

Search Optimization using Multiobjective Particle Swarm Optimization

Charu Chhabra¹, Dr. K. K. Paliwal²

¹Department of Computer Science and Engineering, Panipat Institute of Engineering and Technology
Samalkha, Panipat

²Panipat Institute of Engineering and Technology, Samalkha, Panipat

Abstract: *The reusability provides many benefits such as increasing productivity, Reliability & Quality along with reducing the cost & development time and if the number of components developed is not according to the requirement then the technique of reusability is of great help. The main problem faced by the CBSE in reusability is to select the component for reuse as before reusing there is need to retrieve it from repository & before retrieving; searching of relevant component from repository needs to be performed. The proposed technique is a hybridization of Multiobjective Particle Swarm Optimization and Pareto Dominance Principle.*

Keywords: Component Based Software Engineering (CBSE), Reusability, Multiobjective Particle Swarm Optimization, Pareto Dominance Principle, Optimal Solution.

1. Introduction

CBSE is an approach of software development that depends on software reuse [1]. Software reuse is basically the use of some already existing code, design or knowledge in different ways by aiming at improving reliability, Software Quality & productivity. By using the concept of reusability, development effort is saved multiple times as development is done only once and the component is used multiple times [2]. As we know that Key paradigm of CBSE is reusability and the first step towards reusability is to develop the components that can be reused in future, they should be flexible to get adapted in the new environment. Then these components are stored in a repository which is basically a collection of a large number of components that can be reused. This technique is very useful but it also faces some problem. The main problem faced by CBSE is to select the relevant component from repository as to reuse we need to retrieve the component before reusing and for selecting the component from repository there is a need to search the component and for that an efficient search optimization technique should be used that gives an appropriate result.

2. Related Work

As we know that software reuse is the key paradigm of CBSE and software reuse is efficient if we are able to locate the components easily from the repositories [3] and some work has been done in this field that include either component classification, Searching or retrieval. For enhancing the retrieval mechanism Luqi and Guo [4] presented a work in which they discussed an improvement of two different aspects of retrieval methods for software components and they were Profile matching and signature matching. Veras and Silvio [5] used clustering techniques for the organization of software repositories and through their paper, provided a help to refine the searches by grouping together similar components. Niranjana and Rao [6] focused on the implementation of software tool with a new integrated classification scheme to make classification build of software components to facilitate retrieval of software

components depending upon users requirement. Kaur and Goel [7] presented a work in which they discussed how developers and end users can formulate high level and aspect based queries to retrieve components according to their need. Napocca [8] presented a work by using Pareto Dominance Principle for the selection of components from the repositories. Khode and Bhatia [3] worked on improving the retrieval effectiveness using Ant Colony Optimization and showed that there comes good values of Precision and Recall. Dixit and Saxena [9] used Genetic Algorithm for retrieving the component and showed that this approach minimizes the gap between component needed and component available. Elbeltagi et al. presented a work [10] by comparing five optimizing techniques and those were Genetic Algorithm, Memetic Algorithm, Ant Colony Optimization, Particle Swarm Optimization, Shuffled Frog Leaping and by experiments proved that PSO performs better in terms of Success Rate and Solution Quality. So now this work is a Hybrid technique of Multiobjective Particle Swarm Optimization and Pareto dominance principle to achieve multiple objectives in retrieving the components.

3. Multiobjective Particle Swarm Optimization

Particle swarm optimization is an optimization method. It was developed by Doctor Kennedy and Eberhart in 1995. It is a method of swarm intelligence and is based on the research of bird and its social behavior [11].

3.1 Algorithm:

Multiobjective PSO is similar to PSO except in this an archive of non dominated solutions is kept out of each iteration so the steps of this algorithm are as follows [12]:

1. Initialize the swarm & archive
2. For each particle in the swarm:
 - (a) Select leader from the archive
 - (b) Update position
3. Update the archive of non-dominated solutions
4. Repeat

3.2 Pareto Dominance Principle

Pareto dominance Principle was invented by Vilfredo Pareto. This is also known as 20-80 principle which states that 20% of something is responsible for 80% of something else. This implies that 20% of your effort, would complete 80% of a project.

