Job Scheduling in Computational Grid using the Intelligence of Hybrid Fuzzy - Android System

N. Vijaya Raghavan
PG Scholar
Department of Computer science and Engineering
SriGuru Institute of Technology, Coimbatore-641110

R. Sujitha
PG Scholar
Department of Computer science and Engineering
SriGuru Institute of Technology, Coimbatore-641110

K.S. Suganya
PG Scholar
Department of Computer science and Engineering
SriGuru Institute of Technology, Coimbatore-641110

ABSTRACT
In the computational grid environment, algorithms specified for scheduling plays a vital role in managing the jobs. The main aim of the scheduling algorithms is to allocate the tasks to the availability at the mean time to the suitable resources. The makespan and cost for task execution can be minimized by an efficient task scheduling algorithm; it also helps to improve the load balancing among the resources in the grid environment. In recent days a major problem is, scheduling the independent tasks in a grid environment. In this paper, scheduling the independent task is taken as a challenge and a near optimal solution is obtained. Un-Prevail systematic grouping Genetic algorithm (UPSGA) is used by us to find the optimal solution for the task scheduling problem in grid environment. Fuzzy system is used to schedule the tasks indirectly to improve the load balancing between the resources. Dissimilarity based fuzzy crossover operator-II is proposed along with Android (friend map finder) for scheduling the tasks indirectly. Availability of resources and conflicts of costs provides the chance to cross over efficiently in fuzzy system. Near optimal solution for load balancing between resources are achieved with the help of makespan results.

Keywords
Grid Computing**,** Task Scheduling, Load balancing, Un-prevail systematic grouping Genetic algorithm, Fuzzy system, Android (friend map finder).

1. INTRODUCTION
Due to inventions in scientific research and the integration of the distributed system has shown a way to an emerging computing technology called grid computing, used for sharing the resources between large scale computing tasks. In grid computing, the computers are geographically distributed or it will be in a form of clusters of computers to share the computational resources in an efficient manner [1][2]. Load is to be balanced effectively between the resources to improve the device utilization and to reduce the makespan of the system, as it is the main aim of an efficient load balancing algorithm. Load balancing plays a vital role in minimizing the execution time and overcomes the problem related to overload of a device. In grid environment, scheduling algorithms used to balance the load and shows that all resource nodes results in equal performance, at the meantime the makespan is reduced and utilization of resources is improved[3].

In scientific and industrial problems computational problems are solved by computational grid in the form of resource sharing and coordination. Scheduling the tasks to the suitable resource became a difficult task. The reason behind this is the grid computing resources undergo some properties such as heterogeneous, dynamic and autonomous. To overcome these problems many research works were done by proposing new load balancing algorithms. To tackle the dynamic characteristics of the grid resource market model is used to minimize the scheduling problems based on user interest.

In this paper, to improve the performance and quality a heuristic genetic algorithm, UPSGA is used to find the optimal solution for minimizing the cost and execution time in task scheduling. By using the dissimilarity based fuzzy crossover operator-II task scheduling is done indirectly to balance the load in an effective manner. Mutating bits of a solution along with standard mutation is used based on the probability of bit mutation. The forth coming topics of this paper discuss about the related works, UPSGA approach, Encoding mechanism using UPSGA Approach, Dissimilarity based fuzzy crossover operator, Experimental results and Discussions and conclusion.

2. RELATED WORKS
There are various restrictions for users to schedule the tasks in grid environment due to the ancient scheduling algorithms. Solutions and quality obtained by those algorithms was not based on the user interest. For an instance, they concentrate only on load balancing and execution time, here user interest was ignored [1][2].

In [4], scheduling was done using Genetic Algorithm and variable neighborhood search without balancing the load by considering the cost and makespan. Marketing concepts are used in grid resource scheduling and management to make it as an effective economic model [5]. In [6], to improve the performance in the case of execution time and load balancing the market model is evaluated into two types of genetic algorithm. Particle Swarm Optimization (PSO) has been used to group the jobs in order to avoid the communication overhead and to improve the device utilization; at the meantime it minimizes the makespan of the system [7]. In [8][9] makespan and cost are not considered for balancing the load in computational grid, numerous load balancing approaches are considered to perform the scheduling in many levels by representing the grid environment in a tree structure. The minimized makespan and improved throughput was achieved by hierarchical layered architecture of grid computing services [10].For a distributed computing environment with heterogeneous property UPSGA approach is used to find the near optimal solution for the task scheduling problems. In [11], Mutation rate on diversity and quality of Pareto front has been studied to propose the fuzzy adaptive mutation to overcome the

scheduling problem in market based grid computing; here execution time, load balancing and cost were optimized using three dimensional optimization.

