A Many-objective Evolutionary Approach Using Fuzzy Dominance With Bidirectional Bias

Naheed Anjum Arafat, Siddhartha Shankar Das, Md. Monirul Islam, *Member IEEE*

***Abstract*—The performance of Pareto dominance based evolutionary algorithms degrades for optimization problems with many objectives (more than 3). This is due to the lack of comparability increasing among the solutions with the increase in number of objectives, since most of the solutions are pareto non-dominated.**

In this paper, we propose a new scalable evolutionary algorithm, called Many Objective Evolutionary Algorithm with Bidirectional Bias (MOEA/BB) for solving many-objective optimization problems. Here, the concept of fuzzy logic is used to define a fuzzy dominance relation which is further used to rank the solutions, thus offering comparability. Also, a novel next generation selection scheme is introduced to mitigate the selection bias inherent in the fuzzy dominance relation.

Further experiments were conducted to compare the performance of MOEA/BB with four state-of-the-art algorithms in solving eight WFG benchmark problems with 10, 13 and 19 objectives. The experimental results show that, MOEA/BB is able to find better solutions compared to other algorithms in most of the problems.

***Index Terms*—pareto dominance, many-objective optimization, fuzzy logic, fuzzy dominance, selection bias, evolutionary algorithms**

I. INTRODUCTION

MANY real world problems known as Multi-objective Optimization Problems (MOPs) involve multiple objectives to be satisfied concurrently considering several constraints. A pair of objectives may be conflicting or harmonious depending on their underlying correlation [1]. If there is a positive correlation between them improving one objective is suffice and we get a single solution meeting the both objectives. If they are conflicting instead of a single solution we get a set of solutions each of which can't be improved in one objective without deteriorating in another. Therefore, we have a set of solutions having trade-off among them and we say these solutions are non-dominated to one another. Evolutionary algorithms are particularly suitable for MOPs due to the property of achieving the entire set of solution in one single run.

Recently, MOPs of more than three objectives, commonly termed as Many-objective Optimization Problems (MaOPs) [2][3] has taken Evolutionary Multiobjective Optimization (EMO) researchers' attention. Such problems frequently appear in Scientific areas and Engineering and Industrial design; examples may be:

- Environmental and hydrolic engineering– *water resource engineering applications* [4], *water portfolio planning* [5], *Groundwater pollution remediation* [6], *location of site retail and service facilities* [7]
- Electrical and electronics engineering– *Synthesis of CMOS Opamp* [8]
- Telecommunication and network optimization– *Network Design* [9], *multicalst flows improve wire-antenna geometries* [10]
- Robotics and control engineering– *Fault diagnosis* [11], *Controller design*[12][13], *Design of controller system* [14][15][16][17][18], *Robotic manipulation problems* [19][20]
- Transport Engineering– *Train system* [21][22]
- *Environmental modelling* [23]
- Ecology– *Assessment of ecological models* [24][24]
- *Process planning* [25][26][27][28], *Time Tabling* [29], *Warehouse management* [30]

MaOPs are particularly challenging to the EMO community due to several reasons.

First, with increase in the number of objectives, increases exponentially the number of solutions required for approximating the entire *Pareto front* (PF). If n points are required for approximating a 1D front, then n^M points are required to approximate an M dimensional front. However, the PF of a K-objective optimization problem is a hyper surface of $K-1$ dimension (for bi-objective problems, the PF is a line or curve; for 3-objective problems, the PF is a 2D plane etc. except degenerate surface). Hence for an M-objective MOP n^{M-1} points are required for approximating the whole PF. This phenomenon is also known as the *curse of dimensionality* [31].

Second, visualization of the objective space is important to a decision maker since in any real world application, a final solution must be chosen from the obtained non-dominated solution set. Several multidimensional visualization techniques have been studied in the MCDM literature [32], of which Parallel Coordinate Plot [33][34] has been used extensively to visualize and interpret PFs interactively. With increasing number of objectives it becomes increasingly difficult to visualize and comprehend the PF. *Many Objective Visualization* is currently a promising research area for application to MOPs.

Third, a considerable number of EMO techniques have been proposed in the last few decades [35]. Specially, MOGA [36], NSGA [37], NSGA-II [38], SPEA [39], SPEA2 [40] have gained popularity among the researchers. All these algorithms share a common characteristic– they use either *Pareto dominance based ranking* or *Pareto dominance based counting*

Naheed Anjum Arafat is with Bangladesh University of Engineering and Technology, Bangladesh.

Siddhartha Shankar Das is with Bangladesh University of Engineering and Technology, Bangladesh.

Md. Monirul Islam is with Bangladesh University of Engineering and Technology, Bangladesh.

as primary selection criteria. However they have been proved to be inefficient for solving MaOPs in a number of studies [41][42][43][44]. Even taking large population cannot improve their performance. Attempts have also been made to improve their performance [45].

Ineffectiveness of Pareto based selection algorithms can be attributed to following reasons:

- State-of-the-art evolutionary algorithms rely on *Pareto dominance* relation for discriminating good solutions from bad ones. The assigned fitness value of a solution is a function of the number of solutions dominating it and/or the number of solutions dominated by it. The number of non-dominated individuals increases as the number of objectives increases as illustrated in [46]. As the number of objectives approaches infinity, the proportion of non-dominated space approaches 1. So, almost all individuals will tend to lie on the non-dominated region and be assigned equal fitness value with respect to *Pareto dominance* based criteria. As a result, popular algorithms will fail to provide strong selection pressure towards the true *Pareto front* of a MaOP.
- Some algorithms rely on the data structures that grow exponentially in size with increasing number of objectives. For example, in PAES [47] a map of each grid location (the number of grid location is r^K, if there are K objectives each divided into r division) is maintained, indicating amount of solutions residing in each grid location. For higher dimension in order to avoid the *curse of dimensionality*, these algorithms are forced to divide each objective into lower number of divisions. More solutions lie in the same grid. As a result, diversity suffers, although convergence increases.
- Some algorithms which rely on fixed size archive like PAES or SPEA happen to loss some solutions which might have dominated *archived solutions* at a later generation. This phenomenon is termed as *fitness deterioration* by Hanne [48].

Keeping the poor scalability of the state-of-the-art algorithms in mind, the primary objective is to devise an algorithm for solving MaOPs. Here, we assume, in high dimension almost all of the solutions are non-dominated. In light of this assumption, we can simplify three basic characteristics of our algorithm:

- The approximation set provided by the algorithm must be as close to the *Pareto front* as possible.
- The solutions in the approximation set must be distributed uniformly and as sparse as possible in the objective space.
- We assume no intervention from *Decision Maker* i.e. the algorithm assumes a posteriori approach to solving an MOP. So, the algorithm does not strive towards any particular preferred portion of the frontier or *region of interest* (ROI), rather it is expected to approximate the entire *Pareto front*.

The remainder of the paper is organized as follows. In Section II, an in-short background of the multi-objective optimization and fuzzy sets is reviewed. Afterwards, a detailed description of the proposed algorithm along with differences in working procedure with other existing algorithms are given in Section III. In Section V, we discuss the benchmark problems and performance measures used in our experimental studies. The performance of the proposed approach and its comparison with other algorithms are also demonstrated. Finally, some concluding remarks and an outlook to implied future works are presented in Section VI.

II. LITERATURE REVIEW

A. Multi-Objective Optimization:

A MOP including m kinds of objective function is defined as follows:

$$\text{Minimize/Maximize } y = f(x) = (f_1(x), f_2(x), \dots, f_m(x))$$
$$\text{Subject to } x \in F$$

Where x is a vector of decision variables i.e. the decision vector, F is the decision space, $f : F \rightarrow \mathbb{R}^m$ consists of m real valued objective functions, y is called the objective vector and $\mathbb{R}^m$ is called the m–dimensional objective space. If $x \in \mathbb{R}^n$, all the objectives are continuous and F is described by

$$F = \{x \in \mathbb{R}^n \mid h_j(x) \leq 0, j = 1, 2, \dots, k\}$$

where h_j are continuous constraint functions and n is the number of decision variable.

Solving a MOP is to determine from among the set F of all members– a particular subset $F^* = x_1^*, x_2^*, \dots, x_n^*$ which yields optimum values (minimum/maximum) for all the objective functions.

