

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/261202555>

A new archive based steady state genetic algorithm

Conference Paper · June 2012

DOI: 10.1109/CEC.2012.6256448

CITATIONS

10

READS

178

2 authors:



Kaustuv Nag

Indian Institute of Information Technology Guwahati

18 PUBLICATIONS 248 CITATIONS

SEE PROFILE



Tandra Pal

National Institute of Technology, Durgapur

64 PUBLICATIONS 1,070 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Can we have a simple autoencoder that extracts features like a complex convolutional neural network? [View project](#)



Finding Robust Solutions with Consensus in Multi-Objective Optimization [View project](#)

A new Archive based Steady State Genetic Algorithm

Kaustuv Nag

Department of CSE
National Institute of Technology, Durgapur
Durgapur, India
kaustuv.nag@gmail.com

Tandra Pal

Department of CSE
National Institute of Technology, Durgapur
Durgapur, India
tandra.pal@gmail.com

Abstract— In this paper we have proposed a new archive based steady state multi-objective genetic algorithm, which performs well, especially in higher dimensional space. An improved archive maintenance strategy has been introduced in this algorithm which is adaptive as well as dynamic in size. The archive maintenance strategy tries to maintain only the set of nondominated solutions in the archive. However, it maintains a minimum size of population when the nondominated solutions are not sufficient to fill the population. In this algorithm we have proposed a new environmental selection and a new mating selection. The mating selection reduces the exploration in less probable search region enhancing the exploitation of existing solutions. A new crossover operator DE-3 has also been proposed in this article. The proposed algorithm has been compared with three other existing multi-objective optimization algorithms NSGA-II, SPEA2 and AbYSS. Our algorithm outperforms the other three algorithms for its better diversity and convergence to true Pareto optimal front.

Keywords- Constraint handling, genetic algorithms, multi-objective optimization, Pareto optimality.

I. INTRODUCTION

IN the last few decades many multi-objective evolutionary algorithms (MOEA) have been proposed. MOEAs are used to solve multi-objective optimization problems for its advantages over classical algorithms. Higher dimensional objective space and variable space, constraints, non-convex and discontinuous search space, multimodal function profiles, etc. are the main factors causing difficulties in multi-objective optimization problems. All these problems can be reduced by proper tuning of exploration and exploitation in MOEA.

There are many multi-objective algorithms, but most of them perform well for bi-objective and comparatively less complex test problems. Many popular algorithms like NSGA-II [1], SPEA2 [3] and AbYSS [20] could not solve complex three dimensional test problems like DTLZ3, DTLZ6. Here, we have proposed a new adaptive multi-objective evolutionary algorithm which outperforms the other three existing algorithms for complicated test problems. We have also proposed a new environmental selection scheme for our algorithm which reduces the chance to explore the less desirable objective space. A new mating selection has also been used in the proposed algorithm which increases the probability to exploit more desirable solutions. Based on this

mating selection, a new crossover mechanism, DE-3 has been proposed. This crossover is suitable for steady state algorithms, allowing a large number of genotype variables. The archive truncation method in the proposed algorithm allows it to prune solutions efficiently maintaining the diversity among the solutions of the archive.

Without loss of generality we can restate a constrained multi-objective optimization problem to a constrained multi-objective minimization problem. The proposed algorithm solves the following constrained minimization problem:

Minimize $(f_1(X), f_2(X), \dots, f_M(X))$

Subject to

$$g_j(X) \leq 0, j = 1, 2, \dots, J \quad (1)$$

$$h_k(X) = 0, k = 1, 2, \dots, K$$

$$x_i^{(L)} < x_i < x_i^{(U)}, i = 1, 2, \dots, N.$$

In this article we have used the following definition of *constrained Pareto domination* for constrained problems: a solution p *constrained-dominates* another solution q for an M objective minimization problem if the following conditions are satisfied:

- 1) Solution p and q are both feasible and solution p *dominates* solution q .
- 2) Solution p is feasible and solution q is not.
- 3) Solution p and q are both infeasible, but solution p has a smaller overall constraint violation.

The remainder of this article is organized in the following way. In Section II, we have discussed some related works on multi-objective evolutionary algorithms. The proposed algorithm is presented in Section III. Experimental results and comparisons are in Section IV. At last we conclude in Section V.

II. MULTI-OBJECTIVE EVOLUTIONARY ALGORITHMS

There exist many multi-objective evolutionary algorithms. Some of them are NSGA-II [1], SPEA [7], SPEA2 [3], SPEA2+ [11], PESA [10], PESA-II [19], RDGA [12], DMOEA [13], ϵ -MOEA [14], Omni-Optimizer [8], GDE3 [32], FastPGA [9], AMGA [4], AMGA2 [5], etc. We discuss below various key concepts of some popular algorithms. A more comprehensive survey and references can be found in [15], [16], [18], and [17].