3. UPSGA APPROACH

In a multi objective optimization problem, Pareto-optimal solutions [12] are obtained by searching the solution space by UPSGA algorithm. A clear diversity preserving approach and elitist principle is used by the UPSGA. The UPSGA algorithm provides the Pareto-optimal solution by Pareto front and it stresses Un-Prevail solutions [13]. The subsequent approach of multiobjective techniques is obtained by Pareto Front. Pareto Front is defined as assigning the resources, at the same time a separate or a particular resource cannot be improved by degrading the performance of the any other resources. The use of clear diversity preserving approach is to provide a diversity rank to every separate users (individuals) as they are in similar un-prevail front and have the similar Un-Prevail rank in the population [12].The highest rank is given to the Un-prevail members in the minimum crowded area. A crowding distance metric is used to calculate the density of solutions around a specific solution in the population. This metric is obtained by calculating the average distance between two solutions based on their objectives [13].The Pareto-optimal solutions cannot be taken as best solution in the case any missing information regarding to multi-objective optimization [14].Pareto distribution is a distribution that measured as a name in economic theory. It is called Pareto distribution or power law distribution. For a distribution $F_x(x)$,such that

$$F_x(x) = P\{X>x\} = x^{-\alpha}, \text{for } x\geq \text{ where } 0<\alpha<2 \qquad (1)$$

The above equation is called as Pareto (α) distribution. The Pareto distribution undergo failure rate for $F_x x$.

$$F(x) = P\{X>x\} = x^{-\alpha},\ x\geq 1$$

$$F(x) = P\{X<x\} = 1-x^{-\alpha},\ x\geq 1$$

$$f(x) = dF(x)/dx = \alpha x^{-\alpha-1},\ x\geq 1$$

$$r(x) = f(x)/F(x) = \alpha x^{-\alpha-1}/x^{-\alpha} = \alpha/x,\ x\geq 1.$$

Therefore $r(x) = \alpha/x$ decrease with x, the Pareto distribution has decreasing failure rate. To overcome these failure rates, a bounded Pareto distribution is used. When we look for a curve fir to the measured data, we observe that measured data have a minimum job lifetime and maximum job lifetime. The measured data have all finite moments. To model the measured data, we therefore want a Pareto shape, but truncated on both ends. We refer to such a distribution as a Bounded Pareto distribution. The bounded Pareto (K,P,α) distribution has density function.

$$f(x) = \alpha x^{-\alpha-1}.K^{\alpha}/1-(K/P)^{\alpha}, \text{ for } K \leq x \leq P \text{ and}$$

$$0<\alpha<2.$$

In [15], Pareto front is obtained by user's choice; in this regard the solution obtained is dominant and suitable. A dissimilarity based fuzzy crossover operator with UPSGA is used in parallel distributed computing systems to overcome the problems in independent task scheduling. Certain issues are addressed in this paper. Fuzzy function is used indirectly to balance the load. In this case, the cost and execution time is compared with both UPSGA without fuzzy logic and UPSGA with fuzzy logic. In UPSGA with dissimilarity based fuzzy crossover, inputs for fuzzy function are dissimilarity between costs of individuals and percentage of occurrence of available resources in scheduling.

4. ENCODING MECHANISM USING UPSGA APPROACH

The problem has been designed in the form of vector of images for each solution as it is encoded in the coding scheme. For an instance, take n tasks and m resources, the n tasks are formulated as set of solution variables (chromosome). The set of solution variable (chromosome) contain a solution variable (Gene) in each cell of vector. Here the tasks are allocated to resources in between 1 and m.

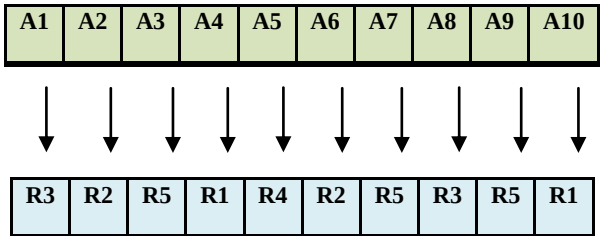

Fig. 1 Set of variables (chromosome) in coding scheme

Each cell of vector is allocated with random number between 1 and m of size n for p times. This allocation is used to generate set of possible values for the solution variable (Population) with p individuals (gene).The ultimate goal our algorithm is to minimize the cost of the users and to obtain the minimum completion time in scheduling the task. According to this problem, cost and makespan are dissimilar to one another this leads to increase in makespan as the cost decreases.

4.1 Makespan

By generally speaking about the task scheduling, the completion time or makespan is the first ultimate goal of our algorithm. Makespan is defined as the time taken by the last system to complete it tasks, it means maximum time taken by a processor to complete it tasks when comparing with other processors [1][3]. Let us assume K_a and G_b be the size of the task *a* and b be the speed of the processor as it is a resource. Therefore, the running time or execution time of the task *a* on the resource *b* can be done as

$$C_{exe}(a,b)=K_a/G_b \qquad (2)$$

There will be completion time for every processor or systems for tasks allocated to them; the completion time is calculated individually for each processor. For an instance, the completion time for each processor is showed in fig 2 based upon fig 1.