Definition 2.1: (Pareto Dominance) : Considering minimization problem, suppose $f(u), f(v) \in \mathbb{R}^m$. u is said to dominate v if and only if $f_i(u) \leq f_i(v)$ for every $i \in 1, \dots, m$ and $f_j(u) < f_j(v)$ for at least one $j \in 1, \dots, m$. Mathematically,

$$\forall i \in M : f_i(u) \leq f_i(y) \cap \exists j \in M : f_j(u) < f_j(v).$$

where $M = 1, 2, \dots, m$. When u dominates v, we denote $f(u) \succeq f(v)$. Pareto dominance is irreflexive, transitive and anti-symmetric.

Definition 2.2: (Pareto-optimality) : We say that a decision vector $x^* \in F$ is *Pareto-optimal* w.r.t F if it is dominated by no other x in F. $f(x)$ is called a *Pareto-optimal vector*. In other words, there exists no feasible decision vector $x \in F$ which would improve some objective without causing simultaneous decrease in at least one other objective. The set of all such *Pareto-optimal* points is called *Pareto-optimal Set* (PS) and the set of all such *Pareto-optimal vectors* is the *Pareto-optimal Front* (PF) [32].

Definition 2.3: (Ideal vector and Nadir vector) : An objective vector $z^{ideal} = (z_1^{ideal}, \dots, z_m^{ideal})$ constructed using the best values of objective functions in the complete Pareto-optimal front PF^* is called an ideal objective vector. On Contrary, Nadir vector is the vector $z^{nadir} = (z_1^{nadir}, \dots, z_m^{nadir})$ constructed using the worst values of the complete Pareto-optimal front. Hence, for minimization problems, we have $z_i^{ideal} = \min_{x \in PF} f_i(x)$ and $z_i^{nadir} = \max_{x \in PF} f_i(x)$

B. Fuzzy-Sets

Fuzzy sets were first introduced by Lotfi Zadeh [49] as extensions to classical set theory. Instead of binary membership to a set (i.e. belongs to or not belongs to), fuzzy set allows partial membership denoted by a real number between 0 and 1. Formally, a fuzzy set is a tuple (A, λ_A) where A is a set and $\lambda_A : A \rightarrow [0, 1]$. For each $x \in A, \lambda_A(x)$ is the degree of member of x in $A, x \in (A, \lambda_A) \leftrightarrow x \in A \wedge \lambda_A(x) \neq 0$.

C. Existing Approaches

It is hard to classify approaches devised by EMO researchers to handle many-objective optimization problems since the term 'many-objective optimization' coined by Farina in 2002 [2]. A diverse set of algorithms exist in the literature.

Modified pareto dominance approaches: The problem of incomparability among the solutions led the researchers to modify pareto dominance relation in order to offer comparability. Some distinguished works are— dominance area control [50], k-optimality [2], fuzzy pareto dominance [51], ϵ-dominance [52], pareto cone-dominance [53], pareto α-dominance [53]. Experiment shows that modified pareto dominance relations offer better convergence than pareto dominance.

Non-dominated ranking: Ranking Non-dominated solutions so that they can be further differentiated is not a new concept. Drechsler et al. [54] proposed relation *favour* which considers the number of objectives in which a solution is better than another as primary selection criteria. It was further modified to consider the amount of betterment in [55]. A set of non-dominated solutions are ranked according to each separate objective and assigned a fitness value as an aggregate function of individual ranks in each objectives in [56]–termed as 'ranking dominance'. Other ranking schemes include 'winning score' [57], di Pierro's k-optimality [4]. Knowles made a comparative study on different non-dominated ranking approaches in [58] and results indicated that 'average ranking' performs better. In [45] they showed that although convergence of NSGA-II increases, diversity suffers. Similar conclusion was drawn in [56].

Indicator based approaches: Indicator based performance metric has been used to measure the quality of non-dominated solution sets for long [59][60][61]. The hypervolume space covered by the non-dominated set, first used as performance measure in [62] has been incorporated in an evolutionary algorithm– IBEA(Indicator Based Evolutionary algorithm) by Zitzler et al. [63]. Wagner et al. [43] reported good result by IBEA for many-objective problems. The prime concern for IBEA is its large computation overhead incurred by hypervolume calculation. Bader and Zitzler devised an algorithm– HypE [64] for faster approximate hypervolume calculation based on Monte-carlo sampling, thus mitigating the computational burden. Recently, WFG devised by While et al. [65] calculates exact hypervolume much faster for more than 5 objectives although the worst case complexity is still exponential ($O(2^n)$ where n = total number of solutions). Another algorithm for S-metric calculation- HOY has worst case time complexity of $O(n^{M/2} * logn)$ where M = number of objectives and n = number of solutions. What makes hypervolume based EA's so popular is that hypervolume has the inherent property of representing convergence and diversity. A non-dominated set with better convergence and better diversity than another non-dominated set will always have hypervolume greater than the other.

Objective reduction: Dimensionality reduction is another method for inducing preference ordering over the points in a solution set. Not all pair of objectives are conflicting. Correlation may exists among them. Hence, if a many-objective problem can be reduced to a 2 or 3 objective instance keeping the problem characteristic intact, we can apply existing algorithms on them. This thought has led to approaches based on objective reduction. Deb and Saxena [66] proposed a principal component based (PCA) NSGA-II procedure which iteratively reduces objectives at each generation and then apply NSGA-II. PCA based NSGA-II performs better when a many-objective problem could be reduced to 2 or 3 objective. However if all the objectives of an M objective problem are conflicting, objective reduction is not possible and the approach is amenable to degraded performance. Brockhoff et al. [67][68] discussed under which condition objective minimization is possible, generalized the notion of conflicting objectives, redundant objectives and minimal objective set and proposed both exact algorithm and heuristic to determine the minimal objective set while preserving the dominance structure of the problem. The main problem of these approaches is their underlying assumption that objectives can be reduced. For many real world problems this assumption is not true.

Aggregation based approaches/Scalarizing function based approaches: These are non-pareto based EA. These approaches convert the set of objectives $(f_1, f_2, f_3, \ldots, f_n)$ into a single objective (F) by expressing F as a weighted sum or weighted Tchebycheff of objectives $(f_1, f_2, f_3, \ldots, f_n)$. And instead of minimizing these objectives, we minimize $F = (\sum w_j * f_j)$ –thus converting the many-objective problem into a single objective optimization problem. The main advantage of this approach is that it eliminates the need for exponential increase of non-dominated solution with increasing number of objectives. Also it is easier to calculate the weighted sum of objectives rather than computing the hypervolume. The main drawback of this approach is to determine a good set of weights since, it grows with the number of objectives. A number of algorithms has been proposed– MSOPS [69], MSOPS-II [70], MOEA/D [71].

Co-evolutionary approaches: Preference information i.e Decision makers priority can offer comparability among non-dominated solutions. Some preference based approaches include [72][73][74]. Preference information can also be incorporated as a set of goal vectors. Co-evolving the set of solutions with the set of goal vectors is another way of solving many-objective optimization problems. One realization of this approach is PICEA-g [75] where solutions will try to dominate goals and goals will try to be non-dominated by the solutions and their individual fitness is proportional to their individual criteria. Co-evolutionary approaches has been long used for solving MOP's. A comprehensive study can be found in [76]. The potential of co-evolutionary approaches in solving many-objective optimization problems, first demonstrated in [77], is still underrated and considered by the authors as a new avenue for exploration.

Fuzzy based approaches: Pareto dominance doesn't incorporate the degree of dominance in its optimality criteria. A solution will either be dominated (dominating) or non-dominated w.r.t another solution and another one will either be dominating (dominated) or non-dominated w.r.t the other. Farina and Amato [78] proposed *fuzzy optimality* where a solution can be dominated/dominating and non-dominated to a certain degree. Fuzzy membership function are used to fuzzify the difference of objective values in each objective and the degree of dominance of one solution over another is the sum of degree of dominance in individual objectives. Nasir et al. [79] suggested a new MOEA based on Decomposition with fuzzy dominance concept (MOEA/DFD) to compare two solutions and used the scalar decomposition method of MOEA/D only when one of the solutions fails to dominate the other in terms of a fuzzy dominance level. Koppen et al. [80] studied the fuzzification of the Pareto dominance relation to yield the degree of dominance between two solutions. The degree of dominance is calculated between each pair of solutions and each solution is assigned a ranking value (FPD ranking) within this set as the maximum of the all pair degree of mutual dominances. Hence, if a solution is dominated by at least one solution it will give a rank value of 1. Lowest rank individuals are considered best individuals. Based on this, an extension of the standard genetic algorithm, fuzzy-dominance-driven GA (FDD-GA), was proposed. In [51] it was demonstrated that FPD ranking resolves the problem of missing dominated points in higher dimension. Among recent approaches Zhenan et al. [81][82] proposed a new fuzzy based dominance and applied it in NSGA-II and SPEA2 instead of pareto dominance. Experimental result shows that it achieves better convergence and diversity than MOEA/D [71], NSGA-II [38], SPEA2 [40], FDD-GA [80] and other fuzzy based approaches. Rachmawati et al. [83] integrated decision makers preference as a set of imprecise goal vectors each of which may be defined with any of three degree of precision: small, medium and large.

