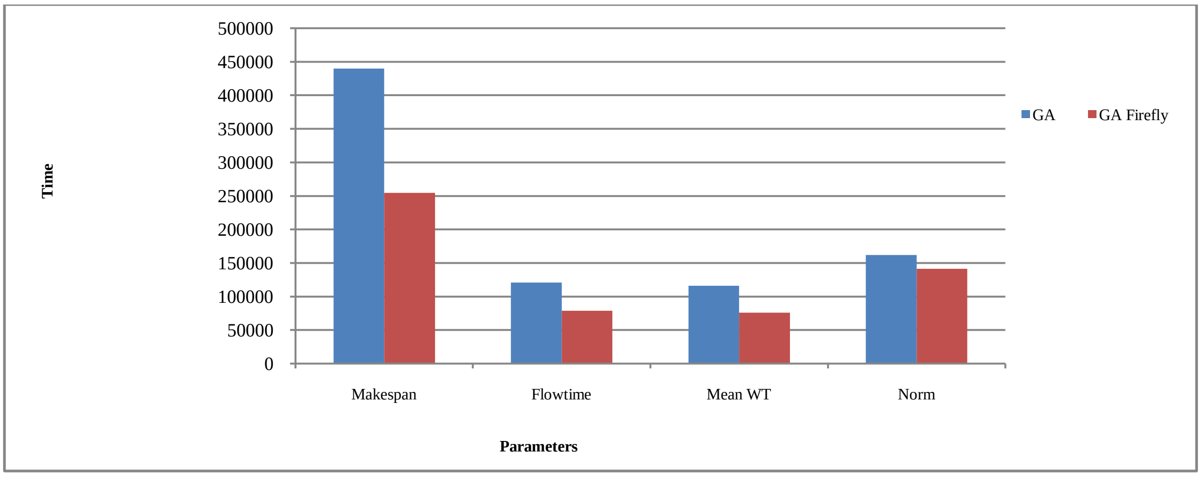
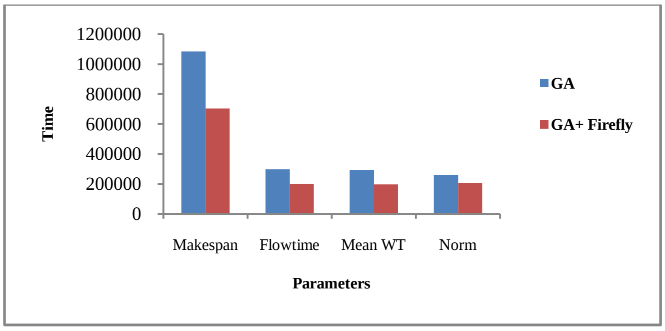
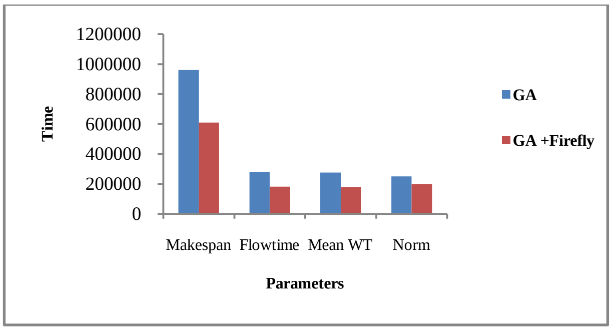
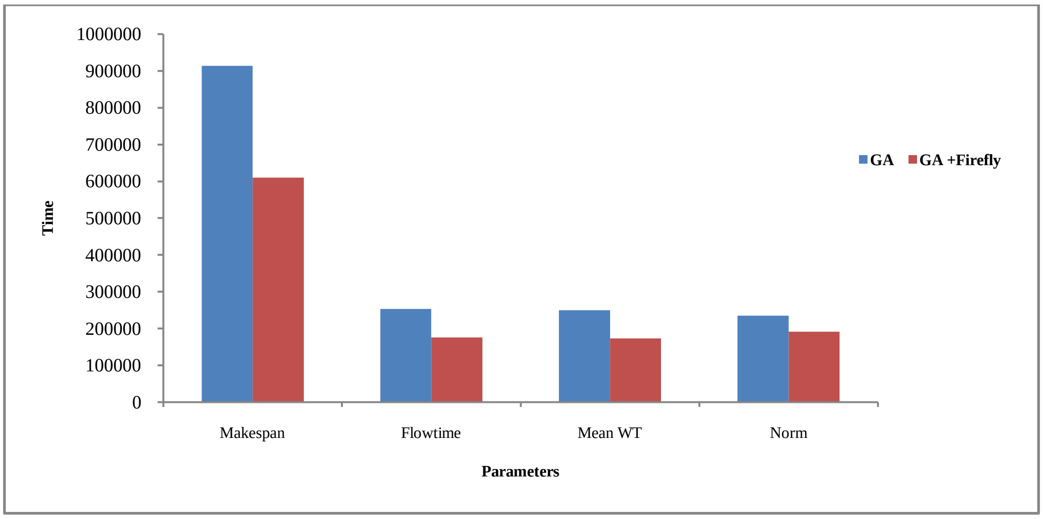


Table 1. 4: Comparison of results GA and GA+Firefly on 700 jobs

Processors	**Makespan**		**Flowtime**		**Mean WT**		**Norm**	
	GA	GA+Firefly	GA	GA+Firefly	GA	GA+Firefly	GA	GA+Firefly
96	1084776.4	703649	297869.3	200128	293437.6	197410	259975.1	207627
112	959628.3	609716	280279.1	181582	276000.3	178777	249974.2	198772
128	914217.8	609815	253563.4	175564	249389.2	172778	234882.6	191727







4.CONCLUSION AND FUTURE WORK

This research work has proposed the hybridisation of GA and Firefly which has done the work on various parameters like make span, flow time, wait time, normalization etc. In this we have also consider the bi-objective function which referred as normalization function which works on the two parameters i.e. makespan and flow time. In this proposed work scheduling Firefly performs better than Genetic algorithm. So it performs better in parallel scheduling. The comparison between the proposed techniques with the existing technique using parameters such as: Flow Time, speedup and Make Span is shown in this work. This comparison has shown that the proposed work results are much better than the existing results. This is because the reduction in make span is 60,321 , wait time is 21,663 and further flow time reduction is 24,490. Hybrid GA+Firefly algorithm based Scheduling does not guarantee the high availability of services i.e. effect of failures are ignored. Therefore, in near future we will propose fault tolerance based Scheduling will be designed to provide high availability of resources.