Pareto dominance principle is such a technique that helps to solve the problem graphically. It plots the solutions on the graph and retrieve the good and optimal solutions. The graph that it plots is called as Pareto Chart.

Pareto dominance technique gives the efficient and the optimal solutions by making the pareto chart. The solutions that lie on the pareto frontier are the non dominated solutions.

4. Proposed Technique

A software product includes the number of integrated components via some process on them. The process of retrieving the relevant component from the repository is a very difficult task. Various techniques have been proposed to select the relevant component from the repository on the basis of the client requirement. Selecting the component if we have a single objective to achieve is a simpler task but what to be done if we have to fulfill multiple objectives.

In this research work we have formulated the problem as multiobjective component selection problem where we have two objectives to be fulfilled and those are:

- 1) The number of used components
- 2) The cost of the involved components.

We have to minimize both the objectives i.e. number of components and the cost of the target system. In this we are making a hybrid technique by using the Multiobjective Particle Swarm Optimization and the Pareto dominance principle to deal with the multiobjective optimization problem.

Consider TR the set of final system requirements (end requirements) as

$TR = \{r_1, r_2, \dots, r_n\}$ and

TC the set of components available for selection as

$TC = \{c_1, c_2, \dots, c_n\}$.

Each component c_j can satisfy a subset of the requirements from TR denoted

$TRC_j = \{r_{i1}, r_{i2}, \dots, r_{ik}\}$.

In addition cost(c_j) is the cost of component c_j .

The aim is to search a set of components Sol in such a way that every requirement r_k ($k = 1, n$) from the set TR can be assigned a component c_j from Sol where r_k is in TRC_j , while minimizing c_j of Sol cost(c_i) and having a minimum number of selected components.[8]

4.1 Flow Chart of the Proposed Work

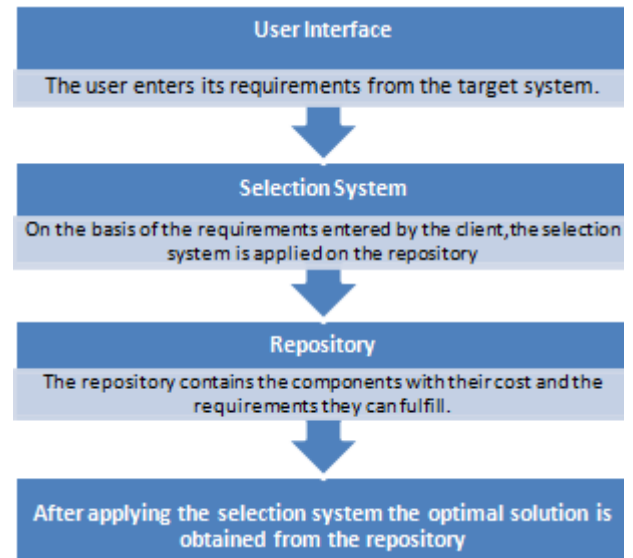


Figure 1: Diagram showing the work flow in this research work

In the proposed work, firstly we have user interface from which user will enter his particular set of requirements that he wish to fulfill with the required system. Then we are having data accessing scenario through which data is accessed from available repository. By using particular selection criteria we get the set of components that will match user's requirements. But what happens? If the set of obtained matched component is too large? How we will come to know that out of these set of components which is best one? Again there will be problem as which is optimal one according to our final system. So we are applying Optimal technique i.e. particle Swarm Optimization which optimizes our results or we can say refine our search. After that we get the best matched component which is having minimum cost plus trying to fulfill most of user's requirement. Here optimization is done in terms of two respective:

Cost associated with the component

Total no of components used to make the system.

