Applying Evolutionary Clustering Technique for finding the most Significant Solution from the Large Result Set obtained in Multi-Objective Evolutionary Algorithms

.P.M.Chaudhari
Ph.D. Student,
Post Graduate Department of Computer Science & Engineering,
G. H. Raisoni College of Engineering, Nagpur,India

R.V. Dharaskar
Director,
Matoshri Pratishthan's Group of Institutions (MPGI) Integrated Campus,
Nanded, India

V. M. Thakare
Professor & Head,
Post Graduate Department of Computer Science,Faculty of Engineering &Technology,
S.G.B. Amravati University,
Amravati.India

ABSTRACT
Multicriteria optimization applications can be implemented using Pareto optimization techniques including evolutionary Multicriteria optimization algorithms. Many real world applications involve multiple objective functions and the Pareto front may contain a very large number of points. Choosing a solution from such a large set is potentially intractable for a decision maker. Previous approaches to this problem aimed to find a representative subset of the solution set. Clustering techniques can be used to organize and classify the solutions. A Evolutionary algorithm-based k-means clustering technique is proposed in this paper. The searching capability of Evolutionary algorithms is exploited in order to search for appropriate cluster centres in the feature space such that a similarity metric of the resulting clusters is optimized. The chromosomes, which are represented as strings of real numbers, encode the centres of a fixed number of clusters. Applicability of this methodology for various applications and in a decision support system is also discussed.

Keywords
Multiobjective,Pareto front ,Clustering techniques

1. INTRODUCTION
Multicriteria optimization is applied to a variety of fields and sufficient computational power exists to generate very large non-dominated sets for these problems. In order to be sufficiently representative of the possibilities and tradeoffs, a non-dominated set may be too large for decision makers to reasonably consider; some means of reducing or organizing the non-dominated set is needed [1].
Several researchers have dealt with this issue using cluster analysis or filtering. This paper differs from their work in that it aims to not only make the non-dominated set tractable but to do so without removing any elements of the non-dominated set before presenting the solutions to the decision makers

.

Cluster analysis can be applied to the results of a Multicriteriaoptimization algorithm to organize or partition solutions based on their objective function values. The goal of clustering is to create an "efficient representation that characterizes the population being sampled" [2]. Such a representation allows a decision maker to further understand the decision by making available the attainable limits for each objective, key decisions and their consequences, and the most relevant variables; this presentation is an improvement on a list of potential solutions and their associated objective function values.

2. MULTICRITERIA OPTIMIZATION
Three approaches can be taken to find a solution to Multicriteria problems (Benson and Sayin 1997). The first approach entails reformulating the problem as a single objective problem. To do so additional information is required from the decision makers such as the relative importance or weights of the objectives, goal levels for the objectives, values functions, etc. The second approach requires that the decision makers interact with the optimization procedure typically by specifying preferences between pairs of presented solutions. The third approach, Pareto optimization, finds a representative set of non-dominated solutions approximating the Pareto front. Pareto optimization methods, such as evolutionary Multicriteriaoptimization algorithms, allow decision makers to investigate the potential solutions without a priori judgments regarding the relative importance of objective functions. Post-Pareto analysis is necessary to select a single solution for implementation.

All three approaches to solving Multicriteriaoptimization problems have shortcomings. The solution returned by the single objective approach can be highly dependent on the weights and, in non-convex problems, the responses to changes in weights or goals may be unpredictable. As well, with conflicting and non-commensurate criteria it can be difficult to make value judgments such as choosing weights or goals for the

criteria. Given decision maker input the first approach returns a single solution. Interactive approaches consider only a small set of non-dominated solutions due to the effort required [3]. Pareto optimization approaches return a potentially large number of solutions for consideration. Selecting a single solution from a large non-dominated set is likely to be difficult for any decision maker. It was proposed that an ideal solution procedure for Multicriteriaoptimization is to provide the decision makers with a globally representative subset of the non-dominated set that is sufficiently small so as to be tractable [4]. This work aims to approach this ideal procedure by accepting the computational effort required for generating a large non-dominated set and subsequently organizing it based on its structure. This approach allows decision makers to tractably consider interesting subsets without a priori removal of any solutions from consideration.