Table I Tasks and their Sizes

Tasks	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
Size	25	42	48	18	22	39	27	23	46	31

Table II Resources and their Speeds

Number of Resources or processors	1	2	3	4	5
Speed of Resources or Processors	1.2	6.9	3.7	1.8	4

The calculation of execution time of allocated resource for each task is shown below:

Cexe(1,3) = 25/3.7 = 6.8
Cexe(2,2) = 42/6.9 = 6.1
Cexe(3,5) = 48/4 =12
Cexe(4,1) = 18/1.2 = 15
Cexe(5,4) = 22/1.8 = 12.2
Cexe(6,2) = 39/6.9 = 5.6
Cexe(7,5) = 27/4 = 6.7
Cexe(8,3) = 23/3.7 = 6.2
Cexe(9,5) = 46/4 = 11.5
Cexe(10,1) = 31/1.2 = 25.8

Makespan is calculated using the formula.

Makespan = Max [$C_{complete}(b)$] where $1 \leq b \leq m$ (3)

Our goal is to reduce the makespan, i.e tasks allocated to the resources should be completed in minimum or shortest time. According to our problem, Resource 1 takes maximum time, so the makespan is 40.8. Now we have to reduce the makespan. Makespan of resources which is shown in Fig 2.

R1 = 15 for task 4 & 25.8 for task 10,

R2 = 6.1 for task 2 & 5.6 for task 6,

R3 = 6.2 for task 8 & 6.8 for task 1,

R4 = 12.2 for task 5,

R5 = 11.5 for task 9 & 6.7 for task 7 & 12 for task 3

Calculation of Makespan for Resources:

Ccomplete(1) = 15+25.8 = 40.8

Ccomplete(2) = 6.1+5.6 = 11.7

Ccomplete(3) = 6.2+6.8 = 13

Ccomplete(4) = 12.2

Ccomplete(5) = 11.5+6.7+12

Therefore makespan = 40.8

4.2 Reducing the Cost

The cost is collected by the resource providers from the users based on the capacity of resource requested or utilized by the users In market based grid environment, the scheduling algorithms takes an account of users wish to finish their work in most economical way [8].

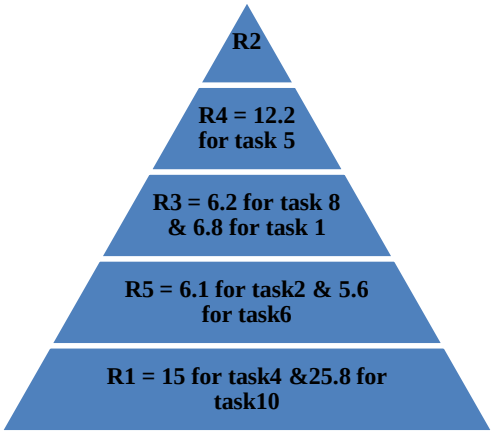

Fig 2.Makespan for Resources

Our next goal other than makespan is to reduce the cost. Let us assume P_b be the component cost for resource b, then the running or execution cost of the task a on the resource b can be calculated using the following formula.

Cost (b) = Ccomplete(b) * P_b (4)

Therefore overall cost for scheduling is calculated using the formula given below

Overall Cost = $\sum_{1 \leq b \leq m}$ Cost (b) (5)

4.3 Improving the Load Balance

To show the improvement in load balancing using our approach, we have to appraise and collate the load balancing functions and load balancing results of the algorithms. The term load balancing is defined as dividing and sharing the load equally among the computational resources and increasing the device or resource utilization at the mean time running or execution time is reduced. To obtain these objectives, our load balancing technique should prove the fairness in diving and sharing the load across computational resources. In this regard the dissimilarity must be reduced between the large loaded resource and small loaded resource. State the average resource or device utilization, by identifying the maximum average resource or device utilization we can conclude that the dividing and sharing the load is well balanced among all computational resources in the grid environment [1]. By dividing the total number of resource or device utilization by the total number of computational resources present in the environment, average resource or device utilization can be calculated [1]. Prospective utilization of each computational resource is calculated based on the allocated tasks. Therefore utilization of resource or device is calculated by dividing the completion time of the task at every individual node by the makespan.

