

Swarm Intelligence based Genetic Algorithm for Multi-Objective Optimization

N. Aravindhu, S. Siva Sathya

Department of Computer Science,

Ramanujan School of Mathematics and Computer Science,

Pondicherry University, Pondicherry, India.

aravindhun18@yahoo.com, sivasathya@yahoo.com

Abstract – Multi-objective optimization forms realistic models for many complex engineering problems. Different types of genetic algorithm solutions have been provided to solve these problems. In this paper a new Swarm Intelligence based Genetic Algorithm (SIGA) is proposed to overcome the disadvantages faced in the previous approaches. The example problem chosen is Multi-objective human resource allocation problem. This problem has already been solved by hybrid genetic algorithm. The newly proposed SIGA outperforms the previous one. The simulation results provided at the end of this paper proves this work.

Index Terms - Multi-objective optimization; Multi-objective human resource allocation problem; Particle swarm optimization; Swarm based genetic algorithm.

I. INTRODUCTION

Since early 1960's multi-objective optimization have received great interest from researchers with various backgrounds. In these problems, multiple objectives need to be optimized simultaneously. It is impossible to obtain a single solution that is best with respect to all objectives. Objectives conflict with each other. A solution which is best with one objective may be worst for another. Therefore there exists a set of solutions for multi objective optimization. The solutions thus obtained are called as non-dominated or pareto optimal solutions and they cannot be compared with each other. For such solutions, it is not possible to improve one objective function without sacrificing atleast one of the other objective functions [1].

In the recent years, genetic algorithms have been mostly applied for solving multi-objective optimization problems. The basic nature of genetic algorithms is to perform multi directional and global searches. The population of the best solutions is maintained from generation to generation. This approach is beneficial in exploring pareto optimal solutions. GA can be applied to a wide range of optimization.. One of the useful features of GA is to handle multi-objective function optimization. GA searches from population of points that can provide globally optimal solutions [2].

Numerous types of genetic algorithms are available for solving multi-objective optimization. Certain disadvantages can be identified from these algorithms. Some of them are listed here. Sub-population genetic algorithm (SPGA) [3] uses weighting sum approach and problem occurs when improper weights are assigned. Multi-objective Tchebycheff genetic algorithm (MOTGA) [4] which is very much useful in solving multi

dimensional knapsack problem can handle only upto two objectives. New multi-objective genetic algorithm (NMOGA) [5] is applied for chemical reaction networks and is purely domain specific. Elitist multi-objective genetic algorithm (EMOGA) [6] is applied for data mining and its complexity level is too high. Hybrid Genetic Algorithm (HGA) [7] is applied for human resource allocation problem and it suffers from local optimality search.

In this paper we propose swarm intelligence based genetic algorithm for solving multi-objective optimization. It overcomes the disadvantages of the previous approaches. The particle swarm optimization [8] derived from swarm intelligence was developed by Kennedy and Eberhart. A group (swarm) of particles moves in discrete intervals through the search space. Particles represent solution instances in the search space. Each particle keeps track of the best solution it encountered in its path which is called as the local best. The best location identified by all particles in called as the global best. The next move of the particles is controlled by the particle best and the global best.

Resource allocation problem (RAP) [7] is the process of allocating resources among the various projects or business units for maximization of profit and minimization of cost. The chosen resource may be a person, asset, material or capital to accomplish the goal. A goal may be an objective derived for the future financial needs. The best solution obtained may maximize profit, minimize cost or achieves the best quality. The example problem chosen in this paper is Multi-objective human resource allocation problem (MOHRAP). The objectives considered here are minimization of cost and maximization of efficiency. This problem is already solved by multi-stage decision based hybrid genetic algorithm (moHGA) [7] by Chi-Ming Lin and Mistuo Gen. Our objective is to provide a simpler strategy that produces better results than the previous one.

This paper is organized as follows. The proposed algorithm for SIGA is described in section III. Section IV provides a numerical example for the problem and the results obtained are compared with moHGA.

II. PROBLEM FORMULATION

In MOHRAP [7] our major task is to assign 'n' number of workers to 'm' number of jobs. The objectives considered are minimization of cost and maximization of benefit. Optimization deals with the problem of seeking solution over a set of choices without loss of generality.