5. Working of the Selection System

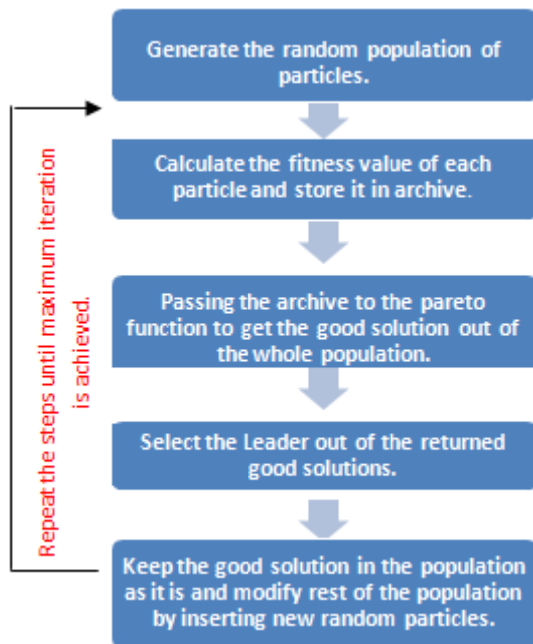


Figure 2: Diagram showing the working of the Selection system

This selection system is composed of Particle Swarm Optimization and Pareto Dominance Principle. According to the Particle Swarm Optimization, we define the population size and the maximum number of iterations. The procedure is started by generating a random population of solutions of size equals to the population size. Then we calculate the fitness value of each and every solution in terms of number of components used and the cost. Then on the basis of the fitnesses the pareto dominance principle returns the good solution to the Particle Swarm Procedure and there it finds the leader out of it. Then the population is modified randomly without replacing the good solutions and the whole process of finding the fitness's and leader is carried out till the maximum number of iterations are achieved. In each iteration, the new leader replaces the previous one if it is better than that and after the last iteration, we get the optimal solutions.

6. Implementation and Results

According to this dissertation aims and objectives, the combination of Multiobjective Particle swarm optimization and pareto dominance principle should result in searching the optimal component solution. So the system is implemented using this Hybrid technique.

Here the component solutions are stored in the repository. The components are associated with the cost and a set of requirements fulfilled by them. According to proposed mechanism, the searching of components using Multiobjective Particle Swarm Optimization and pareto dominance principle with few numbers of components has been taken for the sake of simplicity and implementation is presented here. MATLAB is used to implement the proposed architecture.

Firstly we will present the Repository containing the components. For the implementation few number of components have been taken.

1	2	3	4	5	6	7
1	1	1/2	1	0		
2	2	694	0	1	1	
3	3	169	1	0	0	
4	4	256	0	0	0	
5	5	901	0	1	0	
6	6	80	1	1	0	
7	7	803	1	0	1	
8	8	503	1	1	1	
9	9	190	1	1	0	
10	10	307	1	1	1	
11	11	419	0	1	0	
12	12	91	1	0	1	
13	13	753	1	1	1	
14	14	514	1	1	0	
15	15	906	1	1	1	
16	16	272	0	1	1	
17	17	343	0	0	0	
18	18	243	1	1	0	
19	19	280	1	0	0	
20	20	975	1	1	0	
21	21	604	1	1	1	

Figure 3: Repository: containing components, their cost and requirements fulfilled by them.

Here Fig 3 shows the snapshot of the repository containing components, their cost and the requirements fulfilled by them. we have shown here only few components of the repository ,in actual it contains more than that. Column 1 shows the component number. 2nd column shows the cost associated with the particular component. 3rd column shows whether the component fulfills the requirement1(R1) or not, Value '1' tells that the requirement 1 is fulfilled by that component and the value '0' tells that the requirement is not fulfilled by that component..4th column shows whether the component fulfills the requirement2(R2)or not.5th column shows whether the component fulfills the requirement3(R3)or not .For the sake of simplicity, we have taken maximum three requirements that can be fulfilled by any component and this can be extended according to the usage scenario.

After having a look at the repository, now this is the time to apply our proposed algorithm. As we know that Particle swarm optimization is a population based technique so now the population of the component solution is made according to the users need.