$U_u(b)$ =Ccomplete(b)/Makespan$1 \leq b \leq m$ (6)

Therefore the average device or resource utilization is calculated using the formula given below:

$\bar{U} = (\sum_{1 \leq b \leq m} U_u(b))/m$ (7)

To improve the load balancing among all the nodes, the reduced mean square deviation of $U_u(b)$ is calculated using the formula given below

$U_{msd} = [(\sum_b (U_u(b) - U_e)^2) / m]^{0.5} 1 \leq j \leq m$ (8)

Every existing approach algorithms attain the Pareto-optimal front based on Mean Square Deviation. The reason behind the usage of Pareto optimization is, in fact the computing workloads have highly variable job sizes that are not well described by an exponential distribution. In a distributed environment, the solution of the problem space is analyzed by Pareto optimization. The Pareto Front mainly focuses on economic criteria based on the user's wish. At sometimes the uniform distribution is not possible in resource allocation, in this regard Pareto analysis is performed. Pareto distribution is used in the area of multi criteria decision making. Pareto distribution is used to optimize more than one objective function simultaneously. The comparisons of these results are shown in experimental section.

5. DISSIMILARITY BASED FUZZY CROSSOVER OPERATOR

Dissimilarity based Fuzzy Crossover Operator a new approach is used in this paper and compared with standard Cross over operator. As a result, the dissimilarity based Fuzzy Crossover Operator is proved to show that it is superior to others. Dissimilarity of costs and occurrence of number or percentage

of resources in scheduling is calculated by two functions. The percentage or number of obtainable resources in scheduling is calculated first by the formula given below.

$$\sigma_1^{\,2}(H) = (\sum_b (J_b - \alpha)^2) / m \text{ H€Pareto front} \qquad (9)$$

Here J_b specifies the frequency of resource *b* in present scheduling or J_b is denoted as number or percentage of tasks allocated to resource *b*, number of resources is denoted by m and number of each user in Pareto Front is denoted by n. Each existing user takes Pareto front as input and the dissimilarity of frequency among the resources in scheduling is calculated in the first function. Number of tasks divided by the number of resources is equal to the average of frequency of occurrence of any resources in scheduling. Therefore, from these frequencies we can calculate number or percentage of utilized resources in scheduling. The value of the Pareto front of every user is generated by the function (9) and it states number or percentage of occurrence of all resources in one scheduling. It is calculated using the given formula

$$\text{AvgSD} = (\sum_H \sigma_1^{\,2}(H)) / n \text{ H€Pareto front} \qquad (10)$$

There is an occurrence of maximum value and it is based upon number of tasks and resources. In this regard, values should be normalized within the interval [0, 1] as it results in the average of these values. This is performed using the formula given below

$$Z = (\text{Average Standard Deviation} - \text{Average Standard Deviation}^{min})/(\text{Average Standard Deviation}^{max} - \text{Average Standard Deviation}^{min}) \qquad (11)$$

For example, a low dissimilarity of frequency of liken resources states that occurrence of any resource or device needs to allocate the resources to the task equally. Thus the output of the fuzzy system results in direct decision making. Therefore to improve the load balancing in task scheduling problem, these function used along with makespan. The Pareto front members are the input as well as the arguments for second function. The dissimilarity in average fitness is calculated by the following expression.

$$\sigma_2^{\,2} = (\sum_H (R_H - \lambda)^2 / n \text{ H€Pareto front} \qquad (12)$$

It happens due to different members based upon their costs in all objectives, thus the output of the fuzzy system is inversed. The values are generated in the interval [v, x] by this function. For mapping and normalizing the values in the interval [0, 1] the following expression is used.

$$D = (\sigma_2^{\,2} - v) / (x - v) \qquad (13)$$

The average of cost and makespan of each user in the Pareto Front is denoted by λ, where R_H is denoted as average of cost and makespan of member *H* from the present Pareto Front. These results are given as input to the fuzzy system as the output is generated randomly by performing Crossover in the set of possible values for the solution variable.

6. ANDROID (FRIEND MAP FINDER)

The scheduling of jobs is done in hybrid fuzzy system along with Android (Friend Map Finder). The Android application is installed in the grid environment [16]. The friend map finder is used to identify the local minima and maxima of the neighbor node in grid environment. Performance evaluation for friend map finder is in process.

7. EXPERIMENTAL RESULTS AND DISCUSSIONS

In our experiments, three task scheduling algorithms are compared along with their results. The algorithms are MOPSO, UPSGA without dissimilarity based fuzzy and UPSGA with dissimilarity based fuzzy system along with two goals cost and makespan. The input and output for the membership function is given in the Fig 3, 4 and 5. The rules are generated according to if then rules of fuzzy system for this operator and it are shown in Fig 6.

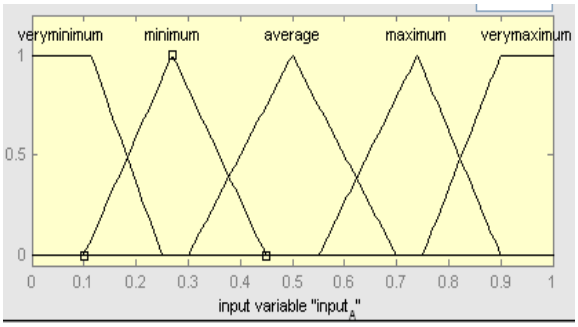

Fig. 3 Input membership function: A

The input of A contains two membership functions, triangular and trapezoidal. The triangular membership function is used for medium, average and minimum makespan. The trapezoidal member function is used for very minimum and very maximum makespan. The interval is set between [0, 1] for both the axes.