Any Pareto optimization method could be employed in this methodology. Evolutionary Multicriteria algorithms apply biologically inspired evolutionary processes as heuristics to generate non-dominated sets of solutions. It should be noted that the solutions returned by evolutionary Multicriteria algorithms may not be Pareto optimal, that is, globally non-dominated, but the algorithms are designed to evolve solutions that approach the Pareto front and spread out to capture the diversity existing on the Pareto front in order to obtain a good approximation of the Pareto front.

3. POST-PARETO ANALYSIS

Post-Pareto analysis aids decision makers in choosing a single solution from the potentially large set of Pareto optimization results. Several researchers have applied clustering methods in different ways to non-dominated sets to aid decision makers. Most of these methods use the similarity of elements in the non-dominated set based on their objective function values and remove elements that are too similar to other elements.

The main goal of multi-objective optimization is to seek Pareto-optimal solutions. Over the years there have been various approaches toward fulfillment of this goal. It has been observed that convergence and diversity are two conflicting criteria which must be balanced in trying to generate the entire efficient front [5]. Clearly, there are two different possible principles for generating a set of solutions representing the entire Pareto-optimal front:

- One-at-a-time strategy, and
- Simultaneous strategy

In the former method, a multi-objective optimizer may be applied one at a time with the goal of finding one single Pareto-optimal solution. Most classical generating multi-objective optimization methods use such an iterative scalarization scheme of standard procedures. The main criticism of most of these approaches is that although there are results for convergence, diversity among obtained Pareto-optimal solutions is hard to maintain in the objective space. Moreover, a careful thought suggests that a systematic variation of weight vectors or "parameters in these scalarization techniques does not guarantee a good diversity in the solution sets [6]. Another important matter is that independent applications of a single-objective optimization algorithm to find different Pareto-optimal solutions one-at-a-time do not make an efficient search and the search effort required to solve the problem to optimality this way needs to be found in every single time the algorithm is applied.

Morse (1980) detailed one of the first applications of cluster analysis to a non-dominated set. The Multicriteria programs considered were linear programs. A solution was removed from the non-dominated set if it was indistinguishable from another solution based on decision maker-defined thresholds. Morse (1980) evaluated seven hierarchical clustering methods. Ward's method, the group average method, and the centroid method performed very well; the other hierarchical clustering methods considered exhibited chaining which reduced the usefulness of the cluster structure. Ward's method was preferred since the clusters at the same level of the hierarchy were of similar size and shape although it performed only slightly better than the centroid and group average methods (Rosenman and Gero 1985).

Rosenman and Gero (1985) applied complete linkage hierarchical clustering to 'reduce the size of the Pareto optimal set whilst retaining its shape'. This method allowed control of the diameter of the resulting clusters. They noted that solutions whose vectors of objective function values are similar may have decision variable vectors that are similar or very different but this idea was not further explored. The objective functions were considered successively in order to avoid the implicit aggregation in applying proximity measures. First, elements of the non-dominated set were clustered using a single criterion. If a solution within a cluster dominated another solution in the cluster on all criteria except the clustering criterion then the dominated solution was eliminated from consideration. The process was repeated for each criterion until the non-dominated set was sufficiently small.

This paper differs from the above since partitional (GA based k-means) clustering is used for combinatorial Multicriteria problems. Either the most interesting cluster, i.e., the 'knee' cluster, was considered in detail by discarding the solutions in other clusters, or one solution from each of the k clusters was considered to form a representative subset of the non-dominated set.

4. CLUSTER ALGORITHM

Cluster analysis, also known as unsupervised learning, is one of the most useful methods in the cluster analysis process for discovering groups. Clustering aims to organize a collection of data items into clusters, such that objects within the same cluster have a high degree of similarity, while objects belonging to different clusters have a high degree of dissimilarity. Cluster analysis makes it possible to look at properties of whole clusters instead of individual objects. This is a simplification that is useful when handling large amounts of data [7].

1. Clustering using Evolutionary algorithms
1.1. Basic principle

The searching capability of GAs has been used in this article for the purpose of appropriately determining a fixed number *K* of cluster centres in R^N; thereby suitably clustering the set of *n* unlabelled points. The clustering metric that has been adopted is the sum of the Euclidean distances of the points from their respective cluster centres. Mathematically, the clustering metric M for the *K* clusters *C*1, *C*2,….., *CK* is given by

$$\mathcal{M}(C_1, C_2, \ldots, C_K) = \sum_{i=1}^{K} \sum_{x_j \in C_i} ||\mathbf{x}_j - \mathbf{z}_i||.$$

The task of the GA is to search for the appropriate cluster

centres z1, z2,….., z*K* such that the clustering metric M is minimized.