Evolutionary algorithms are used to solve NP-hard

problems. So, researchers try to develop algorithms giving better results as well as easy to implement and computationally less expensive. The researchers face the problem to make a balance between computational cost, performance and complexity (regarding implementation) of the algorithm. Some algorithms (like NSGA-II [1]) assign some measures (like crowding distance) before the archive truncation method. Some other algorithms (like SPEA2 [3], GDE3 [32]) assign the measures before archive truncation method and reassign it after deleting each solution. The second approach is better than the first approach. But the second one requires more computation. Though the complexity of some algorithms are less they are difficult to implement. Algorithms like RDGA [12] (computationally complex; complexity is $O(MN^3)$) and DMOEA [13] (computationally efficient; complexity $O(MN)$) are criticized for their difficulty to implement [18].

Some algorithms work very well for bi-objective problems. But they fail to perform well in higher dimensional objective space. For example NSGA-II relies on crowding distance to measure diversity. Crowding distance has severe drawback regarding choosing its neighbors in higher dimensional objective space [2]. There are several other modifications of crowding distance [4], [5], [8]. But all these modifications have the same drawback regarding choosing the neighbors. Till today there are very few algorithms which address higher dimensional objective space. In this paper we have proposed a new algorithm which performs better than other existing state-of-the-art algorithms in higher dimensional objective space.

Algorithms like NSGA-II, SPEA, SPEA2, GDE3, and Omni-Optimizer are generational. If archive size is N , then in each generation they reproduce N number of off-springs. On the other hand algorithms like AMGA, AMGA2 are in between steady state and generational algorithms. FastPGA uses an adaptive strategy which changes the archive size dynamically. FastPGA also dynamically determines the number of offspring solutions to be produced in each generation. All other algorithms mentioned above produce equal number of off-springs in each generation. Thus, for these algorithms, if the maximum number of evaluations is fixed and we apply them with large population size, number of generations will decrease, increasing early exploration and reducing exploitation.

NSGA-II [1], SPEA2 [3], AMGA2 [5], etc. use archive truncation strategy which does not allow boundary solutions to be truncated. PESA [10], SPEA [7], etc. use poor archive truncation strategy which may remove boundary solutions. Again, algorithms like SPEA2+ [11], Omni-Optimizer [8], and AMGA [4] preserve diversity both in objective space (phenotypic space) and in variable space (genotypic space). Whereas, NSGA-II, SPEA2, FastPGA [9], AMGA2 etc. preserve diversity only in phenotypic space. Algorithms like ϵ -MOEA [14], Omni-Optimizer [8] etc. use ϵ -domination while preserving diversity.

Different algorithms also use different selection

mechanisms. NSGA-II uses very popular fast nondominated sorting mechanism. SPEA uses popular strength Pareto approach. This approach is extended in SPEA2 and FastPGA. In most of the cases, domination level is emphasized over the diversity. But it may not be a good choice always. It shows poor performance for multi-modal problems.

Different variation operators are also used in various algorithms. Uni-modal normal distribution crossover [21], simulated binary crossover [22] and parent centric crossover [23] are often used for real coded chromosome. The main problem associated with them is lack of self-adaptability [5]. Sometimes during simulation of algorithms various variation operators are switched. DE crossover [24] shows nice self-adaptability. The performance of DE crossover is greatly dependent on the choice of the parents. A modified version of DE crossover operator is introduced in [5]. It is called DE-2. Most algorithms randomly generate initial population. Omni-Optimizer, AMGA and AMGA2 uses Latin-hypercube [25] based uniform sampling (LH sampling) to generate the initial population.

The above discussion, in short, explains various existing genetic approaches and their differences. Our algorithm is enriched by borrowing the best properties from them. Computational complexity of our algorithm is $O(MN^2)$, which is neither too expensive as SPEA2 or RDGA nor too difficult to implement as DMOEA. The performance of the proposed algorithm is better in higher dimensional space compared to other algorithms. The proposed algorithm is completely steady state genetic algorithm. It uses new archive truncation technique which always preserves boundary solutions. It preserves diversity in phenotypic space. In our algorithm domination level is not always emphasized over diversity. When solutions of first nondominated fronts are not sufficient enough to fill the archive, during selection procedure we emphasize on diversity rather than domination level. Our algorithm uses DE-3 as genetic variation operator for its excellent self-adaptive behavior. This algorithm allows the user to choose the appropriate population initialization process.

III. PROPOSED ALGORITHM

Our proposed algorithm maintains an archive. The main difference of the proposed algorithm from the existing archive based algorithms is that the archive is used to store only nondominated set of solutions and archive size cannot fall below a minimum allowable archive size and cannot go beyond a maximum allowable archive size. Let, N_{min} and N_{max} are respectively the minimum and maximum allowable archive sizes. N_{min} is at least four, as the proposed algorithm uses DE-3 crossover which requires four solutions. We present the proposed algorithm in Fig. 1.