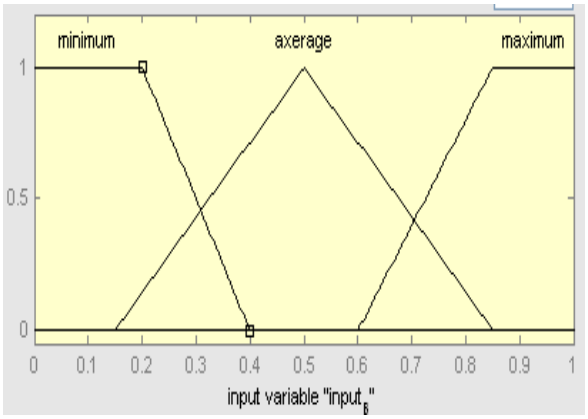

Fig. 4 Input membership function: B

The input of B contains two membership functions, triangular and trapezoidal. The triangular membership function is used for average cost. The trapezoidal member function is used for minimum and maximum cost. The interval is set between [0, 1] for both the axes.

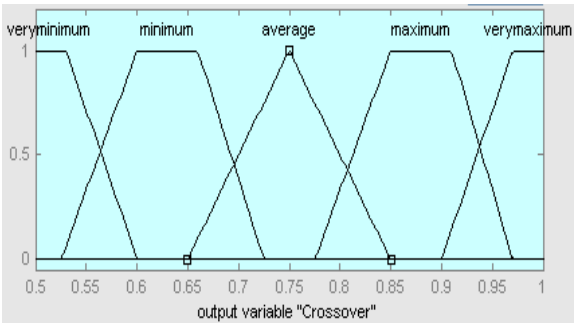

Fig. 5 Output membership function: Fuzzy Crossover

The output Fuzzy Crossover operator contains two membership functions, triangular and trapezoidal. The triangular membership function is used for average crossover. The trapezoidal member function is used for very minimum, very maximum, minimum and maximum crossover. The interval is set between [0, 1] for y axis and [0.5, 1] for x axis.

1. If (input_A is veryminimum) and (input_B is minimum) then (Crossover is average) (1)
2. If (input_A is veryminimum) and (input_B is axerage) then (Crossover is minimum) (1)
3. If (input_A is veryminimum) and (input_B is maximum) then (Crossover is veryminimum) (1)
4. If (input_A is minimum) and (input_B is minimum) then (Crossover is average) (1)
5. If (input_A is minimum) and (input_B is axerage) then (Crossover is minimum) (1)
6. If (input_A is minimum) and (input_B is maximum) then (Crossover is veryminimum) (1)
7. If (input_A is average) and (input_B is minimum) then (Crossover is maximum) (1)
8. If (input_A is average) and (input_B is axerage) then (Crossover is average) (1)
9. If (input_A is average) and (input_B is maximum) then (Crossover is minimum) (1)
10. If (input_A is maximum) and (input_B is minimum) then (Crossover is verymaximum) (1)
11. If (input_A is maximum) and (input_B is axerage) then (Crossover is maximum) (1)
12. If (input_A is maximum) and (input_B is maximum) then (Crossover is average) (1)
13. If (input_A is verymaximum) and (input_B is minimum) then (Crossover is verymaximum) (1
14. If (input_A is verymaximum) and (input_B is axerage) then (Crossover is maximum) (1)
15. If (input_A is verymaximum) and (input_B is maximum) then (Crossover is average) (1)

Fig.6 If then rules of fuzzy system

The proposed approach proved in the improvement of load balancing indirectly and provides the fastest Pareto-optimal solutions with best quality. The Pareto front is generated for each and every user present in the population in a minimum number of iterations. In another dimension we identified that Un-Prevail systematic grouping genetic algorithm provides better solution than particle swarm optimization with multi-objective.

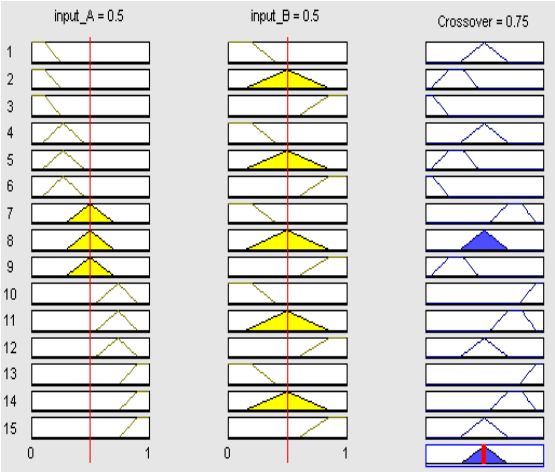

Fig..7 Rule viewer of fuzzy system.