1.2. GA-clustering algorithm
The basic steps of GAs, which are also followed in the GA-clustering algorithm, are shown in Fig. 1.

```
Begin
1.  t=0
2.  initialize population P(t)
3.  compute fitness P(t)
4.  t = t+1
5.  if termination criterion achieved go to step 10
6.  select P(t) from P(t-1)
7.  crossover P(t)
8.  mutate P(t)
9.  go to step 3
10.  Output best and stop
End
```

Fig. 1. Basic steps in GAs.

These are now described in detail.

1.2.1. String representation
Each string is a sequence of real numbers representing the *K* cluster centres. For an *N*-dimensional space, the length of a chromosome is *N***K* words, where the first *N* positions (or, genes) represent the *N* dimensions of the first cluster centre, the next *N* positions represent those of the second cluster centre, and so on. As an illustration let us consider the following example.

Example 1. Let *N*"2 and *K*"3, i.e., the space is two-dimensional and the number of clusters being considered is three. Then the chromosome
51.6 72.3 18.3 15.7 29.1 32.2

represents the three cluster centres (51.6, 72.3), (18.3, 15.7) and (29.1, 32.2). Note that each real number in the chromosome is an indivisible gene.

1.2.2. Population initialization
The *K* cluster centres encoded in each chromosome are initialized to *K* randomly chosen points from the data set. This process is repeated for each of the *P* chromosomes in the population, where *P* is the size of the population.

1.2.3. Fitness computation
The fitness computation process consists of two phases. In the first phase, the clusters are formed according to the centres encoded in the chromosome under consideration. This is done by assigning each point x*i*, *i*"1, 2,2, *n*, to one of the clusters *Cj* with centre z*j* such that

$$\|\mathbf{x}_i - \mathbf{z}_j\| < \|\mathbf{x}_i - \mathbf{z}_p\|, p = 1, 2, \ldots, K, \text{ and } p \neq j.$$

All ties are resolved arbitrarily. After the clustering is done, the cluster centres encoded in the chromosome are replaced by the mean points of the respective clusters. In other words, for cluster *Ci*, the new centre z*i* is computed as

$$\mathbf{z}_i^* = \frac{1}{n_i} \sum_{\mathbf{x}_j \in C_i} \mathbf{x}_j, \quad i = 1, 2, \ldots, K.$$

These z*i* s now replace the previous z*i*s in the chromosome. As an illustration, let us consider the following example.

Example 2. The first cluster centre in the chromosome considered in Example 1 is (51.6, 72.3). With (51.6, 72.3) as centre, let the resulting cluster contain two more points, viz., (50.0, 70.0) and (52.0, 74.0) besides itself i.e., (51.6, 72.3). Hence, the newly computed cluster centre becomes ((50.0+52.0+51.6)/3, (70.0+74.0+72.3)/ 3) = (51.2, 72.1). The new cluster centre (51.2, 72.1) now replaces the previous value of (51.6, 72.3).

Subsequently, the clustering metric M is computed as follows:

$$\mathcal{M} = \sum_{i=1}^{K} \mathcal{M}_i,$$

$$\mathcal{M}_i = \sum_{\mathbf{x}_j \in C_i} \|\mathbf{x}_j - \mathbf{z}_i\|.$$

The fitness function is defined as *f*"1/M, so that maximization of the fitness function leads to minimization of M.

1.2.4. Selection
The selection process selects chromosomes from the mating pool directed by the survival of the fittest concept of natural Evolutionary systems. In the proportional selection strategy adopted in this article, a chromosome is assigned a number of copies, which is proportional to its fitness in the population, that go into the mating pool for further Evolutionary operations. Roulette wheel selection is one common technique that implements the proportional selection strategy.

1.2.5. Crossover
Crossover is a probabilistic process that exchanges information between two parent chromosomes for generating two child chromosomes. In this article single point crossover with a fixed crossover probability of k*c* is used. For chromosomes of length *l*, a random integer, called the crossover point, is generated in the range [1, *l* − 1]. The portions of the chromosomes lying to the right of the crossover point are exchanged to produce two offspring.