REFERENCES

[1] Masdari, M., ValiKardan, S., Shahi, Z., & Azar, S. I. (2016). Towards workflow scheduling in cloud computing: A

[2] comprehensive analysis. Journal of Network and Computer

[3] Applications, 66, 64-82.

[4] Farahnakian, F., Ashraf, A., Pahikkala, T., Liljeberg, P., Plosila, J., Porres, I., & Tenhunen, H. (2015). Using ant

[5] colony system to consolidate vms for green cloud

[6] computing. IEEE Transactions on Services

[7] Computing, 8(2), 187-198.

[8] Alkhanak, Ehab Nabiel, Sai Peck Lee, and Saif Ur Rehman Khan. "Cost-aware challenges for workflow scheduling approaches in cloud computing environments: Taxonomy and opportunities." Future Generation Computer Systems 50

[9] (2015): 3-21.

[10] Arsuaga-Ríos, María, and Miguel A. Vega-Rodríguez. "Energy optimization for task scheduling in distributed systems by an Artificial Bee Colony approach." Nature and Biologically Inspired Computing (NaBIC), 2014 Sixth

[11] World Congress on. IEEE, 2014.

[12] Javanmardi, Saeed, et al. "Hybrid job scheduling algorithm for cloud computing environment." Proceedings of the Fifth International Conference on Innovations in Bio-Inspired

[13] Computing and Applications IBICA 2014. Springer

[14] International Publishing, 2014.

[15] Javanmardi, S., Shojafar, M., Amendola, D., Cordeschi, N., Liu, H., & Abraham, A. (2014). Hybrid job scheduling

[16] algorithm for cloud computing environment. In Proceedings of the Fifth International Conference on Innovations in Bio-Inspired Computing and Applications IBICA 2014 (pp. 43-52). Springer International Publishing.

[17] Dorronsoro, B., Nesmachnow, S., Taheri, J., Zomaya, A. Y., Talbi, E. G., & Bouvry, P. (2014). A hierarchical approach

[18] for energy-efficient scheduling of large workloads in

[19] multicore distributed systems. Sustainable Computing:

[20] Informatics and Systems, 4(4), 252-261.

[21] Sajedi, Hedieh, and Maryam Rabiee. "A metaheuristic

[22] algorithm for job scheduling in grid

[23] computing." International Journal of Modern Education and Computer Science 6.5 (2014): 52.

[24] Bilgaiyan, Saurabh, Santwana Sagnika, and Madhabananda Das. "An analysis of task scheduling in cloud computing using evolutionary and swarm-based

[25] algorithms." International Journal of Computer

[26] Applications 89.2 (2014).

[27] Pinel, F., Dorronsoro, B., Pecero, J. E., Bouvry, P., & Khan, S. U. (2013). A two-phase heuristic for the energy-efficient scheduling of independent tasks on computational

[28] grids. Cluster Computing, 16(3), 421-433.

[29] Civicioglu, Pinar, and Erkan Besdok. "A conceptual

[30] comparison of the Cuckoo-search, particle swarm

[31] optimization, differential evolution and artificial bee colony algorithms." Artificial intelligence review 39.4 (2013): 315-346.

[32] Zhao, Jianfeng, and Hongze Qiu. "Genetic algorithm and ant colony algorithm based Energy-Efficient Task

[33] Scheduling." Information Science and Technology (ICIST), 2013 International Conference on. IEEE, 2013.

[34] Liu, J., Luo, X. G., Zhang, X. M., Zhang, F., & Li, B. N.

[35] (2013). Job scheduling model for cloud computing based on multi-objective genetic algorithm. IJCSI International

[36] Journal of Computer Science Issues, 10(1), 134-139.

[37] Wang, Lizhe, et al. "Energy-aware parallel task scheduling in a cluster." Future Generation Computer Systems 29.7 (2013): 1661-1670.

[38] Li, Jun-Qing, Quan-Ke Pan, and Kai-Zhou Gao. "Pareto-

[39] based discrete artificial bee colony algorithm for multi-objective flexible job shop scheduling problems." The

[40] International Journal of Advanced Manufacturing

[41] Technology 55.9-12 (2011): 1159-1169.

[42] Ferrandi, F., Lanzi, P. L., Pilato, C., Sciuto, D., & Tumeo,

[43] A. (2010). Ant colony heuristic for mapping and scheduling tasks and communications on heterogeneous embedded systems. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 29(6), 911-924.

[44] Liu, S. L., Liu, Y. X., Zhang, F., Tang, G. F., & Jing, N.

[45] (2007). Dynamic web services selection algorithm with QoS global optimal in web services composition. Ruan Jian Xue Bao(Journal of Software), 18(3), 646-656.

[46] Jia, H. Z., Nee, A. Y., Fuh, J. Y., & Zhang, Y. F. (2003). A

[47] modified genetic algorithm for distributed scheduling

[48] problems. Journal of Intelligent Manufacturing, 14(3-4),

[49] 351-362.