Other approaches: A *Grid based Evolutionary algorithm (GrEA)* has been proposed by Yang et al. [3] where a user defined parameter (*div*) is used to partition the objective space between lowest and highest value in each objective into grid. Each solution has an associated *grid coordinate* and *grid dominance* (defined in terms of grid coordinate of two solutions) is used to offer comparability among the solutions. Grid Crowding distance (GCD), Grid ranking (GR) and Grid Coordinate point distance (GCPD) are taken into account while selecting the best individual. GCD ensures diversity and GR and GCPD ensures convergence. The basic drawback of this method is that its performance is sensitive to *div* parameter. If number of partition in each objective *(div)* is smaller, convergence suffers. For larger value of *div*, diversity suffers.

III. A Many-Objective Evolutionary Approach Using Fuzzy Dominance with Bidirectional Bias (MOEA/BB)

A. Motivation

The main issue regarding a MaOP is the lack of comparability among the solutions. Regarding this issue a fuzzy comparability mechanism is introduced which can classify otherwise incomparable solutions. The provided comparability is responsible for bias of the solutions towards different directions. Now, instead of letting bias take its own accord, a controlled mechanism is used. During fitness assignment in each generation, it is possible to forcefully bias the solutions towards targeted directions while keeping progress towards PF. Combining each such biased approach together ensures diversity among the solutions while keeping convergence towards PF.

B. Background Knowledge of Fuzzy Logic

We take the *fuzzification* of *Pareto dominance* approach to solve MaOPs. As discussed before, in conventional *Pareto dominance*, a solution can be *dominated* or *dominating* or *non-dominated.* However objective difference between a pair of solutions can also be interpreted as element of fuzzy set. A solution may dominate or be dominated by another solution by a certain degree simultaneously. For example, $(1,10), (3,5)$ are *Pareto non-dominated*, even if they are *non-dominated*, we say, $(3,5)$ is better than $(1,10)$ in a degree because $(3,5)$ is 2 worse than $(1,10)$ in the 1^{st} objective and 5 better in 2^{nd} objective considering minimization problem. The degree of goodness in 2^{nd} objective excels the degree of badness in 1^{st} objective. To measure this degree of goodness or badness we need *membership function*.

In this study, we define the *membership function* for each objective based on *Sigmoid membership function* (F_g) which is adaptive, means depending on the distribution of the value of the solutions in each objective it adapts its parameters for that objective. A *Sigmoid membership function* for an objective is shown in Fig. 1:

$$F_g(x) = A + \frac{(K-A)}{1+e^{\alpha(x-\mu)}} \tag{1}$$

Where A is the *lower-asymptote* which is 0 for our case. $(K-A)$ set to 1 which is the *upper-asymptote*. Hence, the output value will be bounded by 1 to 0. Here μ is the value where maximum growth occurs and α is the growth rate. Therefore,

$$F_g(x) = \frac{1}{1+e^{\alpha(x-\mu)}} \tag{2}$$

Each objective has a *membership function* of this type. These *Sigmoid membership functions* are then applied in the definition of *fuzzy dominance* relation.

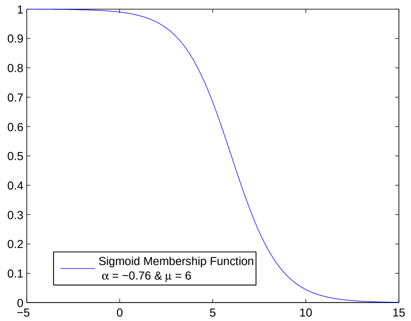

Fig. 1. A simple *Sigmoid Membership Function* with $\alpha = -0.76$ & $\mu = 6$

Since, dominance in each objective has been defined as a fuzzy set, they can be easily combined. Two basic operators are frequently used in fuzzy logic literature - fuzzy *'and'* operator or *t-norms* and fuzzy *'or'* operator or *s-norms*. There are several *t-norm* and *s-norm* operators in fuzzy logic literature. A comprehensive study on them can be found in [84].
Algebric product (as t-norm) and *Algebric sum (as s-norm)* deserves special attention. They are defined as follows:
Algebric product = $\{(x, \mu_A(x) * \mu_B(x)) \mid x \in X\}$
Algebric sum = $\{(x, \mu_A(x)+\mu_B(x)-\mu_A(x)*\mu_B(x)) \mid x \in X\}$
The intersection or Algebric product of *fuzzy dominance* values in each objective indicates how much a solution dominates another.

C. Fuzzy dominance Relation

Consider two objective vectors, $u(u_1, u_2, \cdots, u_k)$ and $v(v_1, v_2, \cdots, v_k)$ for a k-objective minimization problem.

We define $d^i_{u,v} = u_i - v_i$ where $i = 1, 2, 3, \dots, k$. $dom^i_{u,v} = F_g(d^i_{u,v})$ represents the fuzzy dominance degree of u over v in i^{th} objective.

Fuzzy *t-norm* operator is used to combine the fuzzy values, $\phi_{u,v} = \prod_{i=1}^{k} dom^i_{u,v}$ which indicates how much u fuzzy dominates v.

Similarly, $\phi_{v,u} = \prod_{i=1}^{k} dom^i_{v,u}$ indicates how much v fuzzy dominates u.

We say, u dominates v ($u \succeq_F v$) iff $\phi_{u,v} > \phi_{v,u}$ and u and v are non-dominated if $\phi_{u,v}$ = $\phi_{v,u}$. Fuzzy dominance relation is irreflexive, asymmetric like Pareto dominance relation but unlike Pareto dominance it is non-transitive. Non-transitivity is particularly useful in solving MaOPs because it doesn't impose a total ordering over the solution set.

The following example explains how fuzzy domination works in conjunction with these *membership functions*.
Initially, $\phi_{u,v} = 1$ and $\phi_{v,u} = 1$. It means solution u completely dominates solution v and solution v completely dominates solution u. At this stage they are fuzzy non-dominated. Normalization rule is defined for the two solutions as
$\phi^{Norm}_{u,v} = \frac{\phi_{u,v}}{\phi_{u,v}+\phi_{v,u}}$ and $\phi^{Norm}_{v,u} = \frac{\phi_{v,u}}{\phi_{u,v}+\phi_{v,u}}$
Therefore, normalization at this stage results, $\phi^{Norm}_{u,v} = 0.5$ and $\phi^{Norm}_{v,u} = 0.5$.

Now, consider for i^{th} objective $u_i < v_i$. $x = u_i - v_i < 0$ and $y \to 1$. Therefore, $dom^i_{u,v}$ will be still close to 1 i.e. u still dominates v completely. On contrary, $x = v_i - u_i > 0$ and depending on the magnitude, y is within 1 and 0. x for other pair of solutions is $\bar{x}$, on average. Now, the smaller x is relative to $\bar{x}$, the closer y will be to 1. Similarly, the larger x is relative to $\bar{x}$, the closer y will be to 0. So, $dom^i_{v,u}$ will still be between 1 and 0. It indicates how much v dominates u in i^{th} objective.
Say, $y = 0.5$, means $dom^i_{v,u} = 0.5$. Since, $dom^i_{u,v} = 1$ and $dom^i_{v,u} = 0.5$, it indicates u dominates v in i^{th} objective.

Again, say, for j^{th} objective $u_j > v_j$. $y = 0.5$ means $dom^j_{v,u} = 0.5$. Since, $dom^j_{u,v} = 0.2$ and $dom^j_{v,u} = 1$, we say, v dominates u in j^{th} objective.
After inspecting these 2 objective, $\phi_{u,v} = 1 \times dom^i_{u,v} \times dom^j_{u,v} = 1 \times 1 \times 0.2 = 0.2$ and $\phi_{v,u} = 1 \times dom^i_{v,u} \times dom^j_{v,u} = 1 \times 0.5 \times 1 = 0.5$.