A. Population Initialization

The proposed algorithm provides user the privilege for choosing appropriate population initialization mechanism. The user can use prior knowledge integration mechanism during population initialization phase, if available. Even in absence of prior knowledge, LH sampling [25] can also be used in the

1. **proposedAlgorithm()** {
2. Initialize Population of size N_{max}
3. Evaluate the initial solutions
4. Initialize the archive using initial population
5. **while** (termination condition is not satisfied) {
6. Perform mating selection to select one primary parent and three auxiliary parents
7. Perform DE-3 crossover to create an offspring solution
8. Perform polynomial mutation on the offspring
9. Evaluate the mutated offspring
10. Update the archive by using the offspring // Environmental selection
11. } // end of while
12. **return** desired number of solutions from the archive
13. }

Figure 1. Pseudo code of the proposed algorithm.

initialization process. However, in this study we have taken N_{max} number of initial solutions randomly. This has also helped us to compare the proposed algorithm with other existing algorithms as they have also used random initialization. It also shows unbiased performance result.

B. Initializing Archive Using Initial Population

A fast nondominated sorting [1] is performed on the initial population. This sorting is stopped after getting N_{min} number of solutions. It produces a set of sub-fronts $\{F_1, F_2, \dots, F_i\}$ after sorting such that, $(|F_1| + |F_2| + \dots + |F_i|) \geq N_{min}$ and $(|F_1| + |F_2| + \dots + |F_{i-1}|) < N_{min}$ holds. If i is greater than one, then a flag is turned ON indicating that the archive is not a set of nondominated solutions and in future fast nondominated sorting must be used on this archive. We call this flag as *fastNonDominatedSortRequired*. This procedure ensures that archive size does not fall below N_{min} . We always try to maintain a set of nondominated solutions in the archive.

C. Mating Selection

We need to select one primary parent and three auxiliary parents for the proposed DE-3 crossover. To select the primary parent, at first the objective values are normalized. Then, for each solution, distances (in the normalized objective space) from first and second nearest neighbors are calculated. The solution with highest first nearest distance is selected as the primary parent. If a tie occurs, the solution (among the tie making solutions) with the highest second nearest distance is chosen. If again a tie occurs, we select one of them randomly. After selection of primary parent the auxiliary parents are chosen randomly from the archive. Only restriction is that all these four parents must be mutually exclusive to each other.

D. DE-3 Crossover

The details of DE crossover can be found in [32]. The expression used for crossover in the proposed algorithm is given in equation (2). Let, p_i be the primary parent and a_1, a_2, a_3 be the auxiliary parents; N be the number of variables and j_r be a random number uniformly distributed between 1 and N . r_j be a uniformly distributed random number in $[0, 1]$. This DE crossover operator uses two tuning parameters F and CR . Let, o be the offspring and o_j be the j^{th} variable of o , then

$$o_j = \begin{cases} (a_3)_j + F((a_1)_j - (a_2)_j), & \text{if } r_j < CR \text{ or } j = j_r \\ (p_i)_j, & \text{otherwise} \end{cases} \quad (2)$$

DE crossover [32] does not differentiate between selection strategies of primary and auxiliary parents. The primary as well as the auxiliary parents are selected randomly. In [5] a modified version of this DE-crossover, DE-2 is available. The DE-2 crossover [5] has used random selection for auxiliary parents, while primary parent is selected based on the domination level (measured by rank) and diversity (measured by crowding distance) before selecting auxiliary parents. In the proposed DE-3 crossover we select auxiliary parents randomly, while primary parent is selected based on diversity only in normalized objective space, as discussed in Section IIIC.

E. Polynomial Mutation

We have used polynomial mutation proposed by Deb et al [8]. It is defined as follows. Let l_j and u_j are respectively the lower and upper bound of j^{th} variable. If x_j and x'_j are the respective values of j^{th} variable before and after the mutation and r_j is a uniformly distributed random number in $[0, 1]$, then x'_j is as follows.

$$x'_j = x_j + \delta_q(u_j - l_j) \quad (3)$$

where

$$\delta_1 = \frac{x_j - l_j}{u_j - l_j}, \quad \delta_2 = \frac{u_j - x_j}{u_j - l_j} \quad (4)$$

$$\delta_q = \begin{cases} [2r_j + (1 - 2r_j)(1 - \delta_1)^{\eta_m + 1}]^{1/(\eta_m + 1)} - 1, & \text{if } r_j \leq 0.5 \\ 1 - [2(1 - r_j) + 2(r_j - 0.5)(1 - \delta_2)^{\eta_m + 1}]^{1/(\eta_m + 1)}, & \text{otherwise} \end{cases} \quad (5)$$

Here η_m is the distribution index for mutation.

F. Updating Archive Using Offspring

Updating the archive using the offspring is the most crucial part of any multi-objective evolutionary algorithm. In this study, we have introduced a new environmental selection as archive maintenance strategy. We always maintain a set of nondominated solutions in the archive. But, we need to ensure that the archive size should not fall below N_{min} . The allowable maximum archive size is N_{max} . When we add a solution to the archive, two cases may occur: either

fastNonDominatedSortRequired flag is ON (set to true) or OFF (set to false). We discuss both the cases below.