Formulation for the problem is given as follows:

Minimizing the cost

$$\text{Min } Z_1(x) = \sum_{i=1}^n \sum_{j=1}^m C_{ij} X_{ij} \quad (1)$$

Maximization of benefit

$$\text{Max } Z_2(x) = \sum_{i=1}^n \sum_{j=1}^m E_{ij} X_{ij} \quad (2)$$

i – Index of job, $j=1,2,\dots,n$

j – number of workers, $j=0,1,2,\dots,n$

n – Total number of jobs

m – Total number of workers

C_{ij} – Cost of job i when j workers are assigned

E_{ij} – Efficiency of job i when j workers are assigned

$$X_{ij} = \begin{cases} 1, & \text{if } j \text{ worker s are assigned to job } i \\ 0, & \text{otherwise} \end{cases}$$

III. PROPOSED METHODOLOGY

In this paper we propose swarm intelligence based genetic algorithm (SIGA) which overcomes the disadvantages of the previous algorithms. The example problem chosen to implement this algorithm is multi-objective human resource allocation problem.

The processing steps of SIGA are as follows

- Step 1: Generate the initial population P.
- Step 2: Subdivide the population into 'm' subpopulation according to the number of objectives (m)
- Step 3: For each subpopulation 'S' do the following steps.
 - Step 3.1: Evaluate the fitness based on the objectives assigned to each subpopulation.
 - Step 3.2: Select the best chromosome 'X' from 'S'.
 - Step 3.3: Select two chromosomes 'P' and 'Q' from 'S'.
 - Step 3.4: Apply crossover between 'P' & 'Q'. Let $O_{1\text{Best}}$ be the best offspring.
 - Step 3.5: Apply crossover between $O_{1\text{Best}}$ and 'X'. Let $O_{2\text{Best}}$ be the best offspring.
 - Step 3.6: if $O_{2\text{Best}}$ is better than 'X' then replace $O_{2\text{Best}}$ with 'X'.
 - Step 3.7: Iterate steps (3.1 – 3.5) until all chromosomes are considered.
 - Step 3.8: Apply fabrication operator to all low fit offsprings obtained from crossover in steps 3.4 and 3.5.
- Step 4: The best chromosome in each subpopulation is compared for best solutions.
- Step 5: Iterate steps 3 & 4 until best pareto optimal solutions are obtained.

In the first step, generate the initial population randomly. The structure of the chromosomes is shown in fig. 1. Then in the next step, subdivide the population according to the specified objectives. For each subpopulation evaluate the fitness of the chromosomes. The fitness is evaluated based on the specified objectives

for each subpopulation. Select the best chromosome as 'x'.

In the next stage the best chromosomes are selected for mating. The selection operator is very much useful in improving the quality of the population. It gives the higher quality chromosomes a better chance to get copied into the next generation. In this paper, the roulette wheel selection mechanism is used.

The next step is to apply crossover operators. In this paper the uniform crossover operator [1] is used (refer subsection A). Select two chromosomes randomly from the mating pool. Apply crossover between the two chromosomes. The best of the individual is considered as 'O1best'. Then the 'O1best' is again made to crossover with 'x'. Let the best individual selected be 'O2best'. Compare 'O2best' with 'x'. If 'O2best' outwits 'x' then replace 'x' with 'O2best'. Continue this process for all the subpopulations until all the individuals in the mating pool are considered.

Fabrication operator [9] (refer subsection B) is applied to improve the performance of the low fit chromosomes. The low fit offsprings obtained during crossover are collected and fabrication operator is applied to it. The best chromosomes selected during the crossover operation and the chromosomes obtained after applying the fabrication operator are copied to next generation. Repeat these steps for all the subpopulations. At the end of each generation the best solution obtained from each subpopulation are compared for best pareto optimal solutions.

A. Uniform crossover operator

Uniform crossover operator proposed by Syswerda [1] is used in this paper (fig. 1). This method first generates a random mask and then exchanges relative gene under the mask between parents. A mask is simply a binary string with the same size chromosome. The parity of each bit in the mask determines from which parent the offspring will receive each gene. When there is a 1 in the crossover mask, then the gene is copied from the first parent and if there exists a 0, then the gene is copied from the second parent. In this approach, the goal of the crossover is to exchange information between two parent individuals to produce new offspring for the next generation.