We say, v fuzzy dominates u in i and j objectives altogether. This concept can be further extended to any number of objectives. According to the normalization rule defined earlier, $\phi^{Norm}_{u,v} = 0.28$ and $\phi^{Norm}_{v,u} = 0.72$. Therefore, we say, u dominates v by degree 0.28 and v dominates u by degree 0.72, although they are *Pareto non-dominated*. A simple algorithmic procedure is shown in Algorithm 1.

Algorithm 1 Fuzzy Domination

```
Input: Two input solutions $u$ and $v$
FIZZY DOMINATES
$\phi_{u,v} = 1$
$\phi_{v,u} = 1$
for $i = 1$ to $m$ do
    Compute $dom^i_{u,v}$ and $dom^i_{v,u}$
    $\phi_{u,v} = \phi_{u,v} \times dom^i_{u,v}$
    $\phi_{v,u} = \phi_{v,u} \times dom^i_{v,u}$
end for
Output: How much $u$ dominates $v$ and $v$ dominates $u$
```

D. Adaptive definition of membership functions

The *membership function* is used to fuzzify the difference of two solutions in a particular objective. Now instead of defining a fixed parameter *membership function* for all objectives (in that case function parameters will be problem dependent), we define different *membership function* for different objectives. The parameters of each *membership function* are dependent on the distribution of the solutions in that particular objective.

The *membership function* determines how much one solution dominates another with respect to all such dominations for a particular objective. Different pair of solutions produce different dominance degree and the shape of the function depends on its distribution.

For estimating parameters of the *membership function* for a particular objective, absolute differences of solution value for that objective has been taken for all distinct pair solutions.

To estimate α (growth rate) and μ (point of maximum growth) the *membership function* is defined in such a way that it gives value 0.5 when x ($d^i_{u,v}$) is equal to the average difference. If their difference is close to average difference of such pair of solutions, we say they dominate each other by degree 0.5.

Therefore, μ = average of all pair difference in a particular objective.

For a particular objective, the distribution of a set of objective values can be skewed in any side like left, middle or right depending on their mean. Therefore, objective differences can also be skewed to a side. α has been defined such that at $(\mu - 2\sigma)$ the output value returns 0.99. If we set 0.99 at the minimum difference and 0.01 at maximum difference then it could be biased towards a single difference.

Defining α in this way reduces the sensitivity to extreme standalone objective values. The reason behind choosing *Sigmoid function* as the *membership function* is: non-linearity of this function helps differentiate among the most common differences which are likely to lie between the range $(\mu \pm 2\sigma)$.

The mean and variance are numerically computed by Knuth's running mean and running variance estimation algorithm [85] which is $O(n)$. Therefore, the overall performance of the algorithm doesn't degrade.

E. Ranking based on fuzzy dominance

Assuming there are n individuals in the competition pool, every individual is paired with other $n-1$ individuals respectively, forming $n-1$ pairs of competition. For individual u, in each of $n-1$ pairs, it is compared with the other individual v using fuzzy-dominance relation and both $\phi_{u,v}$ (How much u dominates v) and $\phi_{v,u}$ (How much v dominates u). Then, fuzzy product value is normalized as $\frac{\phi_{v,u}}{\phi_{u,v}+\phi_{v,u}}$,which represents domination performance of v with respect to u. After calculating each pair, all performance values of the amount by which v dominates are summed together to obtain a sum value. This sum value is then divided by $(n-1)$ and the divided value is regarded as the fitness value that is used for ranking the solutions. Solutions are then sorted into ascending order by the fitness value. So the first solution in the sorted order is considered as the best solution in terms of being the least dominated by others. The ranking procedure is shown in Algorithm 2.

Algorithm 2 Ranking Process

Input: A competition pool containing n individuals

RANKING

```
for $i = 1$ to $n$ do
    $S_i = 0$ {calculate sum value of point solution $i$}

    for $j = 1$ to $n$ do
        $p = \frac{\phi_{j,i}}{\phi_{i,j}+\phi_{j,i}}$ {calculate performance value of $i$ with $j, j \neq i$}
        $S_i = S_i + p$
    end for
    $S_i = \frac{S_i}{n-1}$
end for
Sort Individuals by the $S$ value in ascending order.
```

Output: Solutions sorted in ascending order by how much they are dominated by others.

F. Next Generation Selection Scheme

Our selection scheme is divided into two strategy: In one, we keep the convergence of the solutions while keeping their diversity biased towards outer extremes and in another we keep the convergence of the solutions while keeping their diversity biased inwards. So combining this two approach we can maintain the diversity while converging towards the entire *Pareto front*.

Consider A pool of $2n$ non-dominated solutions from which n solutions have to be selected as next generation.

The next generation is taken in two sweeps. In the first sweep, $(n+d)$ solutions are selected from $2n$ solutions. After that, n solutions out of these $(n+d)$ solutions are selected. Here d is chosen heuristically. Typically d is $n/4$. To select $(n+d)$ solutions in first sweep, we scan through the sorted ranked list of solutions where solutions are sorted in ascending order of the degree by which they are dominated. Two crowding distance comparators have been defined. One will sort the solutions based on crowding distance in descending order from high value to low value, while keeping corner solutions (solutions whose crowding distances are infinity) in the last of the sorted list i.e. giving them the least priority. We call this comparator *CDCornerPointMinPriority*. The another one will sort them in descending order while keeping those solutions with infinite crowding distances (extreme solutions) in the first positions in the sorted list. In other words, this operator termed as *CDCornerPointMaxPriority* gives the most priority to the corner solutions. Therefore, the selection bias of the two comparators are in two different directions.

For inward tendency, consider m number of objectives and we maintain a list of solutions (*Takensolutionlist*) of initial size 0. Scanning process is continued until half-way of non-dominated solution set (*S*), considering $2m$ solutions at a time. $\lceil m/2 \rceil$ best solutions selected according to *CDCornerPointMinPriority* comparator are added to $Takensolutionlist$ and discarded from S simultaneously. The whole process is continued until the number of taken solutions becomes $(n+d)$.

The selection of $m/2$ solutions from $2m$ solutions is done by first assigning crowding distance to the $2m$ solutions. Solutions are then sorted by *CDCornerPointMinPriority* and we take first $m/2$ solutions from the sorted list. Now in this method we take the required $(n+d)$ solutions during the first sweep. Then for selecting the n solutions among them, we reassign *crowding distance* value among them and sort them using *CDCornerPointMinPriority* so that any crowded solution that may be taken into first sweep are eliminated from final solution set. And we also give least priority to the border solutions. The reason is twofold- first, it has been experimented that giving least priority to the border extreme solutions helps keep diversity in many objective problem. Second, it also helps keep the bias inward.

Here we ignore the solutions that resides in the corner, instead we take solutions that have greater crowding distance but don't reside in the corner. Now as the scanning length is half of solution list length, multiple scans will be needed and solutions are taken mostly from the front of the ranking list. So the convergences of solutions are ensured. At the same time, each time taking solutions in this manner keeps the bias to the inwards. Algorithm 3 show how the procedure works.

Algorithm 3 Selection Scheme with Inward Bias

```
Input: Nondominated Solution Set ($S$) ranked by their degree of being dominated by others, The number of solutions required to be selected from this solution set ($N$)
SELECTION SCHEME I
$Extra = size(S) - N$
$Takensolutionlist = \emptyset$
$TakeInFirstSweep = N + Extra/4 \; \{(n + d)\}$
$maxTake = \lceil m/2 \rceil$
while $size(Takensolutionlist) < TakeInFirstSweep$ do
    $left = 0$
    $right = 2m$
    $k = \lceil size(S)/2 \rceil$
    for $i = left$ to $k$ do
        Reassign Crowding Distance to the solutions from $s[left]$ to $s[max(right, size(S))]$
        $Localbest$ = $maxTake$ number of population taken from $s[left]$ to $s[max(right, size(S))]$ according to CDCornerPointMinPriority
        $Takensolutionlist = Takensolutionlist \cup Localbest$
        $left = right$
        $right+ = 2m$
    end for
    $S = S \setminus Takensolutionlist$
end while
Sort the solutions taken on first sweep by CDCornerPointMinPriority and take the required $N$ population from the sorted solution set.
Output: $N$ population with inward tendency.
```

For outward tendency, Consider again the number of objectives m, and we maintain a list of solutions, *Takensolutionlist* of initial size 0. Scanning process is done until the end of Nondominated Solution Set (*S*), considering $2m$ solutions at a time. m best solutions selected according to the *CDCornerPointMaxPriority* are added to $Takensolutionlist$ and discarded from S simultaneously. Sweeping over the nondominated set (S) in this way is continued until the number of taken solution taken become $(n + d)$.