If the flag is ON, we simply add the offspring solution to the archive and then the new archive is refined. For this purpose we use the same strategy as used in archive initialization (using initial solutions) phase. After refining the archive the *fastNonDominatedSortRequired* flag is turned ON or OFF accordingly. But, a special case may occur, when the archive is exactly $N_{max} + 1$, after adding the offspring. It may happen that if we add (to the archive) all the solutions up to $(i - 1)^{th}$ sub-fronts, archive size falls below N_{min} and then if we add all the solutions of the i^{th} sub-front, the archive size becomes $N_{max} + 1$. To solve this problem, a solution from the i^{th} sub-front is pruned and the remaining solutions are added to the archive. The archive size becomes N_{max} and the *fastNonDominatedSortRequired* flag is turned ON. To prune one solution from i^{th} sub-front we use two different methods: one for bi-objective problems and the other for problems for more than two objectives. If the problem is bi-objective then crowding distance [1] is assigned to the solutions of i^{th} sub-front. Then the solution having minimum crowding distance is deleted. We have used the definition of K. Deb et al. [1] for crowding distance. For problems having more than two objectives, we propose a new archive pruning method as follows. For all solutions of i^{th} sub-front, distances from first and second nearest neighbors are found. In distance calculation all solutions of all the fronts from 1st to $(i - 1)^{th}$ are also considered as the neighbors. Then the solution with minimum first nearest distance is pruned. If a tie occurs, we pick the solution having minimum second nearest distance. If again a tie occurs, we pick one of them randomly. We call this procedure *pruneOneSolution*, which is presented in Fig. 2.

If the flag is OFF, we compare the offspring with all solutions of the archive to add it to the archive. If any of the solutions dominates the offspring, then the offspring is simply discarded and the archive remains unchanged. Otherwise, it will dominate a set of solutions (may be null set). Let the dominated set of solutions is S and the archive is A . Then, if $(|A| - |S|) > N_{min}$, we add the offspring to A and delete solutions of S from A . If $(|A| - |S| + 1) < N_{min}$, we use *pruneOneSolution* procedure to prune one solution from S and add the remaining solutions to A . It ensures that the archive size will not exceed N_{max} . After this procedure a special case may arise. If (i) initially the population size was N_{max} and (ii) the offspring solution did not dominate any solution and (iii) no solution of the archive dominates the offspring, then after the pruning mechanism discussed above, the archive size becomes $N_{max} + 1$. We resolve it by pruning a solution from the archive to make the archive N_{max} using *pruneOneSolution*.

IV. IMPLEMENTATION AND RESULTS

Proposed algorithm is compared with three existing multi-objective optimization algorithms: NSGA-II [1], SPEA2 [3] and AbYSS [20]. For this comparison purpose, bi-objective test problems ZDT4 and ZDT6 of ZDT family [26] and tri-objective DTLZ [27] problems 1, 2, 3, 4, 5, 6, and 7 are used. For measuring the performance of the resulting fronts we have used three metrics: Generational Distance (GD) [30], Generalized Spread (GS) [20] and Hypervolume (HV) [6].

They respectively evaluate the closeness of the obtained front to the true Pareto optimal front, the diversity of the obtained solutions and both the closeness and diversity, which is discussed in detail in [29]. We have implemented the proposed algorithm in Java using jMetal¹, a framework aimed at facilitating the development of meta-heuristics for solving multi-objective optimization problems [28].

a) *Generational Distance (GD)*: Van Veldhuizen and Lamont [30] proposed this metric. It measures how far the elements are in the set of nondominated vectors found from those in the Pareto optimal set. It is defined as follows.

$$GD = \frac{\sqrt{\sum_{i=1}^n d_i^2}}{n} \quad (6)$$

where n is the number of solutions in the set of nondominated solutions found so far and d_i is the Euclidean distance (in phenotypic space) between each of these solutions and the nearest member of the Pareto optimal set. When GD is zero, all the generated solutions are in Pareto front. To obtain the reliable results, nondominated solutions are normalized first before calculation of GD .

b) *Generalized Spread (GS)*: Spread metric is proposed by K. Deb et al. [1]. It is based on calculating the distance between two consecutive solutions, which works only for bi-objective problems. A. J. Nebro et al. [20] has extended it for more objectives as described in [31]. This extended version of spread is called generalized spread by the jMetal team.

Generalized spread can be defined as follows.

$$GS = \frac{\sum_{i=1}^m d(e_i, S) + \sum_{X \in S} |d(X, S) - \bar{d}|}{\sum_{i=1}^m d(e_i, S) + |S| * \bar{d}} \quad (7)$$

where S is a set of solutions, S^* is the set of Pareto optimal solutions, (e_1, e_2, \dots, e_m) are m extreme solutions in S^* , m is the number of objectives, and

$$d(X, S) = \min_{Y \in S, Y \neq X} \|F(X) - F(Y)\|^2 \quad (8)$$

$$\bar{d} = \frac{1}{|S^*|} \sum_{X \in S^*} d(X, S) \quad (9)$$

When GS is zero, the archived solutions are well distributed and include those extreme solutions. Before applying the metric, normalization is performed.

c) *Hypervolume (HV)*: Hypervolume calculates the volume (in phenotypic space) covered by members of a nondominated set of solutions Q for minimization problem [6]. For each solution $i \in Q$, a hypercube v_i is constructed with a reference point and the solution i as the diagonal corners of the hypercube. The reference point can be found by constructing a vector of worst objective function values. Then, a union of all hypercube is found and its hypervolume is computed as follows.