1.2.6. Mutation
Each chromosome undergoes mutation with a fixed probability µ*m*. For binary representation of chromosomes, a bit position (or gene) is mutated by simply flipping its value. Since we are considering floating point representation in this article, we use the following mutation. A number d in the range [0, 1] is generated with uniform distribution. If the value at a gene position is *v*, after mutation it becomes

$$v \pm 2 * \delta * v, \quad v \neq 0,$$

$$v \pm 2 * \delta, \quad v = 0.$$

The ' + ' or ' — ' sign occurs with equal probability. Note that we could have implemented mutation as

$$v \pm \delta * v.$$

However, one problem with this form is that if the values at a particular position in all the chromosomes of a population become positive (or negative), then we will never be able to generate a new chromosome having a negative (or positive) value at that position. In order to overcome this limitation, we have incorporated a factor of 2 while implementing mutation. Other forms like

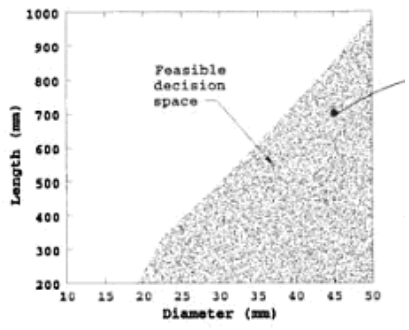
$$v \pm (\delta + \varepsilon) * v,$$

where 0‹Є‹1 would also have satisfied our purpose. One may note in this context that similar sort of mutation operators for real encoding have been used mostly in the realm of evolutionary strategies.

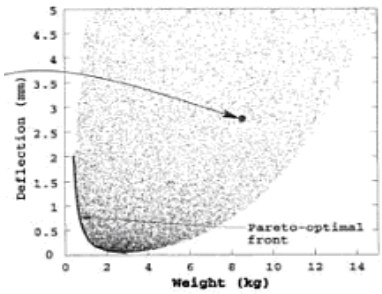
1.2.7. Termination criterion
In this article the processes of fitness computation, selection, crossover, and mutation are executed for a maximum number of iterations. The best string seen up to the last generation provides the solution to the clustering problem. We have implemented elitism at each generation by preserving the best string seen up to that generation in a location outside the population. Thus on termination, this location contains the centres of the final clusters.

5. METHODOLOGY

A cantilever design problem is considered with two decision variables i.e. diameter (d) and length (l). the beam has to carry an end load P. Let us also consider two conflicting objectives of design , i.e. minimization of weight f1 and minimization of end deflection f2. the first objective will resort to an optimum solution having the smaller dimensions of d and l, so that the overall weight of the beam is minimum. Since the dimensions are small , the beam will not be adequately rigid and the end deflection of the beam will be large. On the other hand . if the beam is minimized for end deflection , the dimensions of the beam are expected to be large , thereby making the weight of the beam large .the left plot in Figure 1 marks the feasible decision variable space in the overall search space enclosed by $10 \leq d \leq 50$ mm and $200 \leq l \leq 1000$ mm. it is clear that not all solutions in the rectangular decision space are feasible . Every feasible solution in this space can be mapped to a solution in the feasible objective space shown in the right plot. The correspondence of a point in the left figure with that in the right figure is also shown.



Left Plot



Right Plot

Fig.2 The feasible decision variable space (left) and the feasible objective space (right)

This Fig 2 shows many solutions trading-off differently between the two objectives. Any two solutions can be picked from the feasible objective space and compared. For some pairs of solutions, it can be observed that one solution is better than the other in both objectives as given in Table 1.All solutions lying on this curve are special in the context of multi-objective optimization and are called Pareto-optimal solutions. The curve formed by joining these solutions is known as Pareto-optimal front.

Table 1 Five solutions for the cantilever design problem.

Solution	d (mm)	l (mm)	Weight (kg)	Deflection (mm)
A	18.94	200.00	0.44	2.04
B	21.24	200.00	0.58	1.18
C	34.19	200.00	1.43	0.19
D	50.00	200.00	3.06	0.04
E	33.02	362.49	2.42	1.31

This approach is suitable for decision-makers that do not have *a priori* knowledge of the relative importance of the conflicting objectives in Multicriteriaoptimization problem.