Suppose the user needs all the three requirements then it will give input to the system that it needs all the requirements and the initial population will be made randomly according to all those requirements and the fitness of the whole population is calculated accordingly. Then the pareto dominance principle will select good solutions out of all those solutions on the basis of their fitness's and return back the good solutions . As we cannot get the optimal solution so early from a technique that works on randomness so Particle swarm optimization takes the procedure of iteration. In each iteration it keeps the good solutions of the previous iterations as it is and generate the new random population of solutions and then the fitness of new solutions is calculated and compared with the previous ones and if any of the new solution has fitness better than the previous good solution then it replaces the previous good solution and make the new solution as the good solution.in each iteration we select a leader . This process carries on until we reach the maximum number of iteration. Here the snapshots of this

procedure is shown upto few iterations to make the concept more clear. Firstly the user gives its input to the system through an interface which is shown in the figure 4. Figure 4 shows the input given by the user where he wants a system which should satisfy all the three requirements and it mentions the names of those requirements.

This is upto the user and he can also ask the system that he just want the two requirements to be fulfilled and still the objective will remain same.

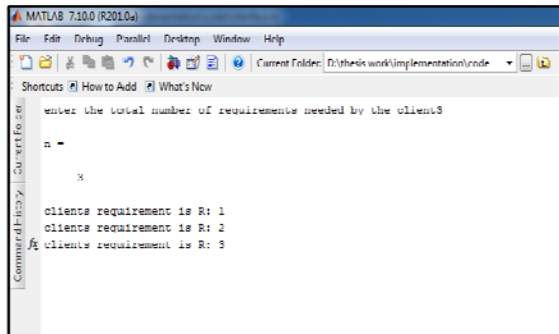


Figure 4: User interface

This figure shows the user interface where user tells that it needs three requirements and the requirements needed by him are R1, R2 and R3. On the basis of the requirement of the user, our technique generates the initial population.

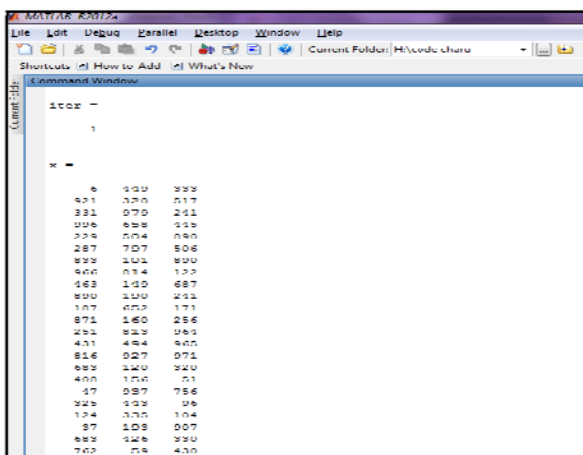


Figure 5: Initial population (population in iteration 1)

Figure 5 shows the generation of initial population means the population in iteration 1. If we see the first solution set (6,449,333) it means that component 6 satisfy R1, component 449 satisfy R2, component 333 satisfy R3. Then on the basis of our proposed technique after generating the initial population, the fitness of all the solutions of population is calculated and stored in archive which is shown in the next figure.

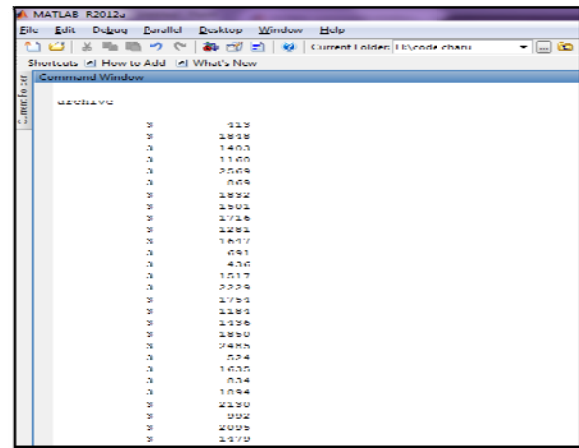


Figure 6: population Fitness generated in iteration 1

The fitness of the whole population in terms of number of components and cost is calculated and stored in the archive according to multiobjective PSO which is shown in the above figure and then Pareto dominance principle solve the problem graphically by plotting the fitness of the solutions on the graph and finding the good solutions out of the whole population.

