Differential Evolution based Multiobjective Optimization-A Review

Deepa Sreedhar
Department of Computer science
Sree Chitra Thirunal College of Engineering
Thiruvananthapuram

Binu Rajan M .R
Department of Computer science
Sree Chitra Thirunal College of Engineering
Thiruvananthapuram

ABSTRACT
Multiobjective differential evolution(MDE) is a powerful, stochastic multi objective optimization(MOO) algorithm based on Differential Evolution(DE) that aims to optimize a problem that involves multiple objective functions. The MDE has many applications in the real world including supply chain planning and management. This paper presents a review of some multi objective (back propagation) differential evolution algorithms.

General Terms
Algorithms.

Keywords
Differential Evolution, Non-dominated sorting, orthogonal crossover, Fitness sharing, random selection, elitist selection.

1. INTRODUCTION
Multiobjective optimization (Vector optimization) algorithms have great impact in solving many real world problems especially from the field of engineering. It concentrated on optimization problems involving more than one objective functions to be optimized at the same time. Differential Evolution(DE) is a simple, stochastic, yet powerful population based optimization approach for faster optimization. It was introduced by Storn and Price in 1996. After the success of DE algorithm, several variants of it is introduced. It was effective in solving single objective optimization problems. All of the optimization problems might not be single objective type. Some optimization problems involve two or more objective functions, that is multi objective where comes the importance of MOO. Due to the conflicting nature of objectives in MOO problems, when value of one objective increases the other degrades. That results in different optimal solutions. Which forms pareto optimal set. All the solutions in this set are non- dominating each other which means no solution in this set is better than any other solution in the same set in terms of value of the objectives. The former approach for solving multi objective optimization was converting the problem in to single objective by using penalty function method or weighing factor method.

The rest of this paper is organized as follows: Section 2 describes, Differential Evolution, Section 3 presents Multiobjective Optimization, Section 4 presents different types of Differential evolution based multiobjective optimization, Section 5 presents Conclusion, Section 6 describes the references.

2. DIFFERENTIAL EVOLUTION (DE)
DE is a simple, powerful, stochastic, and population based evolutionary algorithm for fast optimization. It was originated by Price and Storn in 1997 [6]. Unlike conventional genetic algorithm, it uses real numbers for representing each of the decision variables present in the chromosome. The approach proceeds by creating an initial population P of random individuals. Then create candidate solution (child) for each parent in the population. If the candidate is better than the parent, replace the parent with new candidate. Otherwise the candidate is discarded. Each parent in the population is selected for candidate creation. For creating the child, select a main parent and three different parents randomly from the population and perform mutation. The mutation is done by adding the decision variable's value of one parent with the weighted difference of values of corresponding decision variable of other two parents. Then apply objective function on main parent and newly created child to determine who will pass to the next generation. Obviously, the member having best cost is transferred to the next generation. This process continues until the maximum size of the population (NP) is reached. The significant parameters which control DE are: NP (size of the population), CR(crossover rate) and F(mutation rate which is the weight applied during mutation process). Several variants of DE have been proposed and successfully applied to many complex and non-linear applications.

3. MULTIOBJECTIVE OPTIMIZATION (MOP)
MOP is an area of Multiple Criteria Decision Analysis that optimizes problems that involves multiple objectives to be optimized simultaneously. Due to the conflicting nature of the objectives, the solution obtained does not satisfy all the objectives simultaneously. So there exist a finite set of solutions termed as pareto optimal (non dominated) solutions. All the solutions in pareto optimal set are non-dominating each other, that is none of the solutions are better in terms of value of the objective function than at least one solution in the set. A back propagation optimization problem can be formulated as

$$\text{Min/Max } f_k(y), \; k=1,2,\ldots\ldots\ldots..M$$

Where M is the number of objectives, y is the decision variable. We need to optimize f(y). That is the goal of MOP is to compute a set of Pareto optimal (non-dominated) solutions to the MOP problem. Let u = (u1; : : : ; u_k), and v = (v1; : : : ; v_k), be two vectors. Then, u dominates v if and only if,