¹ jMetal can be freely downloaded from: <http://jmetal.sourceforge.net>.

```

1. pruneOneSolution(SolutionSet P, SolutionSet Q) { // P is the  $i^{th}$  front, Q is set of all solutions belonging to 1st front to  $(i-1)^{th}$  front
2.   if (problemType == BIOBJECTIVE) {
3.     Assign crowding distance to all  $p \in P$ 
4.     Find solution  $p \in P$  having minimum crowding distance
5.   } else {
6.     for each solution  $p \in P$  {
7.       Find first and second nearest distance (Euclidean distance in normalized objective space) of  $p$  assuming all solutions
       of the set  $((P \cup Q) - \{p\})$  as neighbours of  $p$ 
8.     } // end of for
9.     Find solution  $p \in P$  having minimum first nearest distance. If tie occurs pick one having minimum second nearest
       distance. If again a tie occurs, randomly pick one of them
10.  } // end of if-else
11.  Prune  $p$  and return all other solutions
12. } // end of pruneOneSolution

```

Figure 2. Pseudo code of *pruneOneSolution*

$$HV = \text{volume} \left(\bigcup_{i=1}^{|Q|} v_i \right) \quad (10)$$

A larger *HV* indicates a better solutions set. This metric is also computed by using normalized objective function values.

A. Test Configuration

We have compared the performance of our proposed algorithm with NSGA-II [1], SPEA2 [3] and AbYSS [20]. We have considered the results in [20] for NSGA-II, SPEA2 and AbYSS. Here we would like to mention that AbYSS [20] has also been implemented using jMetal² [28] and first five authors of AbYSS [20] belong to the jMetal team. In [20], the authors have mentioned that they have used respective authors' implementation of NSGA-II³ and SPEA2⁴.

The authors of [20] have used the following configuration for their AbYSS. $(I + 1)$ EA improvement method is used, with size of $P = 20$ and archive size = 100. Both the sizes of *RefSet1* and *RefSet2* are taken as 10. SBX crossover is used in the solution combination method with distribution index $\eta_c = 20$.

The real coded version of NSGA-II has been used and the parameter settings are same as used in [1]. SBX crossover and polynomial mutation have been used in NSGA-II. The distribution indices for SBX crossover and polynomial mutation are respectively $\eta_c = 20$ and $\eta_m = 20$. Crossover probability, $p_c = 0.9$ and mutation probability, $p_m = 1/n$ (where n = number of variables) have been used. Population size N , has been taken as 100.

The authors of AbYSS [20] have used the Zitzler's implementation of SPEA2 [3]. It is implemented within the PISA framework [33]. As the implementation of SPEA2 does not contain constraint-handling, the authors of [20] have forcefully modified the original implementation in order to include the same constraint-handling mechanism as used in NSGA-II and AbYSS. They have used the following parameter

values. Both the population and archive size is 100. The crossover and mutation operators are same as those used in NSGA-II, having the same values for their probabilities and distribution indices.

We have used the following settings for our proposed algorithm. The parameters values of DE-3 crossover are $F = 0.5$, $CR = 0.1$. Distribution index of mutation $\eta_m = 20$ with mutation probability $p_m = 1/n$ (n is number of variables) has been used. Maximum and minimum allowable archive sizes are $N_{max} = 100$ and $N_{min} = 4$.

We have considered 25000 function evaluations for each run of each of the algorithms: AbYSS, NSGA-II, SPEA2 and the proposed one. All these four algorithms are executed for 100 independent runs on 9 different test problems. For each problem and for each algorithm, we have taken the median and inter-quartile range (IQR) of the 100 run for each of the metrics. These values are provided in Table I, Table II and Table III respectively. Best values in the tables are indicated by the darkened space.

B. Results and Discussion

According to performance metrics generational distance (GD), generalized spread (GS) and hypervolume (HV), the proposed algorithm is the best compared to the other three algorithms in 6, 9 and 8 test problems respectively, out of the 9 test problems, shown in Table I, Table II and Table III. ZDT4, DTLZ1 and DTLZ3 are multimodal and ZDT4 and DTLZ6 have highly skewed search space and so they challenge the ability of an optimization algorithm. Test problems ZDT4 and DTLZ1 have respectively 99 and 161050 local Pareto optimal fronts. So, ZDT4, ZDT6, DTLZ1, DTLZ3 and DTLZ6 are considered as good benchmark test problems. The proposed algorithm always gives the best output for these five problems according to all of the three performance indicators. As shown in Table III, NSGA-II, SPEA2 and AbYSS cannot solve DTLZ3 and DTLZ6 test problems, since the median and IQR of HV values for these respective fields are 0 (zero), whereas the proposed algorithm can solve these two problems. Our algorithm gives the best diversity, measured by GS, in all these test problems, which is evident from Table II.

