**Proposed Methodology:**

First, NSGA-II algorithm is applied to solve any multi-objective optimization problem. NSGA-II can produce the Pareto front in which all the solutions are optimal and nondominated to each other. Now, the problem is how to extract a single better solution from this front. For this post-Pareto problem, solutions have clustered by k-means. Then, ranks are assigned to the clusters. The decision maker can prefer the clusters according to their ranks. The ranks will be assigned to the clusters according to the number solution. Same rank will be assigned for those clusters which are similar in size. Decision maker can prefer any of them then. The proposed methodology is described below:

**Algorithm Steps:**

Begin {

Step 1: Use NSGA-II for solving any multi-objective optimization problem.

Step 2: Use any clustering method for making the cluster of nondominated solution from the Pareto front. In the proposed work Classical k-means clustering is used

Cluster index = k-means (array of containing NSGA-II results) /cluster index created by k-means algorithm.

**Steps of NSGA-II**

NSGA-II ()

{

Rt = Pt U Qt /combining the parent and offspring population Rt of size 2N.

F = fast nondominated sort (Rt) /F = (F1,F2,….), all are the nondominated fronts of Rt , which are to be sorted according to nondomination. Elitism is ensured for all the population.

Pt+1=NULL and i = 1 / Pt+1 is the new population

until |Pt+1| + |Fi| B N /until the parent population is filled.

Crowding distance assignment (Fi) /calculate crowding distance in Fi that includes ith nondominated front in the parent pop

Pt+1 = Pt+1 + Fi

i = i+1 /check the next front for inclusion

Sort(Fi ,\n) / Sort in descending order using\n (crowded-comparison

operator)

Pt+1 = Pt+1 + Fi [1:(N-| Pt+1|)] /Choose the first (N-| Pt+1|) elements of Fi

Qt+1=make-new-pop(Pt+1) /use selection, crossover, and mutation to create a

new population Qt+1

t = t+1 / increment the generation counter

}

**Steps of k-means**

{

1. Choose the total number of clusters for the final result and set the number as k. Select k patterns in the whole databases as the k centroids of k clusters randomly. All instances are assigned to their closest cluster center according to the Euclidian distance metric.

2. Classify every pattern to the closest cluster centroid. The closest represents the data value that is similar. Other features are also considered.

3. Recompute the cluster centroids, and then, there have k centroids of k clusters as we do after Step 1 of k-means.

4. Repeat the iteration of Step 2 and Step 3 of k-means until the convergence criterion fulfilled. The typical convergence criterion are as follows: no reassignment of any data from one cluster to another or the minimal decrease in squared error.

}

Step 3: Assign the size of each cluster according to the number of their contained solutions.

Size(C1,C2,………….,Cn)

Step 4: Sort the clusters according to their size.

Sort(C1,C2,………….,Cn)

Step 5: Rank the sorted cluster size-wise. The cluster containing the highest number of solution, i.e., highest sized, will be assigned as a rank 1, the next highest among the remaining will be rank 2, and so on.

Rank(C1,C2,………….,Cn)

Step 6: Decision maker will check the most nondominated solution in the firstranked cluster based on user criterion and choose that solution as a final. (In real life, if you will buy something from any shop or market, you will obviously prefer the shop which has more number of options to meet your objective.) If the better solution is not found, then repeat this step for remaining clusters according to their ranks.

end

}

**Implementation and Result Analysis**

The proposed methodology is implemented on MATLAB. There are various types of test problems. Our implementation was done in a machine build with Intel Core i5 processor, 8-GB RAM, and the platform used was Windows 7. We used some of standard test problems such as ZDT and DTLZ problems.

**Test Problems**

The ZDT6 problem can be represented as follows:



This problem has also been tested before by Zitzler et al. This problem is six variable problem (n = 6) having a nonconvex Pareto-optimal set. The density of the solutions across the Pareto-optimal region is non uniform and the density toward the Pareto-optimal front lies within the two objective functions. All variables lie in the region (0, 1), i.e., the Pareto-optimal region corresponds to

0 <=xi\*<=1 and xi\* = 0 for i = 2, 3, …, 6. It is nonconvex in nature, and this causes difficulties for getting true Pareto-optimal front.