$$u_i \leq v_i, \; i=1,2\ldots M$$

and $$u_i < v_i, \; ; \text{ for at least one i:}$$

This property is termed as pareto dominance and it is used to define pareto optimal points. A solution, x, of MOP problem is said to be pareto optimal if and only if, there doesn't exist another solution y such that f(y) dominates f(x). The pareto optimal set is the set of all non-dominated solutions of an MOP problem. The non-dominated vectors are collectively known as pareto front (see Figure 1). Figure 1 shows a bi-objective minimization problem. Here the set of solutions are represented as circles in which the non-dominated solutions are labeled by a,b,c.

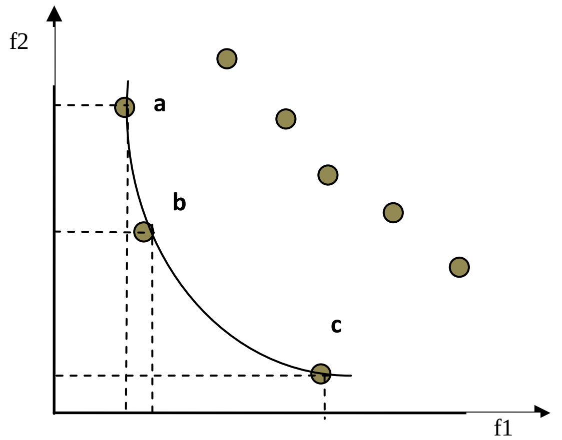

Figure 1: Pareto front

4. DEFFERENTIAL EVOLUTION BASED MOP

The earliest way to solve MOP problems are by using Penalty function method or weighing factor method. The former method is applied to solve biobjective problem [1]. Here consider one objective as constraint and thus converting the multi objective optimization in to single objective. Before the evaluation of the objective function, the latter method multiplies weight to each of the objectives based on the relative importance of the objectives. Then form a new objective function by summing the weighted objectives. Thus it converts multi objectives in to single objective. The two aspects that the researchers have been considered that resulted in the extension of DE to back propagation optimization are:

- How to promote diversity in the population?
- How to perform elitism?

These two aspects are explained as follows:

4.1 Promoting Diversity

Promoting diversity indicate the closeness of the individuals within the population that may be obtained through the selection process by means of mechanisms based on some quality measures. The two most important quality measure for promoting diversity are Crowding distance and fitness sharing.

4.1.1 Crowding distance

The crowding distance factor gives us an idea of how crowded is the closest neighbors of a given individual in the objective function space [14]. This measure estimates the perimeter of the cuboid formed by using the nearest neighbors as the vertices (see figure 2). The filled circles in figure 2 represent solutions of the same dominated front.

4.1.2 Fitness sharing

As a result of resource sharing of an individual with others, the fitness of the individual is degraded in proportion to the number and closeness to individual that surround it within a certain perimeter. A neighborhood of an individual is defined in terms of a parameter called R_p that indicates the radius of the neighborhood. Such neighborhoods are called niches (see Figure 3) [14]. For each individual a niche is defined. The individual whose niche is less crowded are preferred [14].

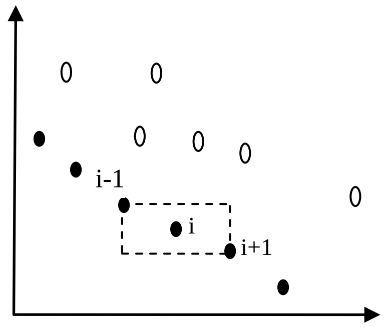

Figure 2:Estimation of crowding distance.

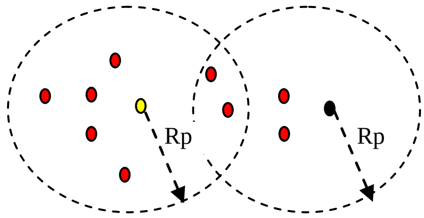

Figure 3: Niches defined for two individuals.