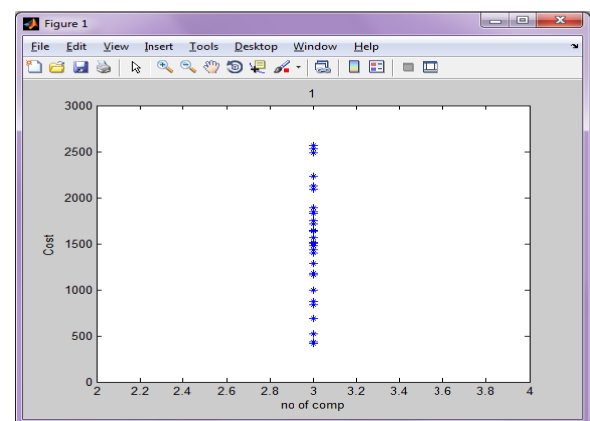


Figure 7: Graph showing the fitness of the population generated in iteration 1

Then the good solutions are taken and shown to the user .Out of these good solutions the leader is selected so that the process can be continued.

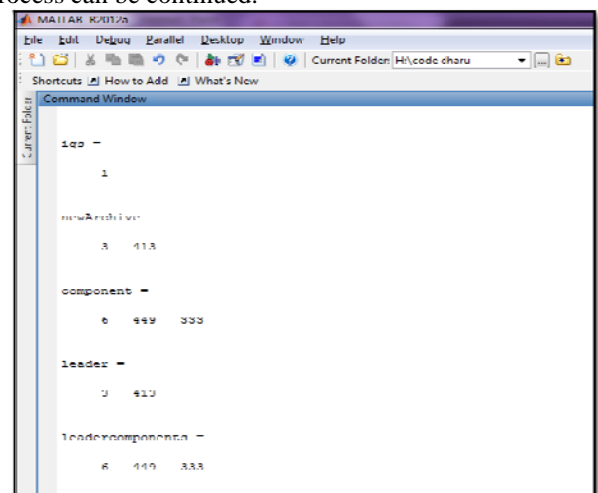


Figure 8: leader selection out of the good solutions returned by the pareto dominance principle.

Now the process is repeated until maximum number of iterations are achieved and every time the population is modified and on the basis of their fitness archive is updated in each iteration and the result obtained after the last iteration is assumed to be the best one. The maximum number of iterations should be equal to the number where previous 10 to 15 iterations are giving the same leader means they are showing that the final optimal solution is achieved and if we give further iterations there will be no change in the good solution. Here in our technique we have taken maximum 100 iterations. If we see the results in the 90th iteration