In many MOEAs, if the population size increases keeping the maximum number of function evaluations fixed, number of

² This implementation of AbYSS is available for download at: <http://neo.lcc.uma.es/metal/index.html>.

³ This implementation of NSGA-II is available for download at: <http://www.iitk.ac.in/kangal/soft.htm>.

⁴ This implementation of SPEA-2 is available for download at: <http://www.tik.ee.ethz.ch/pisa/selectors/spea2/spea2.html>.

TABLE I. MEDIAN AND INTERQUARTILE RANGE OF THE **GENERALATIONAL DISTANCE (GD)** METRIC (25000 FUNCTION EVALUATIONS)

Problem	Proposed Algorithm		AbYSS		NSGA-II		SPEA2	
	Median	IQR	Median	IQR	Median	IQR	Median	IQR
ZDT4	1.498e-04	3.4e-05	5.204e-04	3.9e-04	4.145e-04	3.5e-04	5.743e-02	3.3e-02
ZDT6	5.389e-04	3.1e-05	5.514e-04	1.8e-05	9.911e-04	9.8e-05	8.200e-04	7.3e-05
DTLZ1	9.689e-04	1.1e-04	4.516e-03	7.1e-02	1.819e-03	5.4e-03	1.087e+00	1.8e+00
DTLZ2	8.692e-04	1.2e-04	7.285e-04	6.8e-05	1.254e-03	1.9e-04	1.834e-03	4.7e-04
DTLZ3	2.882e-03	6.4e-03	6.502e-01	6.3e-01	1.693e+00	1.9e+00	7.905e+00	4.7e+00
DTLZ4	5.172e-03	3.0e-04	4.865e-03	3.7e-04	4.430e-03	4.3e-04	5.221e-03	2.0e-03
DTLZ5	3.126e-04	6.6e-05	2.497e-04	4.1e-05	6.342e-04	5.7e-05	7.096e-04	7.6e-05
DTLZ6	5.689e-04	3.2e-05	1.173e-01	4.2e-02	1.053e-01	1.1e-02	1.093e-01	1.3e-02
DTLZ7	1.721e-03	6.3e-04	1.880e-03	1.2e-03	2.512e-03	3.8e-04	3.470e-03	1.7e-03

TABLE II. MEDIAN AND INTERQUARTILE RANGE OF THE **GENERALIZED SPREAD (GS)** METRIC (25000 FUNCTION EVALUATIONS)

Problem	Proposed Algorithm		AbYSS		NSGA-II		SPEA2	
	Median	IQR	Median	IQR	Median	IQR	Median	IQR
ZDT4	1.100e-01	2.6e-02	1.273e-01	3.7e-02	4.028e-01	6.2e-02	4.550e-01	1.8e-01
ZDT6	7.338e-02	3.7e-02	9.002e-02	1.7e-02	4.356e-01	4.7e-02	1.578e-01	2.1e-02
DTLZ1	7.748e-02	2.0e-02	5.126e-01	1.6e-01	5.449e-01	1.1e-01	1.428e+00	5.9e-01
DTLZ2	6.118e-02	1.3e-02	5.000e-01	5.9e-02	5.201e-01	7.0e-02	1.042e-01	1.4e-02
DTLZ3	3.555e-01	3.1e-01	8.148e-01	2.0e-01	1.384e+00	6.2e-01	1.263e+00	2.9e-01
DTLZ4	7.278e-02	2.2e-02	4.634e-01	6.5e-02	4.860e-01	9.2e-02	1.153e-01	4.5e-01
DTLZ5	9.494e-02	2.0e-02	1.497e-01	2.1e-02	4.456e-01	7.0e-02	1.892e-01	2.8e-02
DTLZ6	1.051e-01	2.5e-02	7.059e-01	5.5e-02	6.326e-01	6.7e-02	2.756e-01	2.5e-02
DTLZ7	2.621e-01	2.7e-01	5.439e-01	1.0e-01	5.091e-01	6.0e-02	2.577e-01	3.8e-02

TABLE III. MEDIAN AND INTERQUARTILE RANGE OF THE **HYPERVOLUME (HV)** METRIC (25000 FUNCTION EVALUATIONS)

Problem	Proposed Algorithm		AbYSS		NSGA-II		SPEA2	
	Median	IQR	Median	IQR	Median	IQR	Median	IQR
ZDT4	6.616e-01	6.2e-04	6.553e-01	6.1e-04	6.555e-01	5.5e-03	1.068e-01	2.1e-01
ZDT6	4.014e-01	7.8e-05	4.004e-01	1.8e-04	3.860e-01	1.6e-03	3.926e-01	1.1e-03
DTLZ1	7.853e-01	2.0e-03	7.342e-01	6.8e-01	7.412e-01	1.8e-02	5.842e-01	6.5e-01
DTLZ2	4.142e-01	1.4e-03	3.837e-01	5.7e-03	3.770e-01	6.9e-03	3.938e-01	3.9e-03
DTLZ3	3.935e-01	9.4e-02	0.000e+00	0.0e+00	0.000e+00	0.0e+00	0.000e+00	0.0e+00
DTLZ4	4.078e-01	2.2e-03	3.881e-01	5.7e-03	3.771e-01	9.1e-03	3.869e-01	1.8e-01
DTLZ5	9.378e-02	7.6e-05	9.399e-02	4.0e-05	9.371e-02	2.5e-04	9.334e-02	3.3e-04
DTLZ6	9.480e-02	5.7e-05	0.000e+00	0.0e+00	0.000e+00	0.0e+00	0.000e+00	0.0e+00
DTLZ7	3.000e-01	4.0e-02	2.606e-01	3.1e-02	2.840e-01	4.6e-03	2.838e-01	6.2e-03