In this paper, dissimilarity based fuzzy system is used to produce different solutions for the solution space. The changeable and adjustable rate of crossover in the proposed approach improves the investigation in solution space in the case of low value from different solutions. In this regard there is chance of dissimilarity in fitness of the solutions.

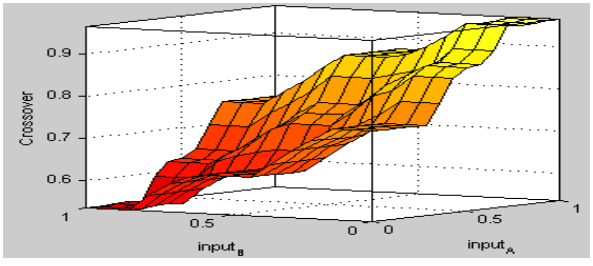

Fig.8 Three dimensional view of Crossover

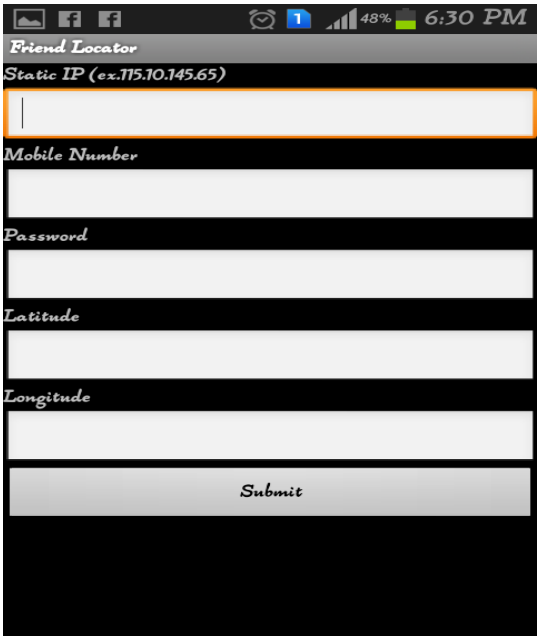

Fig. 9 Friend Map Finder home page

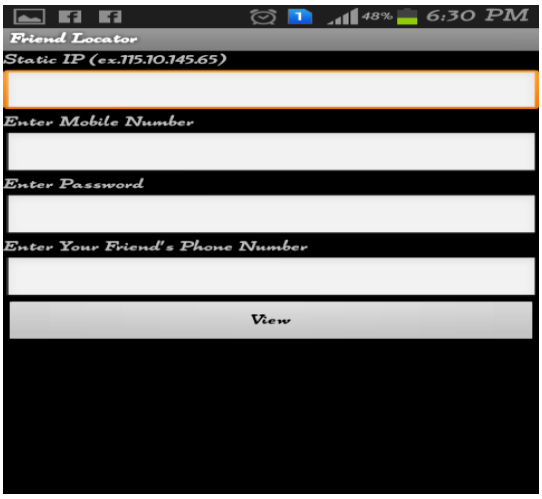

Fig. 10 Friend Locator identifies the nearest neighbor

This leads to the generation of Pareto front as fast as possible in minimum iteration. The fitness value is measured based on the value of cost. If the value of cost is low, then the solutions are same. The set of possible values of solution variable is investigated more to obtain many different solution using the fuzzy system. The Pareto-optimal front is obtained in minimum iteration by different solutions generated by fuzzy system. The makespan is optimized by using the proposed approach and it improves the load balancing.

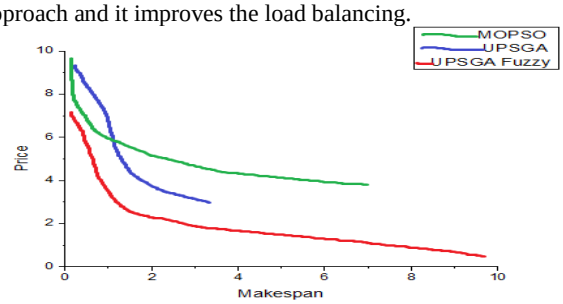

Fig. 11 MOPSO VS UPSGA VS UPSGA Fuzzy

As a result, tasks are allocated equally to the resources. Finally with the help of UPSGA with dissimilar based fuzzy crossover operator, the completion time (makespan) and cost is reduced. Comparison graph in Fig. 9 shows that our UPSGA approach with fuzzy system produces lowest completion time and low cost when compare to other approaches such as MOPSO and UPSGA. The values of existing system results are given in the following Fig. 10 and Fig. 11