The selection of m best solutions is done by first assigning *crowding distance* between the $2m$ solutions. Then they are sorted using comparator *CDCornerPointMaxPriority* and first m of them are selected. So for an m objective problem it will select the corner solutions among the first $2m$ solutions.

Note that selecting among the first $2m$ in ranked gives priority to convergence. And while taking the m solutions in this manner from the $2m$ means we give priority to outward solutions locally.Now in this method we take the required $(n + d)$ solutions during the first sweep.

In second sweep, in order to select n solutions from $n + d$ solutions, we reassign *crowding distance* value among them and sort them using *CDCornerPointMaxPriority* so that any adjacent solutions that may be taken in first sweep are eliminated and solutions are biased outwards. Algorithm 4 shows how the procedure works.

Algorithm 4 Selection Scheme with Outward Bias

```
Input: Nondominated Solution Set ($S$) ranked by their degree of being dominated by others, The number of solutions required to be selected from this solution set ($N$)
SELECTION SCHEME II
$Extra = size(S) - N$
$Takensolutionlist = \emptyset$
$TakeInFirstSweep = N + Extra/4 \; \{(n + d)\}$
$maxTake = m$
while $size(Takensolutionlist) < TakeInFirstSweep$ do
    $left = 0$
    $right = 2m$
    $k = size(S)$
    for $i = left$ to $k$ do
        Reassign Crowding Distance to the solutions from $s[left]$ to $s[max(right, size(S))]$
        $Localbest$ = $maxTake$ number of solutions taken from $s[left]$ to $s[max(right, size(S))]$ according to CDCornerPointMaxPriority
        $Takensolutionlist = Takensolutionlist \cup Localbest$
        $left = right$
        $right+ = 2m$
    end for
    $S = S \setminus Takensolutionlist$
end while
Sort the solutions taken on first sweep by CDCornerPointMaxPriority Take the required $N$ population from the sorted solution set.
Output: $N$ solutions with outward tendency.
```

Combining these two approaches we can keep offspring generation tendency outward and inward while keeping the convergence. Therefore, ultimately it will keep the diversity among the solutions while maintaining selection pressure towards the *Pareto-optimal front*.

The pseudocode for the proposed MOEA/BB algorithm is presented in Algorithm 5

IV. Difference with other work

There are some non-trivial differences between the proposed method and other fuzzy-based approaches. Farina and amato [78] compared solutions based on the number of objectives and degree of betterment in each objective. Three fuzzy sets are defined and three membership functions are defined to quantify them: better than,worse than and equal. Two types of membership function (linear and Gaussian) has been advised to use and DM will specify their parameters according to

Algorithm 5 Many Objective Evolutionary Algorithm with Bidirectional Bias

```
Input: population size ($N$); max. generation ($G$); search space
MAIN LOOP OF MOEA/BB
1: Initialize populations randomly. Generation count, $t = 0$.
2: A parent population $P_t$ is used to create mating pool.
   Parents selected using Binary Tournament are recombined
   and mutated to create a child population $Q_t$ of size $N$.
3: $R_t = P_t \cup Q_t$. The population $R_t$ will have size $2N$.
4: Compute the parameters of Sigmoid membership function
   for each objective.
5: Rank the solutions using fuzzy dominance.
6: Next generation is selected using bidirectional bias. Next
   generation population $|P_{t+1}| = N$ is selected using
   inward bias in even generations and using outward bias in
   odd generations.
7: $P_{t+1}$ is used as next generation parent. The process is
   continued from Step 2 until $t$ reaches max. generation, $G$.
Output: Non-dominated solutions at generation $T_G$.
```

domain knowledge. *t-norm* and *t-conorms* are applied in optimality criteria. Nasir et al. [79] calculated fuzzy dominance level between two solution by passing their objective difference to an exponential membership function and using product operator to combine them. Zhenan et al. [82] used left *Gaussian membership function* to fuzzify the objective differences. Fuzzy product operator is used and a problem-dependent heuristically chosen threshold value is used to rank the solutions. NSGA-II is used as baseline algorithm.

Our approach is different due to the following reasons-

- All fuzzy based approaches share a common property they somehow fuzzify the objective differences and combine those values using some operator. Our approach is not an exception. But none of them considered that different objective might have different trade-off magnitude. As a result, the magnitude might not be same and their differences might not follow the same distribution. Using the same membership function to fuzzify them would fail to offer proper comparability between two solutions. For example, consider the of objective values for 3 solution in Table I (range of each objective is shown in brackets). *Solution2* is better than *Solution1* in 1 objective, whereas *solution3* is better in 3 objectives as shown in Fig. 2. If membership function is defined in $[1, 5]$ range (if difference tends to 5, fuzzy value tends to 1). As 1 is the asymptote of the membership function, for difference value beyond 5 fuzzy value increases very slowly. As a result, $\phi_{2,1}$ will be very little different from $\phi_{3,1}$ which naturally shouldn't be.
- Although only fuzzifying Pareto relation would help converge the solutions which is the prime issue for Many-objective problems, still there remains a separate unresolved issue- diversity. Therefore, a separate diversity maintenance scheme is required. Nasir et al. [79] used MOEA with decomposition, Zhenan et al. [82] used NSGA-II as the baseline algorithm. There is no diversity maintenance mechanism in FDD-GA in [80] by Koppen et al. We used prioritized crowding distance scheme for maintaining diversity. Since, only sampling solutions with high crowding distance and high rank might maintain high density around corners and low density around middle of the Pareto front.

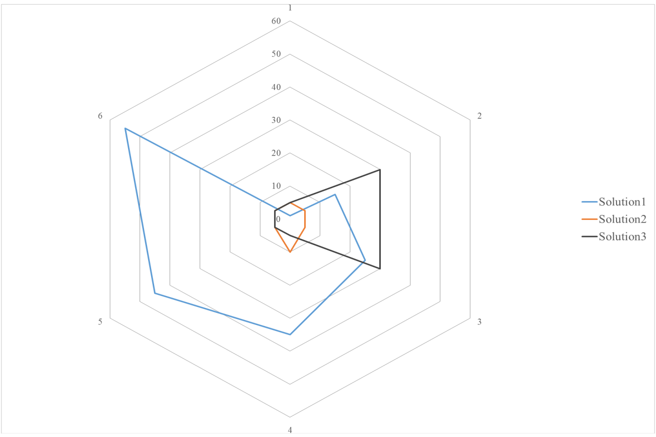

Fig. 2. Differentiability of 3 Nondominated Solutions

V. EXPERIMENTAL STUDIES

To understand the efficacy of our algorithm we have to compare it with several MOEA on existing benchmark problem suits. Although many algorithms require special parameter settings, some generic EA parameter settings remains the same.

A. *Selected MOEAs for Comparison*

We have selected PICEA-g [75], GDE3 [86], HypE [64], MOEA/D [71] as the competitors of our algorithm. PICEA-g has been selected because it is one of the latest algorithms which has shown promising results on several many-objective scalable benchmark problem suits. GDE3 is an improved version of differential evolution based evolutionary algorithm GDE [87]. HypE and MOEA/D have been selected as contenders from indicator based and aggregation based evolutionary approaches. Both of them have proved their worthiness in solving MaOPs in their respective areas.

B. *Benchmark Problems*

WFG toolkit [88] is the benchmark problem suit for our experimental studies. WFG2-WFG9 have been used in our study. WFG test suit has been chosen because all WFG problems are scalable to high dimension with the following additional features

- No extremal or medial parameters.
- Have known pareto optimal sets i.e known analytic solution.
- Scalable number of parameters.
- Dissimilar tradeoff ranges i.e. different objective values has different domain magnitude.
- Have no constraints.

TABLE I
SOLUTION VALUES

	Obj1 [1, 5]	Obj2 [5, 30]	Obj3 [5, 30]	Obj4 [5, 35]	Obj5 [5, 45]	Obj6 [5, 55]
Solution1	1	15	25	35	45	55
Solution2	5	5	5	5	5	5
Solution3	5	30	30	5	5	5

WFG2 is convex, WFG3 is linear and others have a concave pareto front. In terms of separability, only WFG4, WFG5 and WFG7 are separable. WFG2 is disconnected and WFG3 is degenerate. In terms of modality only WFG2, WFG4 and WFG9 are multimodal i.e. they have multiple local optima. WFG5 and WFG9 are deceptive making the global optima in an unlikely place. WFG2, WFG3, WFG6, WFG7, WFG8 are unimodal.