The developed approach is based on the following steps:
1. Obtain the entire Pareto-optimal set or sub-set of solutions by using a multiple-objective evolutionary algorithm (MOEA) or by another means.

2. Apply the GA based clustering algorithm to form clusters on the solutions contained in the Pareto set.
3. To determine the "optimal" number of clusters, *k,* in this set, silhouette plots are used. A value of the silhouette width, $s(i)$, is obtained for several values of *k.* The clustering with the highest average silhouette width is selected as the "optimal" number of clusters in the Pareto-optimal set.
4. For each cluster, select a representative solution. To do this, the solution that is closest to its respective cluster centroid is chosen as a good representative solution.
5. Analyze the results. At this point, the decision-maker can either:
5.1 Analyze the "knee" cluster. The suggestion is to focus on the cluster that has solutions that conform to the "knee" region. The "knee" is formed by those solutions of the Pareto-optimal front where a small improvement in one objective would lead to a large deterioration in at least one other objective. Moreover, from this "knee" cluster the decision maker can select a promising solution for system implementation. This would be the solution closest to the ideal or utopian solution of the multiple objective problem, in a standardized space.
5.2 Analyze the *k* representative solutions and/or select the most promising solutions among this *k* set, selecting the solution closest to the ideal point. By applying the proposed technique, the Pareto-optimal front of a multiple objective problem can be reduced to the "knee cluster" as in 5.1, or to a set of *k* solutions as in 5.2. In both cases the decision maker can choose a good tradeoff for system implementation by selecting the closest solution to the ideal or utopian solution of the multiple objective problems, in a standardized space.

A Matlab code is developed to perform the steps of the proposed technique. From standardized data, the code will run the *clustering* algorithm and from two to a specified number of means it will calculate the average silhouette values and it will return the value of *k* suggesting the most optimal allocation. After this, it will also return the "knee cluster" of the optimal partition, the *k* representative solutions of the Pareto front, and in both cases, the solution closest to the ideal or utopian point.Fig.3, Fig.4 & Fig.5 shows the solution sets.

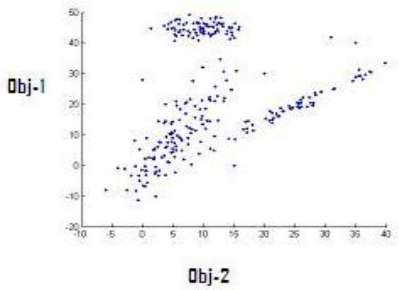


Fig.3 The Solution Set 1

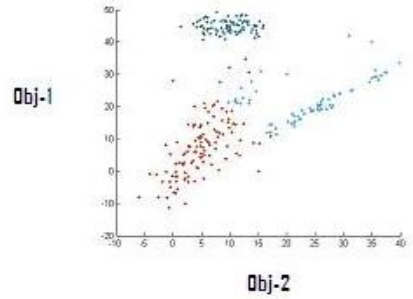


Fig.4 The Solution Set 2

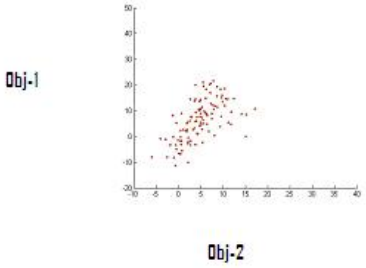


Fig.5 The Solution Set 3

6. CONCLUSION

Pareto optimization methods allow the use of Multicriteriaoptimization models without a priori decision maker preferences. The decision makers can consider the possibilities and trade-offs between objectives before selecting a solution for implementation. These methods suffer from the shortcoming of requiring the decision makers to consider many possible solutions resulting from the optimization procedure. This paper developed and evaluated a cluster analysis methodology to address this issue.

Previous methods involved eliminating some of the Pareto optimal solutions before presenting them to the decision makers. The proposed methodology allows the entire non-dominated set to be presented to the decision makers by providing a tractable structure for the results. This methodology will continue to be applicable as computational power increases and Pareto optimization algorithms improve, leading to the generation of larger non-dominated sets.

This approach is applicable to Multicriteria problems with discrete decision variables. Multicriteria configuration optimization problems and the more general class of combinatorial Multicriteriaoptimization problems have discrete Pareto fronts. It may also be applicable to problems containing highly discontinuous Pareto fronts.