B. Fabrication Operator

To improve the performance of low fit chromosomes fabrication operator [9] proposed by Chang, Wang and Liu is used. The fabrication operator is used to generate a new set of chromosomes by collecting information from the low fit chromosomes. A dominance matrix is formed by the voting process. In the matrix the job represents the gene and the sequence represents the position each job is assigned. Then the voting is done by counting the number of occurrences of each gene. In the next step first the position of gene is determined randomly. Then copy less than 50% of the genes to the new chromosome. Only the gene which has highest occurrence is copied to the new chromosome. Then the rest of the genes are randomly inserted to the chromosomes. The chromosomes obtained through this process produced best fitness than the low fit chromosomes.

Random Mask	0	0	1	0
Parent 1	8	6	5	3
Parent 2	7	2	4	1
Offspring 1	8	2	6	1
Offspring 2	7	5	4	3

Figure 1. Uniform Crossover Operator

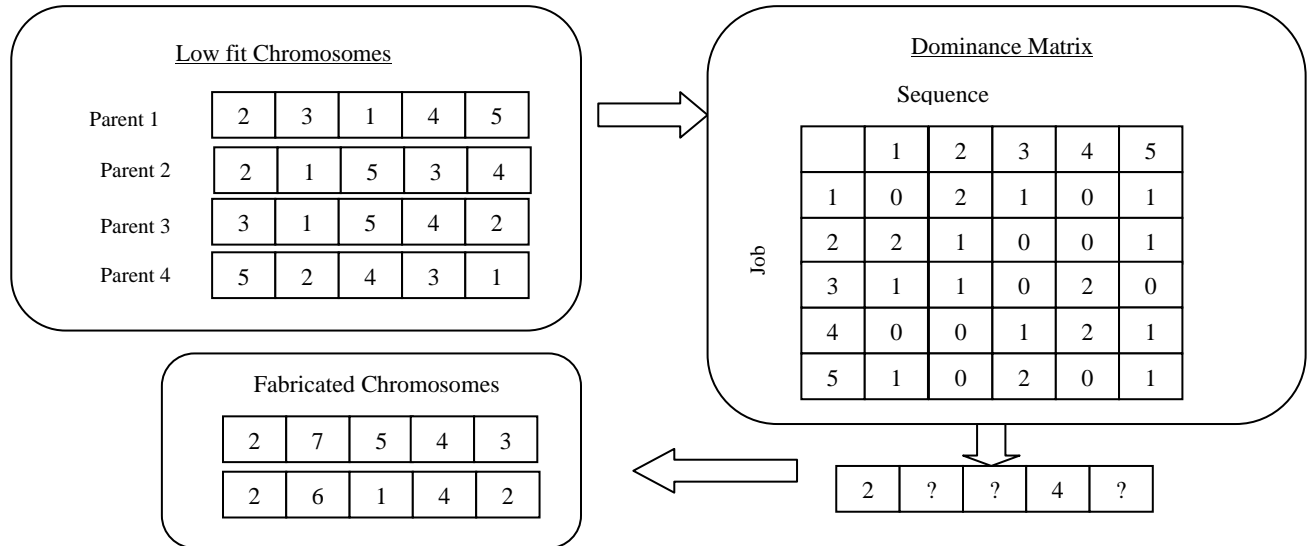


Figure 2. Fabrication Operator

IV. EXPERIMENTAL RESULTS

In this paper we have chosen problem with 10 workers allocated for a set of 4 jobs. The expected cost is given in table 1. The expected efficiency is given in table 2. The data specified here is from moHGA [7]. The experimental results are given in table 3. It is inferred from table 3 that the overall cost obtained by applying SIGA is lesser than moHGA and the overall efficiency is greater than

moHGA. Moreover the difference between the cost and efficiency is maximum in case of SIGA when compared with moHGA. In fig. 3 the triangle represents the results obtained from SIGA and the square represents the results obtained from moHGA. From this we can infer that the curve obtained by SIGA is far better than moHGA. So from table 3 and fig. 3 we can prove that SIGA obtains far better results than moHGA.

Table 1: Expected cost c_{ij}

Number of Jobs: i	Workers: j										
	0	1	2	3	4	5	6	7	8	9	10
1	41	38	46	32	78	76	72	84	80	92	96
2	45	54	36	55	87	82	90	132	97	121	134
3	36	43	68	56	72	59	32	67	86	88	100
4	46	78	88	64	90	80	120	104	96	86	120

Table 2: Expected efficiency e_{ij}

Number of Jobs: i	Workers: j										
	0	1	2	3	4	5	6	7	8	9	10
1	0	37	42	50	54	56	58	65	72	80	95
2	0	49	55	59	62	67	73	80	87	95	102
3	0	45	49	57	64	77	88	92	100	105	110
4	0	60	67	72	79	83	88	97	102	110	120