4.2 Performing Elitism

The elitism can be implemented by maintaining best individual of each generation to conform the next population. One of the most important approaches for selecting the best individual of the combined population of parent and child is called non dominated sorting approach.

5. MULTIOBJECTIVE DE ALGORITHMS

This paper deliberates some of the DE based MOP algorithms to solve back propagation problems.

5.1 Multi objective DE for Constrained Optimization (DE-MOC)

The main characteristics of this approach (DE-MOC) are that it makes use of orthogonal design method with quantization to create an initial population [4]. Also, a new constrained handling method is incorporated, which is based on the Constraint first Objective next model [14]. In this model, the constraint leads the objectives because the feasibility of x is more important than minimization of F(x). The method (DE-MOC) helps to locate good points for further exploration in succeeding iterations. It can also use to generate the initial archive and initial evolutionary population (EP). A new crossover operator called orthogonal crossover, is employed to enhance the local search ability and accelerate convergence speed of this approach, which selects the best individual to replace one individual randomly chosen from the population. A hybrid selection mechanism is proposed in this approach, which is a combination of random selection and elitist selection. In elitist selection, the base parent is chosen randomly from the archive and chooses other two parents from the EP. A selection parameter λ is used to regulate the selection pressure.

$$\text{Selection} = \begin{cases} \text{Random selection, if eval} < (\lambda * \text{max\_eval}) \\ \text{Elitist selection, Otherwise.} \end{cases} \quad (1)$$

where, eval is the current number of fitness function evaluations (NFFEs), and max_eval is the maximal NFFEs predefined by the user [4]. If one or more of the variable in the new solution, which are generated by the original DE scheme, are out of bound, i.e $x_i \in [l_i, u_i]$, then the following repair rule is applied.

$$x_i = \begin{cases} li + rndi[0, 1] \times (ui - li) \text{ if } xi < li \\ ui - rndi[0, 1] \times (ui - li) \text{ if } xi > ui \end{cases} \quad (2)$$

Where, $rand_i$ is the uniform random integer in the interval [0,1] in each dimension[4].The main procedure of DE-MOC is as follows:

The algorithm starts by generating an orthogonal initial population by using orthogonal design method. Then produce new offspring by employing original DE scheme. If the offspring and the parent are non-dominated to each other, then the offspring is added to the temporary child population (CP). Append the offspring in to the archive using ϵ-dominance concept [4]. This step repeats until NP number of offspring is created. If the population has enlarged truncate it for the next step of the algorithm. The truncation mechanism performs non dominated sorting on combined set (initial population plus child population) to find best individual among them. The performance of this approach is evaluated on thirteen benchmark problems. The results demonstrate that this approach has a substantial capability of handling various COPs and its solution quality is quite robust and stable.

5.2 Multi objective Differential Evolution (MODE)

The MODE algorithm is a variants of DE, in which best individual is adopted to create new offspring [5]. Also pareto based approach is introduced to implement the selection process. In MODE, randomly generate an initial population. Then a non-dominated sorting of current population members are applied after each generation. The purpose of this process is to remove all dominated points from the population and to give a better direction to the algorithm towards pareto front. Also, it adopts a (μ+λ) selection, pareto ranking and crowding distance in order to produce and maintain well distributed solutions. The working principle of MODE can be shown in Fig:5. In order to confirm the robustness and performance of MODE algorithm, six different benchmark test problems are applied to it. It is found that MODE can handle all type of back propagation problems.

5.3 Pareto based Differential Evolution (PBDE)

In PBDE algorithm, DE is extended to back propagation optimization by incorporating non dominated sorting and ranking selection procedure [2]. Here new candidates are obtained using DE operators which are combined with existing parent population and then choose best individuals from the combined population. The algorithm is not compared with any other approach and is tested on different unconstraint problems performing 250 evaluations. The algorithm is tested on test suite of problems (MOP1-7) which is described in [2]. The result shows that the algorithm performs well in all cases.