X	Y	X	Y
0.231991E+00	0.717949E+01	0.259740E+00	0.931728E+01
0.156926E+01	0.249084E+01	0.454545E+00	0.847800E+01
0.180917E+01	0.238095E+01	0.649350E+00	0.799446E+01
0.222915E+01	0.223443E+01	0.746753E+00	0.775268E+01
0.269138E+01	0.208791E+01	0.811588E+00	0.747292E+01
0.297858E+01	0.179487E+01	0.941558E+00	0.699245E+01
0.360877E+01	0.164835E+01	0.974026E+00	0.679329E+01
0.411928E+01	0.164835E+01	0.107143E+01	0.627483E+01
0.456951E+01	0.157509E+01	0.110390E+01	0.595709E+01
0.513976E+01	0.146520E+01	0.120130E+01	0.524100E+01
0.555996E+01	0.139194E+01	0.129870E+01	0.488065E+01
0.739058E+01	0.989011E+00	0.149351E+01	0.447616E+01
0.784059E+01	0.842491E+00	0.165584E+01	0.387557E+01
0.835099E+01	0.805861E+00	0.185065E+01	0.362918E+01
0.886106E+01	0.659341E+00	0.207792E+01	0.330219E+01
0.910108E+01	0.586081E+00	0.266234E+01	0.280016E+01
0.937135E+01	0.586081E+00	0.295455E+01	0.270725E+01
0.949125E+01	0.512821E+00	0.331169E+01	0.245316E+01
0.976119E+01	0.402930E+00	0.311588E+01	0.254145E+01

Fig. 12 Values of UPSGA with Fuzzy and without Fuzzy (without Pareto Front)

X	Y
0.180504E+00	0.974331E+01
0.179396E+00	0.937566E+01
0.208188E+00	0.893459E+01
0.175410E+00	0.805211E+01
0.295007E+00	0.775844E+01
0.354473E+00	0.750130E+01
0.444061E+00	0.724428E+01
0.533094E+00	0.680343E+01
0.622582E+00	0.654640E+01
0.772400E+00	0.625284E+01
0.952792E+00	0.614321E+01
0.128290E+01	0.574001E+01
0.167359E+01	0.544733E+01
0.203404E+01	0.511777E+01
0.245540E+01	0.500903E+01
0.305704E+01	0.475388E+01
0.495288E+01	0.417262E+01
0.591597E+01	0.391881E+01
0.684938E+01	0.381194E+01

Fig.13 Values of MOPSO without Pareto Front

The Comparisons of MOPSO, UPSGA and UPSGA

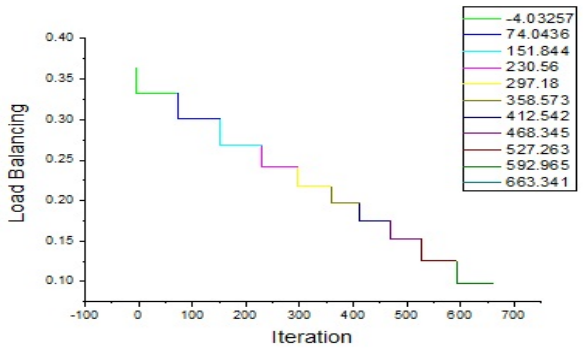

Fig.14 Load Balancing Using Pareto Front (MOPSO)

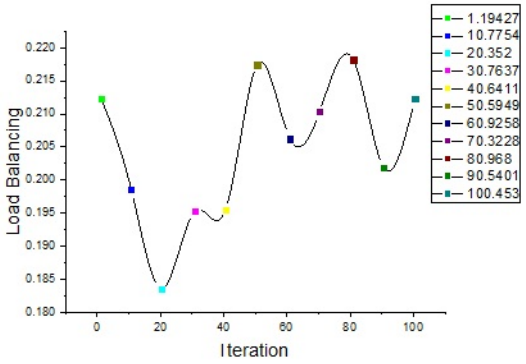

Fig.15 Load Balancing Using Pareto Front (UPSGA)

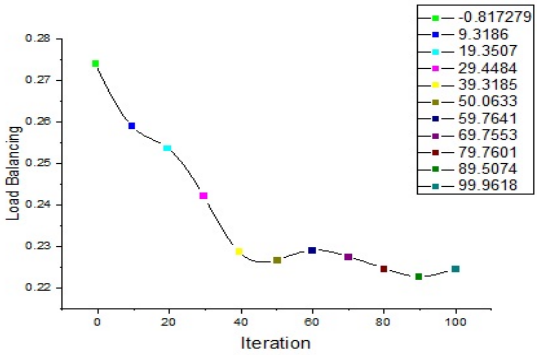

Fig.16 Load Balancing Using Pareto Front (UPSGA with Fuzzy)

8. CONCLUSION

In this paper, UPSGA with dissimilarity based fuzzy crossover operator II has been implemented in grid environment to schedule the task in an effective manner and compared with UPSGA without fuzzy based crossover and MOPSO. Our proposed algorithm proved that it performs better than existing approach and improves the quality of scheduling. In terms of Pareto front, it is obtained by UPSGA with dissimilarity based fuzzy crossover system and it maintains equal spread of solutions in the Pareto-optimal front. There is an increase in distribution and relation of solutions in the proposed approach compared to others. As a result, the performance and quality is improved by enhancing the intelligence of hybrid Fuzzy system using the proposed algorithm in the case of an adjustable environment.