generations decreases. In the proposed algorithm, in each generation we evaluate only one offspring. So, the maximum possible generation is same as the number of function evaluation making it a steady state genetic algorithm. We have concentrated only on diversity and domination level respectively in mating selection and environmental selection. Thus after a few function evaluations, when the archive contains only nondominated set of solutions, during mating selection the most diverse solution is always chosen as the primary parent of the proposed DE-3 crossover. As the offspring generated in DE-3 crossover is very close to its primary parent, the probability to obtain a diverse solution is increased. So, the proposed mating selection enhances the probability to produce offsprings in more probable search region, reducing exploration in less probable region. Since, DE-3 crossover performs better than SBX crossover in higher dimensional variable space; our algorithm is well suited for problems having lengthy genotype. For more than bi-objective

space the use of first nearest distance to measure diversity makes the proposed algorithm suitable to address higher dimensional objective problems. Thus, the proposed approach addresses both the higher dimensional objective and variable space.

V. CONCLUSIONS AND FUTURE WORKS

In this paper we have introduced a new adaptive and dynamic size archive based steady state multi-objective genetic algorithm. A new mating selection and a new environmental selection have also been introduced in the algorithm. These selection strategies increase exploitation of existing solutions reducing the probability of exploration in less probable search region. A new variation operator DE-3 crossover has also been proposed. We have also proposed a new archive updating strategy. This strategy allows the algorithm to maintain an archive containing only

nondominated set of solutions. If the archive size falls below a minimum size, dominated set of solutions are added to fill the archive. Our algorithm is also applicable for constrained problems.

The proposed algorithm is compared with three existing multi-objective optimization algorithms: NSGA-II, SPEA2 and ABYSS using three performance measures on 9 standard test problems and it outperforms the other three algorithms.

All the test problems considered here are unconstrained and have maximum three objectives. In future, the proposed algorithm can be tested for more test problems including the constrained and more than three objective test problems. We are also interested to study the performance of the algorithm for real life problems.