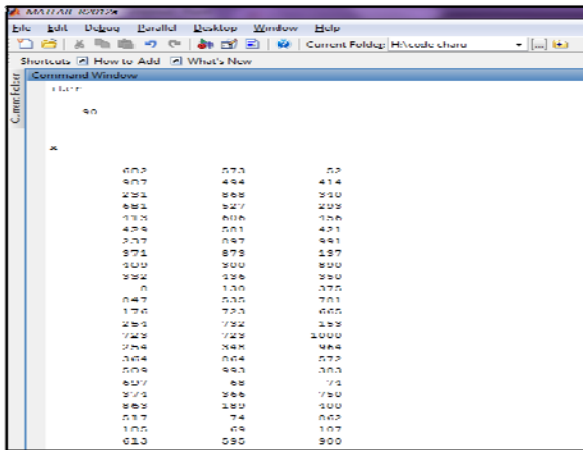


Figure 9: population generated in iteration 90

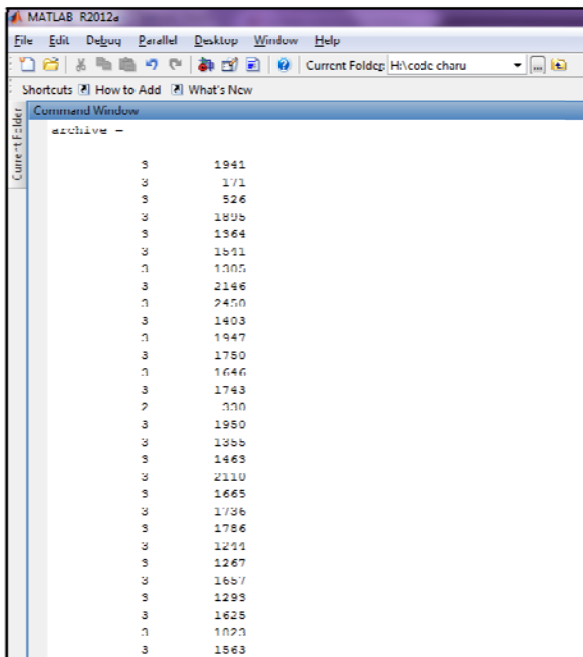


Figure 10: population Fitness generated in iteration 90

Now the fitness of the whole population is calculated in the 90th iteration and pareto dominance principle solves the problem and updates the archive and gives the leader solution which is shown in figure 12.

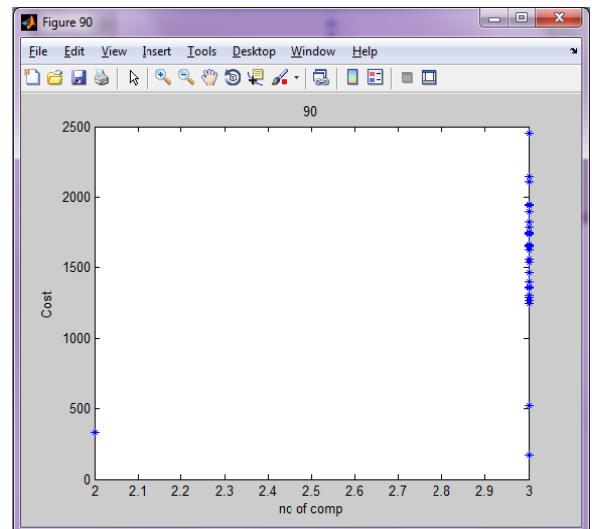


Figure 11: Graph showing the fitness of the population generated in iteration 90

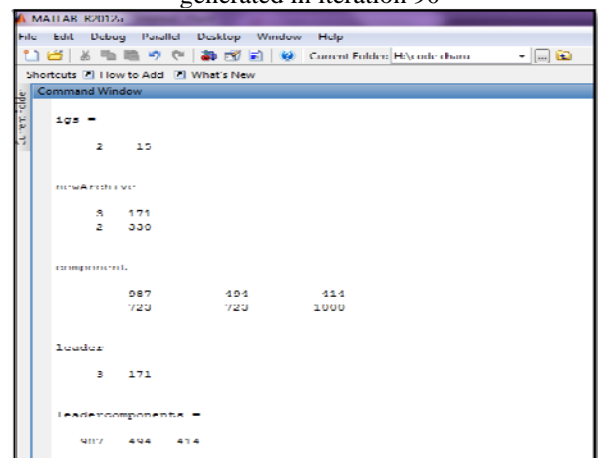


Figure 12: leader selection out of the good solutions returned by the pareto dominance principle

Now as we can see after so many iterations, in the 90th iteration the new archive contains two good solutions those replaced the previous good solutions and the new leader replaced the previous leader. As the maximum number of iterations in our technique is 100 so lets see the result of the 100th iteration