5.4 Vector Evaluated DE for MOP (VEDE)

VEDE is a parallel, multi population DE algorithm which is inspired by vector evaluation genetic algorithm (VEGA) [11] approach. A number of M subpopulations are considered which is in ring topology (see Figure:3). Each subpopulation is evaluated using one of the objective functions at hand. Consider there are k objective functions and k<M, then i^{th} population is evaluated with respect to j^{th} objective, where

$$j = \begin{cases} i \bmod k, \text{ if } i \neq rk, r=1,2\ldots.. \\ k, \text{ Otherwise, } i=1,2,\ldots.M \end{cases} \quad (3)$$

Also the best individual obtained from the i^{th} population in every generation is allocated to $(i+1)^{th}$ population of the ring. That is best individual is used by the $(i+1)^{th}$ population to produce mutation vector in the $(G+1)^{th}$ generation. Using migration operator the information is exchanged among the population. Also, a domination selection procedure is incorporated which is described in the equation (4). The equation determines, whether the vector U_i^{G+1} should be a member of the population including the next generation by comparing it with the vector X_i^G.

$$X_i^{G+1} = \begin{cases} U_i^{G+1}, \text{ if } f(U_i^{G+1}) \text{ dominates } f(X_i^G) \\ X_i^G, \quad \text{Otherwise} \end{cases} \quad (4)$$

Here f denotes the objective function under consideration. Four well known bench marking problems are used to evaluate the performance of VEDE [10]. This approach is compared with VEGA approach. The result shows that the VEDE outperforms VEGA in all cases.

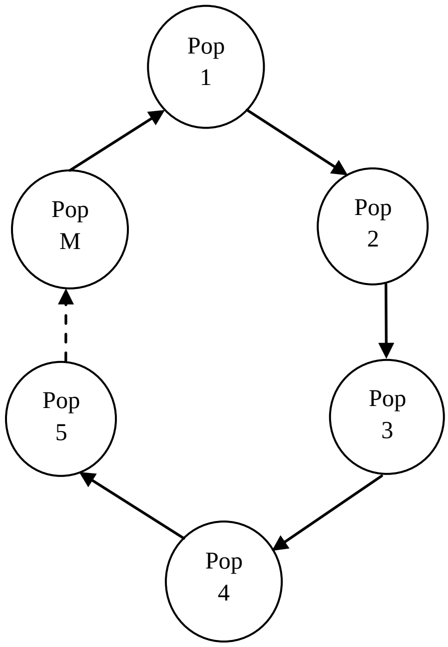

Fig 4: The ring topology.

5.5 Non-Dominated Sorting DE (NSDE)

It is a simple extension of NSGA-II [12]. The main difference between this method and NSGA-II is in the method of generating new offspring. In NSGA II, real coded mutation and crossover operator is applied. But in NSDE, it is replaced by the operators of DE itself. The main procedure of the algorithm are : initially it will set the main parameters i.e the F, CR, NP (number of populations) and max_gen-maximum number of generations of NSDE, then randomly initialize the population points within the bounds of decision variables, after that NSDE creates a new child population of size N from a parent population of size N. The child is entering in to the population only if it dominates the parent. The crossover and mutation operator of the algorithm is same as that of DE. After that combine the two populations (child and parent population) and apply non dominated sorting to it based on the domination level that produces a non-dominated set. The member of this set is not dominated by any other element in the population. After non dominated sorting we will get better individuals which are allocated to next generation. The performance of this approach is evaluated on rotated problems. The result was that NSDE maintains a significantly better convergence, coverage and spread than the NSGA-II. The result also shows that the NSDE consistently converged closely to the Pareto-optimal front, independent of the degree of rotation.