For the nonconvex problems, we are not getting the true Pareto-optimal front. Thus, we have taken another test problem ZDT1 which is convex in nature. The problem can be represented as follows:



In this problem, all variables lie in [0, 1] and the Pareto-optimal region corresponds

to 0<=xi\*<=1 and xi = 0 for i = 2, 3, …, 6. We can achieve the continuous Pareto-optimal front in objective space. Only difficulty of this problem is that the number of decision variables is large.

Researches have also been done on more than two objectives for solving multiobjective

optimization. But most of researches were restricted to two-objective problems because

1. Balancing for convergence near the Pareto-optimal front and a good diversity can be tested adequately with two objectives.

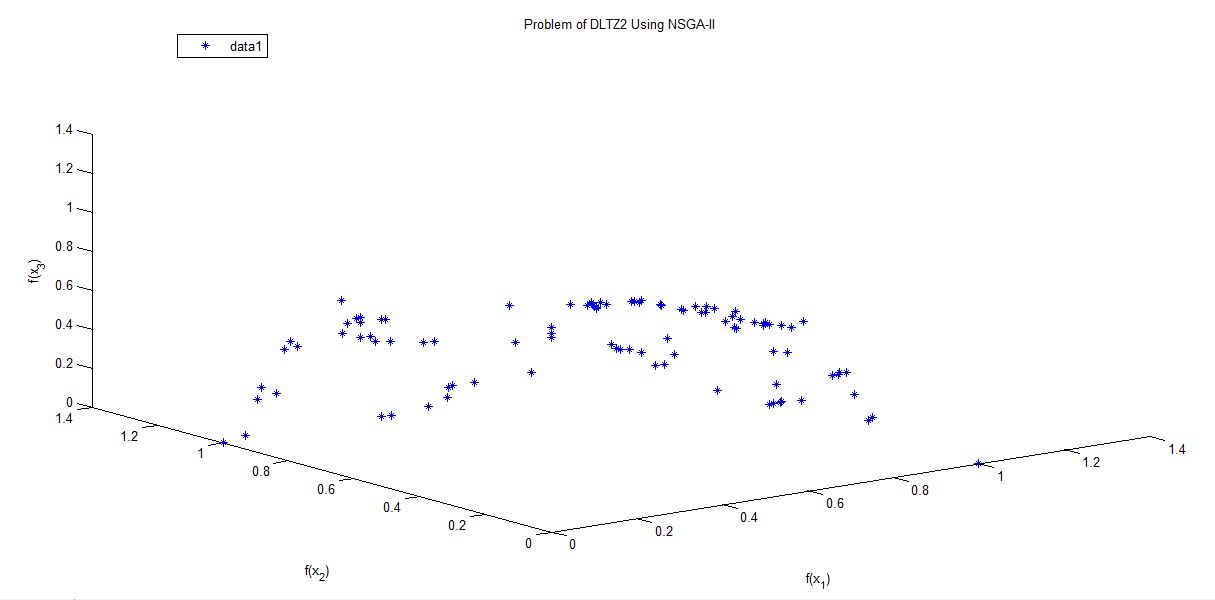
2. Graphical representation of trade-off solutions in multi-objective problem with more than two objectives is difficult to achieve.

Therefore, we have also used a problem consisting of three-objective function named as DTLZ2.