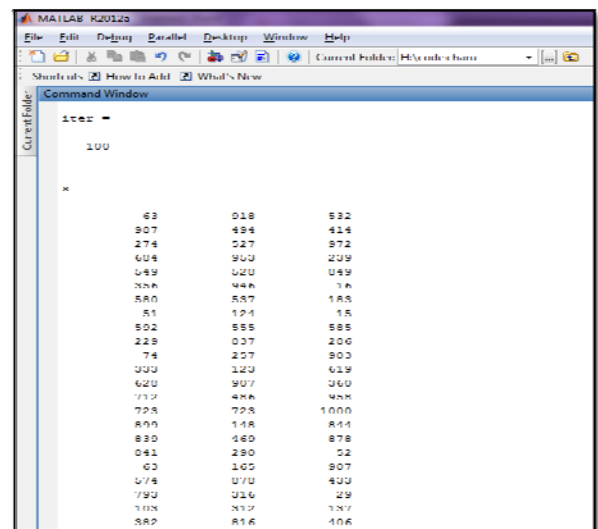


Figure 13: population generated in iteration 100

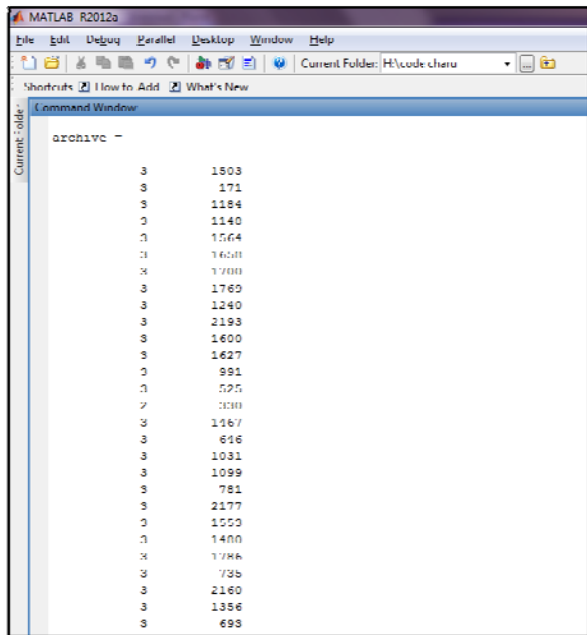


Figure 14: Fitness of the population generated in iteration 100

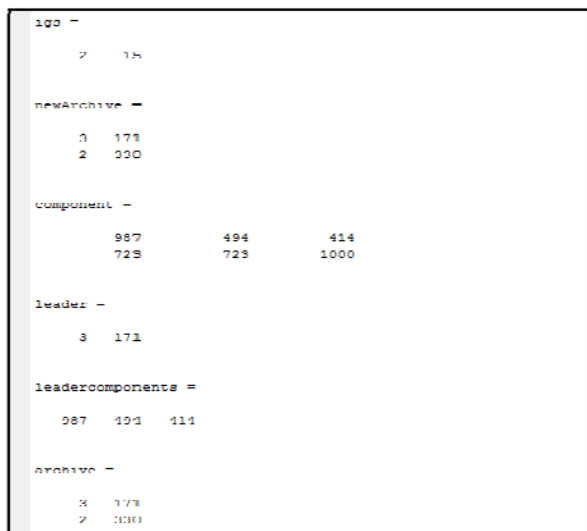


Figure 15: leader selection out of the good solutions returned by the pareto dominance principle.

Now we can see that the archive in the 100th iteration contains the two solutions which were obtained in the 90th iteration also, that means from the last 10 iterations the archive is not changing which implies that the optimal solution is found so we have achieved our objective of finding the optimal solution and the leader is selected out of that. The graph showing the good solutions obtained in the last iteration is shown in figure 15.