9. ACKNOWLEDGMENT

I thank G. Sudha Sadasivam, for her valuable suggestions and ideas through her paper regarding to task scheduling in Grid computing. I thank management and Department of computer science and Engineering of Kongu Engineering College for allowing me to use grid computing lab to attain the results of my paper. I thank P. Marikkannu for his valuable ideas and suggestions through his paper regarding to load balancing between the resources in grid computing and also in Fuzzy system results.

10. REFERENCES

[1] Y. Li, Y. Yang, and R. Zhu, A Hybrid Load balancing Strategy of Sequential Tasks for Computational Grids, International Conference on Networking and Digital Society, IEEE, 2009.

[2] M. I. Daoud and N. Kharma, A high performance algorithm for static task scheduling in heterogeneous distributed computing systems, Journal of Parallel and Distributed Computing (Elsevier), Volume 68, Issue 4, 2008, pp. 399–409.

[3] G. Bronevich, and W. Meyer, Load balancing algorithms based on gradient methods and their analysis through algebraic graph theory, Journal of Parallel and Distributed Computing (Elsevier), Volume 68, Issue 2, 2008, pp. 209–220.

[4] S. Kardani-Moghaddam, F. Khodadadi, R. Entezari-Maleki, and A. Movaghar, A Hybrid Genetic Algorithm and Variable Neighborhood Search for Task Scheduling Problem in Grid Environment, International Workshop on Information and Electronics Engineering (IWIEE), Procedia Engineering 29, 2012 , pp. 3808 – 3814.

[5] R. Buyya, J. Giddy, and D. Abramson, An Evaluation of Economy-based Resource Trading and Scheduling on Computational Power Grids for Parameter Sweep Applications, Proceedings of the 2nd International Workshop on Active Middleware Services (AMS 2000), August 1, 2000, Pittsburgh, USA, Kluwer Academic Press, 2000.

[6] A. Omara, and M. M. Arafa, Genetic algorithms for task scheduling problem, Journal of Parallel and Distributed Computing (Elsevier) Volume 70, Issue 1, 2010, pp. 13–22.

[7] S. Sadasivam, and V. Rajendran. V, An Efficient Approach To Task Scheduling In Computational Grids, International Journal of Computer Science and Applications, Techno mathematics Research Foundation Vol. 6, No. 1, pp. 53 – 69, 2009.

[8] S. Prakash, and D. P. Vidyarthi, Load Balancing in Computational Grid Using Genetic Algorithm, Advances in Computing, DOI: 10.5923/j.ac.02, 1(1) pp. 8-17, 2011.

[9] Chandra Patni, M. S. Aswal, O. Prakash Pal, and A. Gupta, Load balancing Strategies for Grid Computing, IEEE, pp. 239-243, 2011.

[10] Touzene, S. Al-Yahai, H. AlMuqbali, A. Bouabdallah, and Y. Challal, Performance Evaluation of Load Balancing in Hierarchical Architecture for Grid Computing Service Middleware, IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 2, ISSN: 1694-0814, 2011.

[11] R. Salimi, H. Motameni, and H. Omranpour, "Task Scheduling with Load Balancing for Computational Grid Using NSGA II with Fuzzy Mutation", 2012 2nd IEEE International Conference on Parallel, Distributed and Grid Computing, pp. 79-84, 2012.

[12] Deb, S. Agrawal, A. Pratap, and T. Meyarivan, A fast and elitist multi-objective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6, pp. 182–197, 2002.

[13] K. Madavan, "Multiobjective Optimization Using a Pareto Differential Evolution Approach," IEEE , 2002.

[14] T. B. Khoo, B. Veeravalli, T. Hung, C. W. S. See, A multi-dimensional scheduling scheme in a Grid computing environment, Journal of Parallel and Distributed Computing (Elsevier), Volume 67, Issue 6, 2007, pp. 659–673.

[15] S. Padhee, N. Nayak, S.K. Panda, and S.S. Mahapatra, Multi-objective parametric optimization of powder mixed electro-discharge machining using response surface methodology and non-dominated sorting genetic algorithm, Indian Academy of Sciences, Sadhana Vol. 37, Part 2, pp. 223–240, 2012.

[16] L. Blazevic, S. Giordano, and J.-Y. LeBoudec, "A Location Based Routing Method for Mobile Ad Hoc Networks," IEEE Trans.Mobile Computing.