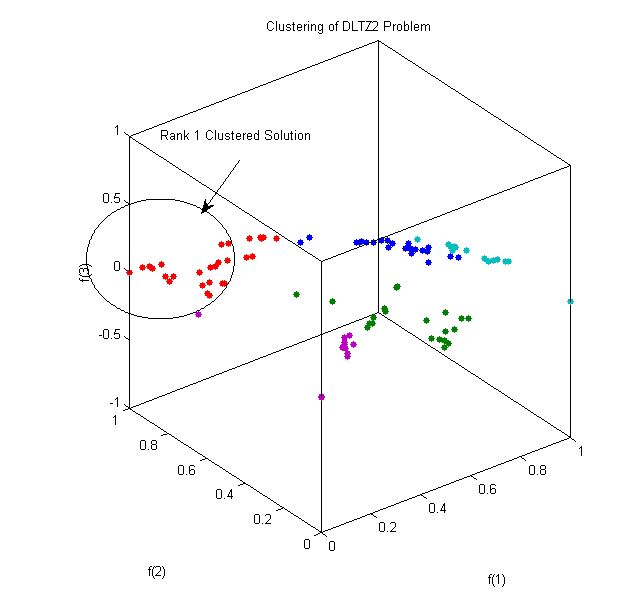


Our improvement was done on a real-coded NSGA-II which uses simulated binary crossover (SBX) operator and polynomial mutation . The crossover probability of Pc = 0.9 and mutation probability of Pm = 1/n where n is the number of decision variables for this algorithm. We use distribution indexes for crossover and mutation operators as gc = 20 and gm = 20. K-means clustering used to the results obtained from the Pareto front. After cluster ranking, we obtain the following results for 2,000 generations. Some screenshots have been given for better understanding of our proposed work (Figs. 1 and 2).

From our resulting clusters in the ‘‘Result and Comparison with Other Clustering Methods’’ section, it is very clear that the good options can be found in the first-ranked cluster. We obtained some good results from the first-ranked cluster on the basis of our objectives or criterion.



**Fig 1: DLTZ2 Using NSGA-II Problem**



**Fig 2: Clustering of DLTZ2 Using K-means**

Table: For Centre Information

|  |  |  |  |
| --- | --- | --- | --- |
| **Cluster Name** | **Centre for F(1)** | **Centre for F(2)** | **Centre for F(3)** |
| **Cluster 1** | 0.763886363636364 | 0.619268181818182 | 0.448000000000000 |
| **Cluster 2** | 0.519838095238095 | 0.217809523809524 | 0.836966666666667 |
| **Cluster 3** | 0.229171428571429 | 0.884817857142857 | 0.378835714285714 |
| **Cluster 4** | 0.942506666666667 | 0.402866666666667 | 0.169773333333333 |
| **Cluster 5** | 0.170042857142857 | 0.181885714285714 | 0.986171428571429 |

**Results and Comparison with Other Clustering Methods**

We have used other clustering techniques such as k-medoids, k-means ++, and fuzzy C-means and compared the ranks with the used k-means results. We have taken a single solution obtained by k-means as a representative for other clustering techniques.

From Table 1, the following interesting observations are made:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Clustering Technique** | **Test Problems** | | | **Rank of Clusters** | | |
|  | **ZDT1** | **ZDT6** | **DLTZ2** | **ZDT1** | **ZDT6** | **DLTZ2** |
| **K-means** | 0.7545 | 0.9410 | 0.1325 | 1 | 1 | 1 |
|  |  |  | 0.9903 |  |  |  |
|  | 0.1320 | 0.1145 | 0.0543 |  |  |  |
| **K-means++** | 0.9133 | 0.9410 | 0.1325 | 1 | 1 | 2 |
|  |  |  | 0.9903 |  |  |  |
|  | 0.0444 | 0.1145 | 0.0543 |  |  |  |
| **FCM** | 0.7545 | 0.9410 | 0.1325 | 1 | 1 | 1 |
|  |  |  | 0.9903 |  |  |  |
|  | 0.1320 | 0.1145 | 0.0543 |  |  |  |
| **K-medoids** | 0.9133 | 0.9410 | 0.1325 | 1 | 1 | 2 |
|  |  |  | 0.9903 |  |  |  |
|  | 0.0444 | 0.1145 | 0.0543 |  |  |  |

In case of second test problem, i.e., ZDT6, same rank of clusters is obtained by every clustering technique which validates the concept of proposed ranking mechanism.

• In case of DTLZ2 and ZDT1, we got different ranks in some clustering techniques. So, we believe this may happened due to the nature of clustering techniques.

• We found that for ZDT1, k-means ++ and k-medoids produced same result which is different from our representative solution. This is happened for the nature of both clustering techniques. In rank 1 cluster, we found that this is a better result from our representative solution which also exists there.