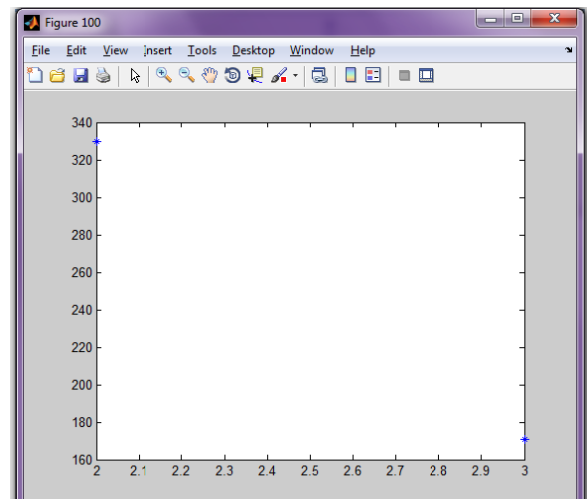


Figure 16: Graph showing the fitness of the population generated in iteration 100

This graph shows the optimal solution as the maximum number of iterations are achieved. The similar results come when the user does not need all the three requirements or we can say if it needs the system satisfying only two requirements that can be (R1,R2), (R1,R3) or (R2,R3)

7. Conclusion and Future Scope

The implementation of proof-of-concept proved successful that the searching mechanism based upon PSO and Pareto Dominance, is capable of returning accurate search results. This technique is efficient and effective but still some future work can be done like there can be more than two objectives or the requirement set can be extended.

References

- [1] Ian Sommerville., "Software Engineering" 7th edition, Addison Wesley, pp. 1-31, 2004.
- [2] Gupta S.C., Kumar A., "A Neural Network based Method to Optimize the Software Component Searching Results in K-Model", International Journal of Computer Applications (0975 – 8887) Volume 72– No.7, pp. 20-27, 2013
- [3] Khode,S.G., and Bhatia,R., "Improving Retrieval Effectiveness using Ant Colony Optimization", International Conference on Advances in Computing, Control and Telecommunication Technologies © IEEE, ISBN: 978-1-4244-5321-4, pp. 737-741, 2009.
- [4] Luqi., and Guo,J., "Toward Automated Retrieval for a Software Component Repository", Proceedings of IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS), pp. 99-105, 1999.
- [5] C. Veras,R., and Silvio,L., " Comparative Study of Clustering Techniques for the Organization of Software Repositories", ISSN : 1082-3409,Vol. 1,pp. 210 -214, 2007.
- [6] Niranjana,P., and Guru Rao,C., "A Mock- Up Tool for Software Component Reuse Repository", International Journal of Software Engineering and Applications (IJSEA),Vol.1,No .2, 2010.

- [7] Kaur,V., and Goel,S., “Facets of Software Component Repository”, International Journal on Computer Science and Engineering (IJCSE), ISSN: 0975-3397, Vol.3, No.6, 2011.
- [8] Napoca , C.,” Pareto dominance - based approach for the Component Selection Problem”,Second UKSIM European Symposium on Computer Modeling and Simulation , IEEE Computer Society, pp.58-63,2008
- [9] Dixit,A., and Saxena,P.C., “ Software Component Retrieval Using Genetic Algorithms” International Conference on Computer and Automation Engineering © IEEE, ISBN: 978-0-7695-3569-2, pp. 151-155, 2009.
- [10]Elbeltagi,E., Hegazy,T. and Grierson, D., “Comparison among five evolutionary - based optimization algorithms”, Elsevier Ltd. / Advanced Engineering Informatics 19, pp. 43-53,2005
- [11]Bai Q,” Analysis of Particle Swarm Optimization Algorithm”, Computer and Information Science,Vol. 3,No. 1 ,pp.180-184, 2010
- [12]Reyes-Sierra M.,Coello C.,” Multi-objective particle swarm optimizers: A survey of the state- of-the-art.” International Journal of Computational Intelligence Research , 2(3): pp. 287-308, 2006.

Author Profile

Charu Chhabra received the B.E. degree in Computer Science and Engineering from Doon Valley Institute of Engineering and Technology, Karnal. At present pursuing the Master of Technology in Computer Science and Engineering Department at Panipat institute of Engineering and Technology, Samalkha , Panipat.

Dr. K.K.Paliwal is B.E, M.E and Ph.D. . He is presently working as a Director in Panipat Institute of Engineering and Technology, Samalkha , Panipat.