C. Performance Metrics

Generational Distance (GD): The concept of generational distance (GD) was defined by Veldhuizen and Lamont (1998) as a way of estimating how far is PF_{true} (true/actual pareto front) from PF_{obtain}, (obtained final non-dominated set) and is defined as: $GD = \sqrt{(\sum_{i=1}^{n})d_i^2}$

Where n is the *cardinality* of PF_{obtain} and d_i is the *Euclidean phenotypic distance* between each of these and the nearest member of PF_{true}. $GD = 0$ indicates that all the solutions are on the *true pareto front*. Lower value of GD means better convergence.

Inverted Generational Distance (IGD): Another convergence metric. But distance is measured with respect to PF_{true} i.e how far the PF_{obtain} is from PF_{true}. Hence, $IGD = \frac{\sqrt{(\sum_{i=1}^{n} d_i^2)}}{n}$

Here, n is the cardinality of PF_{true} and d_i is the *Euclidean phenotypic distance* between each of these and the nearest member of PF_{obtain}. In case, PF_{obtain} has few points and clustered together GD value will be small but IGD will still be large.

Space Uniformity: Suppose, $d_1, d_2, \ldots, d_n$ are defined as follows: $d_i = min(D_{i,j}|i \neq j)$ here, D_{ij} means *manhattan distance* between solution i and solution j. Space uniformity is the standard deviation of $d_1, d_2, \ldots, d_n$.

HyperVolume(HV): For an m-dimensional objective space *HyperVolume* is the volume bounded by the *nadir vector* and PF_{obtain} in Euclidean m space.

HyperVolume Ratio (HVR): Suppose,H_1 be the *HyperVolume* of the non-dominated set PF_{obtain} and H_2 be the HyperVolume of PF_{true}. HR is defined as $\frac{H_1}{H_2}$. HR will always be in the interval $[0, 1]$.

To measure convergence in our study Generational Distance (GD) and Inverted Generational Distance (IGD) metric is used, while Space Uniformity (SU) is used to measure diversity. Hypervolume Ratio (HR) is used for measuring both convergence and diversity [89]. These performace measures are widely used in the EMO literature. A reference set is required for computing GD, IGD and HR. Analytic solutions to the WFG problems are known. We generated 100000 pareto optimal solutions uniformly in the decision space using the analytic form of a pareto optimal solution. The source code of the toolkit has been obtained from WFG Website[1].

A reference point is required in Hypervolume (HV) calculation. Theoretically, the worst possible point in the feasible objective space i.e the worst point is chosen as the reference point. Practically, for most problems there is no upper bound of the objective space or even if there is one, it might be unknown. One suggestion is to take the worst value in each objective from any of the sets being compared. Then generate 50000 random samples in between the reference point and each of 50000 randomly chosen (with replacement) pareto optimal points. The proportion of number of sampling points being dominated by individuals of a population is the hypervolume ratio of that population. Such method of approximating hypervolume has been used in [3] and [64].

D. Experimental Setup

Before discussing algorithm settings, a discussion on benchmark problem settings is in order. We have run 8 WFG problems (WFG2-WFG8) with 10, 13 and 19 objectives. All of them have 54 decision variables, first 18 of which are position related parameters (k) and the rest are distance related parameters (l). The performance of our algorithm- Many Objective Evolutionary Algorithm with Bidirectional Bias (MOEA/BB) is going to be compared with 4 other competitive algorithms: GDE3, HypE, MOEA/D, PICEA-g. The source code for HypE has been obtained from PISA[2]. The source code for GDE3 is available in jMetal Framework[3] and MOEA/D is available in MOEA Framework[4]. The Matlab code for PICEA-g has been collected from Wang[5]. Each of these algorithms has been run on some general parameter settings and their respective algorithm-specific parameter settings. The generic parameter settings used in our experimental studies and shown in Table II are same for all algorithms being compared unless stated otherwise. *Population size* is set to 250, 75000 function evaluation is accomplished and 20 run of each algorithm is performed in the same experimental setup for a certain problem with a certain dimension. *Simulated Binary Crossover (SBX)* with Crossover probability of 1 and *distribution index* of 15 are used. *Polynomial Mutation (PM)* with mutation rate of

[1] http://www.wfg.csse.uwa.edu.au/toolkit/

[2] http://www.tik.ee.ethz.ch/pisa/

[3] http://jmetal.sourceforge.net/

[4] http://www.moeaframework.org/

[5] http://www.sheffield.ac.uk/acse/staff/rstu/ruiwang/index

$1/(numberofdecisionvariable)$ and *distribution index* of 20 are used. Unless stated otherwise, *Binary tournament selection* is used as selection scheme in the algorithms.

TABLE II
General Parameters

Objectives, M	10, 13, 19
Position Parameter, k	18
Distance Parameter, l	36
Decision Variable, n	$n = k + l$
Population Size, N	250
Max. Generations, *maxGen*	300
Crossover Operator	Simulated Binary Crossover, $(p_c = 1, n_c = 15)$
Mutation Operator	Polynomial Mutation, $(p_m = 1/n, n_m = 20)$

Now, some algorithm-specific parameter settings are to be noted. *Differential Evolution Crossover* is used in GDE3 with *crossover operation control* parameter value of 0.5 and *scaling factor* for mutation as 0.5. *Differential Evolution Selection* scheme is used. For HypE $(50, 50, \ldots, M^{th}$ 50) is used as the reference point for *monte-carlo sampling*. $M \times 100$ number of sample points is generated and used for estimating hypervolume. The reference point is chosen keeping in mind that the maximum objective is 19 in our study and the objective values at each objective cant exceed 40 for 19 objective WFG problems. In case of MOEA/D, the size of the *neighbourhood (T)* is chosen 8% of the population(20), the maximum number of population slots a solution can replace (n_r) has been chosen 10% of T (2). PICEA-g is run with $M \times 100$ goal vectors (Ngoal). Sensitivity analysis of parameters for PICEA-g, HypE and MOEA/D can be found in [75].

E. Experimental Results

Table III-X show the GD, IGD, Spacing and Hypervolume comparison among the five algorithms considered here including MOEA/BB. All the values are the arithmetic mean of 20 runs of each algorithm. Best performing values are bold faced here. In terms of mean GD, MOEA/BB outperformed other algorithms significantly in all problems. Since, fuzzy dominance can continuously differentiate among the solutions, fast convergence is quite expected. Only HypE has a slightly better mean IGD than MOEA/BB in one 10 objective WFG3 problem. WFG3 has been particularly difficult for MOEA/BB because of its degenerate nature. In spite of having fantastic value in GD metric, Spacing of MOEA/BB is not that bad in 10 objective problems. PICEA-g and GDE3 are the best contender of MOEA/BB in Spacing metric. PICEA-g and MOEA/BB are either best or second best in almost all problems. MOEA/BB achieves better mean hypervolume than its nearest competitor PICEA-g in six out of eight 10-objective instances.

Fig. 3 shows a boxplot of the Hypervolume results for 10 objective instances.

TABLE III
Comparison among *GDE3, MOEA/D, HypE, PICEA-g, MOEA/BB* for **WFG2**. Each results represents average of 20 independent runs. The best results are highlighted with boldface type.

	Algorithm	GD	IGD	SU	HYP
Obj10	GDE3	7.71043	0.11080	1.24971	0.699322
	MOEA/D	2.53110	0.08294	1.14524	0.208151
	HypE	3.10068	0.05297	1.21159	0.594319
	PICEA-g	1.44936	0.00907	**0.96048**	0.86909
	MOEA/BB	**0.17587**	**0.00353**	1.08698	**0.8970**
Obj13	GDE3	49.42019	0.71555	1.53544	0.6149
	MOEA/D	12.59845	0.47863	1.34418	0.08003
	HypE	14.72979	0.26894	1.42528	0.434222
	PICEA-g	10.86821	0.04795	**1.12715**	0.83589
	MOEA/BB	**0.01498**	**0.00128**	1.19467	**0.8510**
Obj19	GDE3	2340.62	35.8	2.03112	0.55207
	MOEA/D	477.77375	22.44564	1.65526	0.00617
	HypE	310.26787	11.62374	1.85976	0.21137
	PICEA-g	645.83520	2.28593	**1.41514**	0.77352
	MOEA/BB	**71.44717**	**0.81850**	2.06645	**0.9132**

TABLE IV
Comparison among *GDE3, MOEA/D, HypE, PICEA-g, MOEA/BB* for **WFG3**. Each results represents average of 20 independent runs. The best results are highlighted with boldface type.