5.6 Differential Evolution for Multi-Objective Optimization (DEMO)

DEMO was proposed in [8]. Unlike original DE algorithm, this algorithm adopts an additional mechanism called truncation which combines the non-dominated sorting and crowding distance metric to keep best N individuals in the population. Also a child replaces the parent only when it dominates the parent. DEMO maintains only one population and it is extended when newly created candidates take part immediately in the creation of subsequent candidates. By selecting newly created candidate for reproduction that highlights the elitism in reproduction. That results in a fast convergence towards the true Pareto front, while the use of non-dominated sorting and crowding distance (derived from the NSGA-II [10]) of the extended population promotes the uniform spread of solutions [8]. DEMO is compared in five high-dimensionality unconstrained problems outperforming in some problems to the NSGA-II, PDEA, PAES, SPEA and MODE. DEMOwSA [7] is a version of DEMO. The difference is that in DEMOwSA a self-adaptation mechanism for adjusting the values of control parameters such as F (mutation control parameter) and CR (crossover control parameter) is incorporated in the algorithm so that the values of the parameters can be adjusted to appropriate values in the evolutionary process. Unlike DE, the value of F is different for trial vector i in each generation which is adjusted based on the value of F for i in previous generation($[F_G]_i$. The new value of F for trial vector i in the generation G+1($F_{i,G+1}$) can be obtained by the following formula:

$$F_{i,G+1} = [F_G]i * e^{TN(0,1)} \quad (5)$$

$$\text{Where } [F_G]_i = \frac{F_{i,G} + F_{r1,G} + F_{r2,G} + F_{r3,G}}{4} \quad (6)$$

Similarly the new value of CR for trial vector i in the generation G+1 can be obtained by the formula:

$$CR_{i,G+1} = (CR_G)_i * e^{TN(0,1)} \quad (7)$$

$$(CR_G)_i = \frac{CR_{i,G} + CR_{r1,G} + CR_{r2,G} + CR_{r3,G}}{4} \quad (8)$$

For details refer [7]. The performance of this algorithm is assessed from the result acquired after running it on 19 test functions using four performance metrics. The result illustrates that the algorithm performed better for all functions except three functions.

5.7 Elitist Multi objective DE (E-MODE)

This approach involves the processing of initial population of size NP [9]. In order to generate a new vector, select three vectors randomly, which are X_a, X_b, X_c. The weighted difference of X_a and X_b and are added to the vector X_c, which is termed as noisy random vector. Crossover is applied to generate a Trial vector(X_c') from noisy and Target vectors. Then the parent population is sorted to get non-dominated solutions. Let the number of non-dominated solutions be Q. The Q number of non-dominated solution obtained after non-dominated sorting is combined with the initial parent population NP to get a combined population (NP+Q). Again perform repeated non-dominated sorting on (NP+Q) number of solutions to classify the solutions as different fronts like front1,front2…..frontN. All of these fronts are't copied to the next generation. Only NP numbers of solutions are to be copied to the next generation. To ensure this, a crowded tournament selection operator is used [9]. This operator helps to win a solution i with another solution j if any one of the following conditions met:

1. If the solution i has better rank.
2. If the solutions have same rank, but solution i has better crowding distance than the solution j (r_i=r_j and di>dj).

While adding fronts to the next generation (NP_{t+1}), the last considered front may exceed the limit on the number of solutions allowable for $NP_{t+1.}$ Then that fronts aren't inserted in to NP_{t+1} and perform crowded distance sorting on that front. The crowding distance represents how the solutions are close to the reference point. The best solution obtained from the sorting is copied to NP_{t+1}. This process continues till the convergence criterion is met.

This algorithm is applied on well-known test problems. The result obtained are compared with the results of MODE algorithm. The result shows that E-MODE can give better spread and diversity in solutions.

5.8 Multi objective based DE(ϵ-MyDE)

The main motivations for this algorithm [3] are two common and effective mechanisms, namely, pareto ranking and crowding distance. The selection criteria and the concept of ϵ dominance allows the good spread of solutions. Also this technique does not allow a difference less than ϵ in the i^{th} objective for two solutions to be non-dominated with regard to each other. Also this algorithm maintains two populations: a main population and secondary population. The main population is the initial population from which parents are selected to generate offspring. The non-dominated solutions obtained are stored in the secondary population. This algorithm represents the chromosome (solution) as a vector of real numbers. Each number denotes the decision variable of the problem. The algorithm starts by generating an initial population.