REFERENCES

- [1] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II", *IEEE Trans. Evol. Comput.*, vol. 6, pp. 182–197, 2002.
- [2] KanGAL 200607. S. Kukkonen and K. Deb, "Improved pruning of non-dominated solutions based on crowding distance for bi-objective optimization problems", Technical report 7, Indian Institute of Technology, Kanpur, India, 2006.
- [3] E. Zitzler, M. Laumanns and L. Thiele, "SPEA2: Improving the strength pareto evolutionary algorithm", *Comput. Eng. Netw. Lab. (TIK)*, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, Tech. Rep. 103, 2001.
- [4] S. Tiwari, P. Koch, G. Fadel and K. Deb, "AMGA: An archive-based micro genetic algorithm for multi-objective optimization", in *Proc. Genetic Evol. Comput. Conf. (GECCO 2008)*, ACM, pp. 729–736, 2008.
- [5] S. Tiwari, G. Fadel and K. Deb, "AMGA2: improving the performance of the archive-based micro-genetic algorithm for multi-objective optimization", *Eng. Opt.*, Taylor and Francis, pp. 371–401, 2011.
- [6] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach", *IEEE Trans. Evol. Comput.*, vol. 3, pp. 257–271, 1999.
- [7] E. Zitzler, "Evolutionary algorithms for multiobjective optimization: methods and applications", Doctoral dissertation ETH 13398, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, 1999.
- [8] K. Deb and S. Tiwari, "Omni-optimizer: a generic evolutionary algorithm for single and multi-objective optimization", *Eur. J. Oper. Res.*, vol. 185, pp. 1062–1087, 2006.
- [9] H. Eskandari, C. D. Geiger and G. B. Lamnot, "FastPGA: a dynamic population sizing approach for solving expensive multiobjective optimization problems", in *Proc. Evol. Multiobjective Opt. Conf. (EMO 2007)*, Lecture notes on computer science, vol. 4403, Springer-Verlag, pp. 141–155, 2007.
- [10] D. W. Corne, J. D. Knowles and M. J. Oates, "The pareto envelop-based selection algorithm for multiobjective optimization", in M. S. et al. (Ed.), *Parallel Prob. Solving from Nature – PPSN VI*, Springer, pp. 839–848, 2000.
- [11] M. Kim, T. Hiroyasu, M. Miki and S. Watanabe, "SPEA2+: improving the performance of the strength pareto evolutionary algorithm 2", *Parallel Problem Solving from Nature - PPSN VIII*, Springer, pp. 742–751, 2004.
- [12] H. Lu and G. G. Yen, "Rank-density-based multiobjective genetic algorithm and benchmark test function study", *IEEE Trans. Evol. Comput.*, vol. 7, no. 4, pp. 325–343, 2003.
- [13] G. G. Yen and H. Lu, "Dynamic multiobjective evolutionary algorithm: adaptive cell-based rank and density estimation", *IEEE Trans. Evol. Comput.*, vol. 7, no. 3, pp. 253–274, 2003.
- [14] K. Deb, M. Mohan and S. Mishra, "Evaluating the ϵ -domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions", *Evol. Comput. J.*, vol. 13, no. 4, pp. 501–525, 2005.
- [15] C. A. C. Coello, "A comprehensive survey of evolutionary-based multiobjective optimization techniques", *Knowledge Information Systems*, vol. 1, no. 3, pp. 269–308, 1999.
- [16] C. A. C. Coello, "Evolutionary multi-objective optimization: a historical view of the field", *IEEE Comput. Intelligence Mag.*, vol. 1, no. 1, pp. 28–36, 2006.
- [17] C. A. C. Coello, "List of references on evolutionary multiobjective optimization", Technical report, CINVESTAV-IPN, <http://delta.cs.cinvestav.mx/ccello/EMOO>, 2009.
- [18] A. Konak, D. W. Cott and A. E. Smith, "Multi-objective optimization using genetic algorithms: A tutorial", *Reliability Eng. Sys. Safety*, Elsevier, vol. 91, pp. 992–1007, 2006.
- [19] D. W. Corne, N. R. Jerram, J. D. Knowles and M. J. Oates, "PESA-II: region based selection in multiobjective evolutionary optimization", in *Proc. 6th Int. Conf. Pparallel Prob. Solving from Nature PPSN-VI*, Springer-Verlag, pp. 839–848, 2000.
- [20] A. J. Nebro, F. Luna, E. Alba, B. Dorronsoro, J. J. Durillo, and A. Beham, "ABYSS: Adapting scatter search to multiobjective optimization", *IEEE Tans. Evol. Comput.*, vol. 12, no. 4, pp. 439–453, 2008.
- [21] O. Isao, S. Hiroshi and K. Shigenobu, "Areal-coded genetic algorithm for function optimization using the unimodal normal distribution crossover", *J. Jap. Society Artificial Intelligence*, vol. 14, no. 6, pp. 1146–1155, 1999.
- [22] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space", *Complex Systems*, vol. 9, no. 2, pp. 115–148, 1995.
- [23] K. Deb, A. Anand and D. Joshi, "A computationally efficient evolutionary algorithm for real-parameter optimization", *Evol. Comput. J.*, vol. 10, no. 4, pp. 371–395, 2002.
- [24] R. Storn and K. Price, "Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces", *J. Global Opt.*, vol. 11, no. 4, pp. 341–359, 1997.
- [25] W. L. Loh, "On Latin hypercube sampling", *Annals Stat.*, vol. 33, no. 6, pp. 2058–2080, 2005.
- [26] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results", *Evol. Comput.*, vol. 8, no. 2, pp. 173–195, 2000.
- [27] K. Deb, L. Thiele, M. Laumanns and E. Zitzler, "Scalable test problems for evolutionary multi-objective optimization", in *Evolutionary Multiobjective Optimization. Theoretical Advances and Applications*, A. Abraham, L. Jain, and R. Goldberg, Eds. Berlin, Germany: Springer-Verlag, pp. 105–145, 2005.
- [28] J. J. Durillo, A. J. Nebro, F. Luna, B. Dorronsoro, and E. Alba, jMetal: A Java Framework for Developing Multi-Objective Optimization Metaheuristics Departamento de Lenguajes y Ciencias de la Computación, University of Málaga, E.T.S.I. Informática, Campus de Teatinos, Tech. Rep. ITI-2006-10, 2006.
- [29] K. Deb, Multi objective optimization using evolutionary algorithms, New York, 2001.
- [30] D. A. Van Veldhuizen and G. B. Lamont, Multiobjective Evolutionary Algorithm Research: A History and Analysis Dept. Elec. Comput. Eng., Graduate School of Eng., Air Force Inst. Technol., Wright-Patterson, AFB, OH, Tech. Rep. TR-98-03, 1998.
- [31] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff, and E. Tsang, "Combining model-based and genetics-based offspring generation for multi-objective optimization using a convergence criterion", in *Proc. 2006 IEEE Congr. Evol. Comput.*, pp. 3234–3241, 2006.
- [32] S. Kukkonen and J. Lampinen, "GDE3: The third Evolution Step of Generalized Differential Evolution", in *Proc. IEEE Cong. Evol. Comput. (CEC 2005)*, vol. 1, IEEE Press, pp. 443–450, 2005.
- [33] S. Bleuler, M. Laumanns, L. Thiele and E. Zitzler, "Pisa—A platform and programming language independent interface for search algorithms", in *Proc. Conf. Evol. Multi-Criterion Opt. (EMO 2003)*, pp. 494–508, 2003.