Table 3: Comparative results

Solution k	moHGA						SIGA					
	jx_{1j}	jx_{2j}	jx_{3j}	jx_{4j}	Overall cost	Overall efficiency	jx_{1j}	jx_{2j}	jx_{3j}	jx_{4j}	Overall cost	Overall efficiency
1	3	2	1	4	201	229	1	2	6	1	184	240
2	0	2	6	2	197	210	3	1	6	3	182	259
3	3	2	5	0	173	182	1	2	6	3	170	252
4	1	1	6	2	212	241	1	3	6	3	189	256
5	1	1	5	2	215	235	1	2	1	5	191	233
6	0	1	6	3	191	209	2	2	6	3	178	257

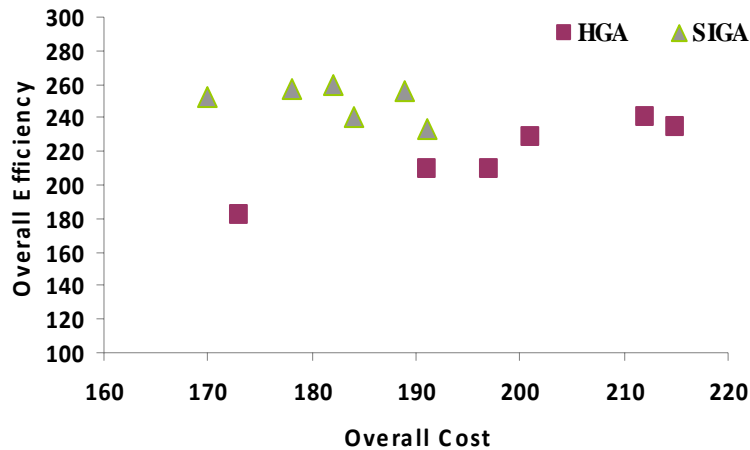


Figure 3. Comparative Analysis Graph

V. CONCLUSION

In this paper a new type of algorithm SIGA has been proposed to overcome the disadvantages of the previous methodologies. The problem chosen is MOHRAP. This problem is already solved by moHGA. SIGA obtained best pareto optimal solutions when compared to moHGA. A comparative analysis table and graph is provided at the end of this paper. From this we can conclude that SIGA obtains far better results than the moHGA. In future we can extend this algorithm for solving numerous complex multi-objective optimization problems.

REFERENCES

- [1] M. Gen, R. Cheng, "Genetic Algorithms and Engineering Optimization," John Wiley & Sons, Inc., New York, 2000.
- [2] M. Gen, R. Cheng, "Genetic Algorithms and Engineering Design," John Wiley & Sons, Inc., New York, 1997.
- [3] Pei-Chann Chang, Shih-Hsin Chen, Chen-Hao Liu, "Subpopulation genetic algorithm with mining gene structures for multi-objective flow shop scheduling problems," *Expert Systems with Applications*, 33, 2007, PP. 762-771.
- [4] Maria Joao Alves, Marla Almeida, "MOTGA: A Multi-Objective Tchebycheff based Genetic Algorithm for the multi-dimensional Knapsack problem," *Computers and Operations Research*, 34, 2007, PP. 3458-3470.
- [5] N. Chakraborti, B. Siva Kumar, V. Satish Babu, S. Morita, A. Mukhopadhyay, "A new multi-objective genetic algorithm applied to hot rolling process," *Applied Mathematical Modeling*, 32, 2007, PP. 1781-1789.
- [6] S. Dehuri, S. Patnaik, A. Ghosh, R. Mall, "Application of elitist multi-objective genetic algorithm for classification of rule generation," *Applied Soft Computing*, 8, 2007, PP. 477- 487.
- [7] Chi-Ming Lin, Mitsuo Gen, "Multi-objective resource allocation problem by multistage decision-based hybrid genetic algorithm," *Applied Mathematics and Computation* 187, 2007, PP. 574-583.
- [8] A.R. Rahimi-Vahed, S.M. Mirghorbani, M .Rabbani, "A new particle swarm algorithm for a multi-objective mixed-model assembly line sequencing problem," *Soft Computing* 11, 2007, PP. 997-1012.
- [9] Chang P.C, Wang Y.W, Liu C.H, "New operators for faster convergence and better solution quality in modified genetic algorithm," *Lecture Notes in Computer Science*, 3611, 2005, PP. 983-991.