Then normalize the decision variables around its allowable bound. The remaining steps of this algorithm are same as that of previous MOP algorithms except that it adopts two selection strategies, namely, random selection and elitist selection. The selection strategies are chosen based on the total number of generations 'gmax' and a parameter, sel, which regulates the selection pressure (sel ϵ(.2-1)). For first g(sel% of gmax) generations adopt random selection and for remaining gmax-g generations adopt elitist selection. This algorithm also incorporates a constraint handling mechanism that allows infeasible solution to interfere during recombination so that it helps to solve the constrained optimization problem in a more efficient way. In order to evaluate the performance of ϵ-MyDE, it is compared to other algorithms namely NSGA-II and PDA, also some test functions and specialized metrics are adopted. The result shows that ϵ-MyDE was the best overall performer.

5.9 Pareto Differential Evolution(PDE)

PDE is also a back propagation optimization approach[2]. The main points of this algorithm are: the algorithm generate an initial population according to Gaussian distribution N (.5,.15), then all dominated solutions are removed from the population, carry out crossover only with non-dominated solutions at each generation, if the number of non-dominated solutions exceeds the limit, then find out distance metric relation D(x) between non-dominated solutions in order to remove one which is closer to any of the non-dominated solution in the set and for producing new child, randomly select three parents from the population. The newly generated child replaces the main parent in the population only if it dominates the main parent. The algorithm was tested on two bench mark problems which contain two objective function and thirty variables. The solutions of the two test problem, provided by PDE algorithm, are compared with 12 other multi objective evolutionary algorithms (MEAs). Out of 12 algorithms no algorithm produces optimal result. PDE is significantly better than some of the MEAs. But there is no single crossover rate for which PDE is superior than all other algorithms.

6. CONCLUSION

We have presented the review of nine back propagation differential evolution algorithms that are used to solve multi objective optimization problems. Each of these algorithms is compared with multi objective evolutionary algorithms (MEAs) and the result was that in most of the cases MDE algorithm was superior.

7. REFERENCES

[1] B.V Babu, and Mathew, L. J. 2003. “Differential Evolution for Multi-Objective Optimization”. (Dec 2003), 2696–2703.

[2] Nateri, K. M. 2002. “Multiobjective Optimization Using a Pareto Differential Evolution Approach”. (May 2002), 1145–1150.

[3] Luis, V. S. and Carlos, A. C. C. 2005. “An Algorithm Based on Differential Evolution for Multi-Objective Problems”. 151–169.

[4] Wenyin, G. and Zhihua C. A. 2008. “Multiobjective Differential Evolution Algorithm for Constrained Optimization”. 182-188.

[5] Ashish, M. G. B V Babu. Improved Strategies of Multi-objective Differential Evolution (MODE) for Multi-objective Optimization.

[6] R. Storn and K. Price. 1995 “Differential evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces”.

[7] Ales Z, Janez B, Borko B, and Viljem Z. 2007. “Differential Evolution for Multiobjective Optimization with Self Adaptation”. 3617-3624.

[8] Tea R and Bogdan F. 2005 “DEMO: Differential Evolution for Multiobjective Optimization.pages”. 520-523.

[9] B. V. Babu, Ashish M. G. 2007. “Elitist- Multi-objective Differential Evolution(MODE) algorithm for Multi-objective Optimization. 441-456.

[10] K.E. Parsopoulos, D.K. Taoulis, N.G. Pavlidis, V.P. Plagianakos, and M.N. Vrahatis. 2004. “Vector Evaluated Differential Evolution for Multiobjective Optimization”. (June 2004), 204–211.

[11] J. D. Schaffer. 1984.“Multiple Objective Optimization With Vector Evaluated Genetic Algorithms”.

[12] Antony W. L. and Xiaodong L. 2004. “Solving rotated multi-objective optimization problems using differential evolution”. 861–872.

[13] Kalyanmoy, D. Amrit, P. Sameer, A. and T. Meyarivan. “A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II”. (April 2002) , 182-197.

[14] Efren M.M, Margarita R.S.,Carlos A. C. C. “Multi-Objective Optimization using Differential Evolution: A Survey of the State-of-the-Art”.