	Algorithm	GD	IGD	SU	HYP
Obj10	GDE3	8.00788	0.10619	1.03575	0.60769
	MOEA/D	11.59076	0.14476	1.03620	0.466062
	HypE	0.11602	**0.00218**	1.07663	0.70545
	PICEA-g	3.83625	0.00701	**0.76851**	**0.89368**
	MOEA/BB	**0.04357**	0.00375	0.90428	0.29699
Obj13	GDE3	60.37440	0.80356	1.30543	0.526438
	MOEA/D	77.44776	1.17038	1.26608	0.26979
	HypE	42.03246	0.31342	1.15788	0.68235
	PICEA-g	36.14414	0.047449	**0.912524**	**0.80017**
	MOEA/BB	**0.86708**	**0.01668**	1.02440	0.42307
Obj19	GDE3	3761.86794	53.54205	1.76087	0.34329
	MOEA/D	4384.14314	67.32029	1.66155	0.09249
	HypE	2853.35970	22.21496	1.60481	0.49412
	PICEA-g	2704.887001	**3.117955**	**1.161841**	0.58260
	MOEA/BB	**1779.29983**	7.64127	1.35488	**0.63901**

TABLE V
Comparison among *GDE3, MOEA/D, HypE, PICEA-g, MOEA/BB* for **WFG4**. Each results represents average of 20 independent runs. The best results are highlighted with boldface type.

	Algorithm	GD	IGD	SU	HYP
Obj10	GDE3	0.60583	0.00478	0.8108	0.81497
	MOEA/D	0.70529	0.00311	1.12128	0.750278
	HypE	0.11602	0.00218	1.07663	0.70545
	PICEA-g	0.02389	0.00131	0.75533	**0.95914**
	MOEA/BB	**0.02328**	**0.00099**	**0.75212**	0.92196
Obj13	GDE3	1.43315	0.01118	1.09153	0.75113
	MOEA/D	1.43967	0.00487	1.40401	0.75050
	HypE	0.22521	0.00342	1.45375	0.488358
	PICEA-g	0.33409	**0.00123**	0.96767	**0.95868**
	MOEA/BB	**0.05899**	0.00220	**0.93928**	0.70602
Obj19	GDE3	11.81693	0.08406	1.60924	0.67848
	MOEA/D	5.87401	0.03930	1.79685	0.64263
	HypE	**0.50525**	0.01887	1.80740	0.23337
	PICEA-g	3.19733	**0.00148**	**1.24902**	**0.95204**
	MOEA/BB	1.96522	0.01415	1.51652	0.66492

Some key points can be observed from Fig 3 plot:

- PICEA-g is always amongst the consistently best performing algorithms of all the algorithms.
- MOEA/BB is better than PICEA-g and all others in all algorithms except WFG3 and WFG4. In WFG3 all

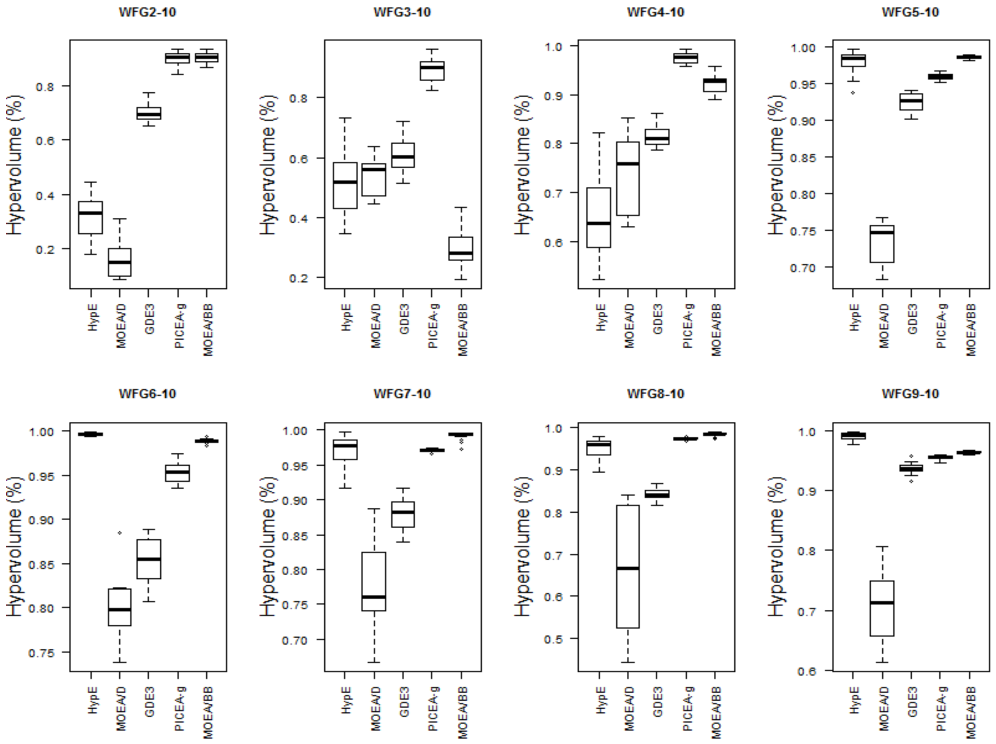

Fig. 3. Hypervolume performance metric comparison for WFG2-WFG9 in 10 objectives

TABLE VI
COMPARISON AMONG *GDE3, MOEA/D, HypE, PICEA-g, MOEA/BB* FOR **WFG5**. EACH RESULTS REPRESENTS AVERAGE OF 20 INDEPENDENT RUNS. THE BEST RESULTS ARE HIGHLIGHTED WITH BOLDFACE TYPE.

	Algorithm	GD	IGD	SU	HYP
Obj10	GDE3	0.06545	0.00134	**0.77466**	0.924715
	MOEA/D	0.10221	0.00255	1.15377	0.84278
	HypE	0.09281	0.00327	1.18936	0.67635
	PICEA-g	1.44936	0.00907	0.96048	0.86909
	MOEA/BB	**0.01460**	**0.00129**	0.79944	**0.98597**
Obj13	GDE3	0.11806	0.0018	1.02481	0.8672
	MOEA/D	0.19160	0.00279	1.49834	0.78071
	HypE	0.18259	0.00501	1.50987	0.33438
	PICEA-g	0.04543	0.00165	**0.93753**	0.91029
	MOEA/BB	**0.02491**	**0.00116**	1.03808	**0.91635**
Obj19	GDE3	0.34459	0.00284	1.47576	0.76512
	MOEA/D	0.54532	0.00379	2.03472	0.73057
	HypE	0.34879	0.00899	1.95595	0.19082
	PICEA-g	**0.13395**	**0.00187**	**1.25929**	0.72601
	MOEA/BB	0.21954	0.00256	1.62636	**0.78777**

TABLE VII
COMPARISON AMONG *GDE3, MOEA/D, HypE, PICEA-g, MOEA/BB* FOR **WFG6**. EACH RESULTS REPRESENTS AVERAGE OF 20 INDEPENDENT RUNS. THE BEST RESULTS ARE HIGHLIGHTED WITH BOLDFACE TYPE.

	Algorithm	GD	IGD	SU	HYP
Obj10	GDE3	0.04919	0.00181	0.84245	0.85440
	MOEA/D	0.05988	0.00254	1.14880	0.86263
	HypE	0.05468	0.00293	1.34552	0.32830
	PICEA-g	0.02407	0.00131	**0.73139**	0.95241
	MOEA/BB	**0.01686**	**0.00129**	0.86698	**0.98846**
Obj13	GDE3	0.07543	0.00225	1.11159	0.757525
	MOEA/D	0.09213	0.00269	1.48415	0.77306
	HypE	0.09281	0.00327	1.18936	0.67635
	PICEA-g	0.03020	0.00152	**0.93696**	0.93535
	MOEA/BB	**0.02226**	**0.00134**	1.14187	**0.95707**
Obj19	GDE3	0.22497	0.00394	1.59587	0.59778
	MOEA/D	0.27058	0.00325	2.00231	0.73495
	HypE	0.17662	0.00619	1.82611	0.12760
	PICEA-g	0.08071	**0.00172**	**1.2749**	0.81226
	MOEA/BB	**0.07069**	0.00185	1.72894	**0.85395**

algorithms outperforms it whereasin WFG4 only PICEA-g outperforms it.