This methodology is particularly useful if similarly performing solutions based on the objective function values may be distinguishable to the decision makers based on the importance of the decision variable values or unmodeled aspects of the problem. Previous approaches to this issue would have eliminated similarly

performing solutions from consideration.

Future work will revisit the issues in cluster analysis including scaling, proximity measures, selection of algorithms, and validation as well as improved visualizations. This work could be extended to consider the proximity of the solutions based on their decision variable values. It may be desirable in some applications to highlight clusters containing similarly performing solutions with very different decision variable values; these solutions could denote unmodeled aspects of the problem or possible freedom in the decision.

7. REFERENCES

[1] Zitzler E, Laumanns M, Thiele L. SPEA2: improving the strength Pareto evolutionary algorithm. Swiss Federal Institute Techonology: Zurich, Switzerland; 2001.

[2] Rosenman, M. A. and J. S. Gero. 1985. Reducing the pareto optimal set in multicriteria optimization (with applications to pareto optimal dynamic programming). Engineering Optimization, 8, 189–206.

[3] Kata Praditwong and Xin Yao. How Well Do Multi-objective Evolutionary Algorithms Scale to Large Problems. 2007 IEEE Congress on Evolutionary Computation (CEC 2007)

[4] M. Laumanns, L. Thiele, E. Zitzler, and K. Deb. Archiving with guaranteed convergence and diversity in multi-objective optimization. In W. B. Langdon, E. Cant´u-Paz, K. Mathias, R. Roy, D. Davis, R. Poli, K. Balakrishnan, V. Honavar, G. Rudolph, J.Wegener, L. Bull, M. A. Potter, A. C. Schultz, J. F. Miller, E. Burke, and N. Jonoska, editors, GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference, pages 439–447, New York, 9-13 July 2002. Morgan Kaufmann Publishers.

[5] A. Mosavi. Multiple Criteria Decision-Making Preprocessing Using Data Mining Tools. IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 2, No 1, March 2010 ISSN (Online): 1694-0784 ISSN (Print): 1694-0814

[6] Lily Rachmawati, and Dipti Srinivasan, Senior Member, IEEE. MulticriteriaEvolutionary Algorithm with Controllable Focus on the Knees of the Pareto Front. IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 13, NO. 4, AUGUST 2009

[7] Helmuth Spaeth. Cluster Analysis Algorithms. John Wiley and Sons, 1980.

[8] Cherhan Foo and Michael Kirley. An analysis of the effects of clustering in graph-based evolutionary Algorithms. 2008 IEEE Congress on Evolutionary Computation (CEC 2008)

[9] Kiri Wagsta , Claire Cardie, Seth Rogers, Stefan Schroedl. Constrained K-means Clustering with Background Knowledge. Proceedings of the Eighteenth International Conference on Machine Learning, 2001, p. 577-584.

[10] Yiu-Ming Cheung. k_-Means: A new generalized k-means clustering algorithm. Pattern Recognition Letters 24 (2003) 2883–2893.

[11] Pradyumn Kumar Shukla and Kalyanmoy Deb. On Finding Multiple Pareto-Optimal Solutions Using Classical and Evolutionary Generating Methods. KanGAL Report Number 2005006

[12] Dilip Datta, Kalyanmoy Deb and Carlos M. Fonseca. Solving Class Timetabling Problem of IIT Kanpur using Multi-Objective Evolutionary Algorithm. KanGAL Report Number 2006006

[13] Eckart Zitzler, Marco Laumanns, and Lothar Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. TIK-Report 103,May 2001

[14] Maiyaporn Phanich, Phathrajarin Pholkul, and Suphakant Phimoltares. Food Recommendation System Using Clustering Analysis for Diabetic Patients. Advanced Virtual and Intelligent Computing (AVIC) Research Center

[15] Jun Zhang, Member, IEEE, Henry Shu-Hung Chung, Senior Member, IEEE, and Wai-Lun Lo, Member, IEEE. Clustering-Based Adaptive Crossover and Mutation Probabilities for Genetic Algorithms. IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 11, NO. 3, JUNE 2007

[16] P.M. Chaudhari, R. V. Dharaskar , V. M. Thakare , “Computing the Most Significant Solution from Pareto Front obtained in Multi-objective Evolutionary Algorithms”, International Journal of Advanced Computer Science and Applications (IJACSA 2010), Vol. 1(4), pp. 63-68