.

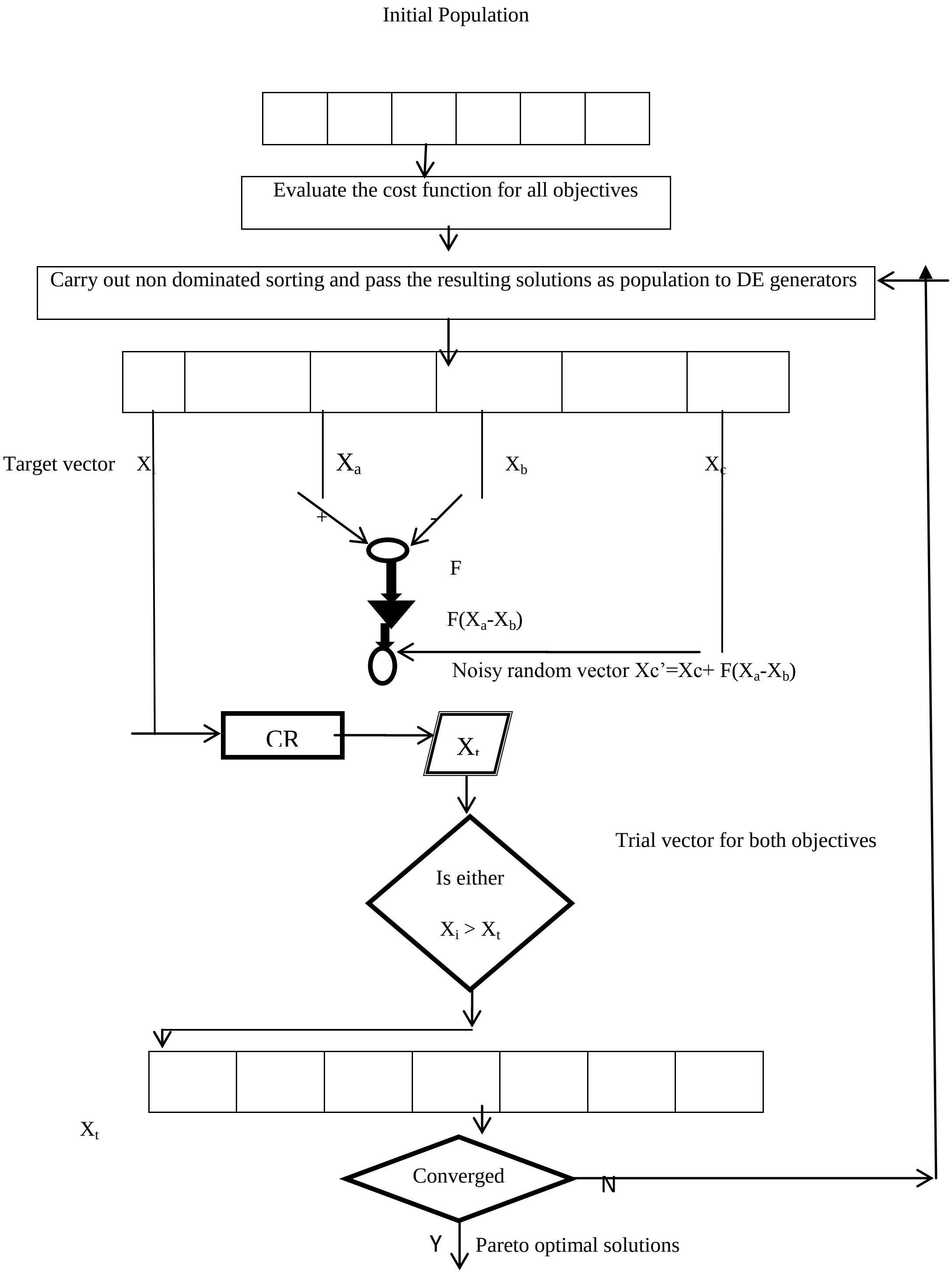

Fig 5: Working rule of MODE algorithm