- In all 10-objective problems except WFG3, MOEA/BB covers 90% of the globally optimal hypervolume.
- MOEA/BB has a very small standard deviation in comparison to PICEA-g in all problems it outperforming the other (PICEA-g).
- Judging by the plot, GDE3 is the 3rd best among the 5 algorithms being compared. It has better hypervolume than HypE and MOEA/D in WFG2, WFG4, WFG5, WFG7-WFG9.
- Both HypE and GDE3 algorithms are equally well. Each of them is better than the other in four problems. In all problems except WFG3 and WFG9, HypE is the worst algorithms among them. Therefore, with increasing objective, HypE performs worse than the others. This is due to the need of much more sampling points in order to approximate the hypervolume.

When the number of objectives increases to 13, we can observe that MOEA/BB and PICEA-g again perform consistently well in 13-objective problems. In all the problems except WFG9, PICEA-g and MOEA/BB is ranked either 1st or 2nd. GDE3 does quite good in WFG9 problem. In fact, other algorithms perform worse than the corresponding 10

TABLE VIII
COMPARISON AMONG *GDE3, MOEA/D, HypE, PICEA-g, MOEA/BB* FOR **WFG7**. EACH RESULTS REPRESENTS AVERAGE OF 20 INDEPENDENT RUNS. THE BEST RESULTS ARE HIGHLIGHTED WITH BOLDFACE TYPE.

	Algorithm	**GD**	**IGD**	**SU**	**HYP**
Obj10	GDE3	0.16138	0.00204	0.84921	0.87956
	MOEA/D	0.17670	0.00265	1.13763	0.82257
	HypE	0.12123	0.00286	1.14295	0.725438
	PICEA-g	0.02557	0.00147	0.82113	0.97098
	MOEA/BB	**0.02193**	**0.00124**	**0.74881**	**0.99188**
Obj13	GDE3	0.31206	0.00307	1.12076	0.828448
	MOEA/D	0.34467	0.00294	1.46486	0.75620
	HypE	0.20080	0.00372	1.48296	0.432298
	PICEA-g	0.06171	0.00177	**0.96635**	**0.96476**
	MOEA/BB	**0.03592**	**0.00143**	1.00958	0.94680
Obj19	GDE3	1.31333	0.01004	1.60094	0.74693
	MOEA/D	1.18019	0.00869	1.85692	0.59579
	HypE	**0.15767**	0.00404	1.79541	0.141968
	PICEA-g	0.39753	**0.00160**	**1.28362**	**0.95866**
	MOEA/BB	0.24799	0.00268	1.60236	0.78406

TABLE IX
COMPARISON AMONG *GDE3, MOEA/D, HypE, PICEA-g, MOEA/BB* FOR **WFG8**. EACH RESULTS REPRESENTS AVERAGE OF 20 INDEPENDENT RUNS. THE BEST RESULTS ARE HIGHLIGHTED WITH BOLDFACE TYPE.

	Algorithm	**GD**	**IGD**	**SU**	**HYP**
Obj10	GDE3	0.11020	0.00172	0.82926	0.84315
	MOEA/D	0.13136	0.00215	1.09462	0.77853
	HypE	0.04545	0.00213	1.06097	0.81903
	PICEA-g	0.02385	0.00126	**0.75795**	0.97432
	MOEA/BB	**0.01896**	**0.00104**	0.82057	**0.98515**
Obj13	GDE3	0.21165	0.00267	1.10237	0.76431
	MOEA/D	0.20269	0.00243	1.35819	0.75751
	HypE	0.06081	0.00232	1.39087	0.59963
	PICEA-g	0.04550	0.00152	**0.93390**	**0.96615**
	MOEA/BB	**0.02746**	**0.00122**	1.0626	0.94578
Obj19	GDE3	0.61467	0.00661	1.58778	0.64189
	MOEA/D	0.15536	0.00313	1.64494	0.50849
	HypE	**0.08911**	0.00326	1.81848	0.13724
	PICEA-g	0.17692	**0.00145**	**1.23305**	**0.94128**
	MOEA/BB	0.22662	0.00254	1.44729	0.65610

TABLE X
COMPARISON AMONG *GDE3, MOEA/D, HypE, PICEA-g, MOEA/BB* FOR **WFG9**. EACH RESULTS REPRESENTS AVERAGE OF 20 INDEPENDENT RUNS. THE BEST RESULTS ARE HIGHLIGHTED WITH BOLDFACE TYPE.

	Algorithm	**GD**	**IGD**	**SU**	**HYP**
Obj10	GDE3	0.05776	0.00168	**0.74933**	0.93629
	MOEA/D	0.08871	0.00245	1.14893	0.80084
	HypE	0.03420	0.00205	1.02134	0.89466
	PICEA-g	0.02872	0.00141	0.77512	0.95565
	MOEA/BB	**0.01186**	**0.00122**	0.91838	**0.9636**
Obj13	GDE3	0.10079	0.00194	0.97423	**0.88205**
	MOEA/D	0.15488	0.00270	1.48022	0.73682
	HypE	0.05472	0.00206	1.34446	0.85566
	PICEA-g	0.04224	0.00165	**0.96702**	0.86361
	MOEA/BB	**0.01498**	**0.00128**	1.19467	0.85109
Obj19	GDE3	0.40230	0.00332	1.35467	0.75359
	MOEA/D	0.60477	0.00486	1.97535	0.58450
	HypE	0.11399	0.00281	1.88884	0.62075
	PICEA-g	0.21274	0.00292	**1.26942**	0.64537
	MOEA/BB	**0.10726**	**0.00183**	1.61895	**0.89054**

objective instance. PICEA-g has better Space Uniformity than MOEA/BB in these problems. In all the problems, coverage of the algorithms is worse than its 10-objective counterpart. Still, convergence of MOEA/BB far excels other algorithms. In three problems, hypervolume values are better than PICA-g while it is four vice versa. In fact, their hypervolume is close to each other for all problems except in which MOEA/BB is worse than PICEA-g.

Fig 4 shows a boxplot of the Hypervolume results for 19 objective instances. MOEA/BB achieves better mean hypervolume than its nearest competitor PICEA-g in five out of eight 19-objective instances. Fig. 3 shows a boxplot of the Hypervolume results for 19 objective instances.

There are significant changes in the results for 19-objective problems. Upon closer examination:

- Even if at WFG4, WFG7 and WFG8 MOEA/BB is worse than PICEA-g, its improvement in WFG3 and WFG9 is notable. However, The other algorithms are still worse than MOEA/BB in those problems.
- Improvement of MOEA/BB came in the cost of convergence.
- In almost all problems except in WFG3 and WFG9, HypE covers less than 25% of the total hypervolume, which is considered a serious degradation.
- PICEA-g has better Space uniformity than other algorithms.

Considering both 10 objective and 19 objective, total 18 number WFG problem instances, MOEA/BB outperformed other algorithms 10 times, outperformed by other algorithms 5 times and was equally good in comparison to at least one algorithm 1 times. Out of those worst performing times, only one algorithm consistently outperformed MOEA/BB. In terms of overall performance, WFG2, WFG5, WFG6 and WFG9 are the only benchmark functions in which MOEA/BB is consistently better than any other algorithm considered in this study. On the other hand, its closest competitor PICEA-g does consistently well in WFG3, WFG4, WFG7 and WFG8. As we know from No Free Lunch theorem [90], no one algorithm can perform better than the other in all class of problems, the competency of both of the algorithms is understandable.

VI. CONCLUSION

In this paper, we proposed a new fuzzy based evolutionary approach to solving many-objective optimization problems. We utilized the power of fuzzy set for discriminating among the non-dominated populations and introduced crowding distance sampling mechanism for maintaining diversity. We applied our algorithm along with other existing algorithms on WFG benchmark problems, made a comparative study based on some well-established performance metrics in the MOEA literature and got competitive results.

Bias of a selection scheme largely affects performance of an algorithm. So far, there is no clear definition of selection bias. No comprehensive study has been made on how the bias influences a many-objective optimizer's performance. Some selection schemes can offer comparability but in a certain direction of the PF i.e. they selects solution in such a way that solutions lying in a certain direction get preference. Several such schemes can be combined and subsampled to get the entire front. Future research may be extended in this direction.

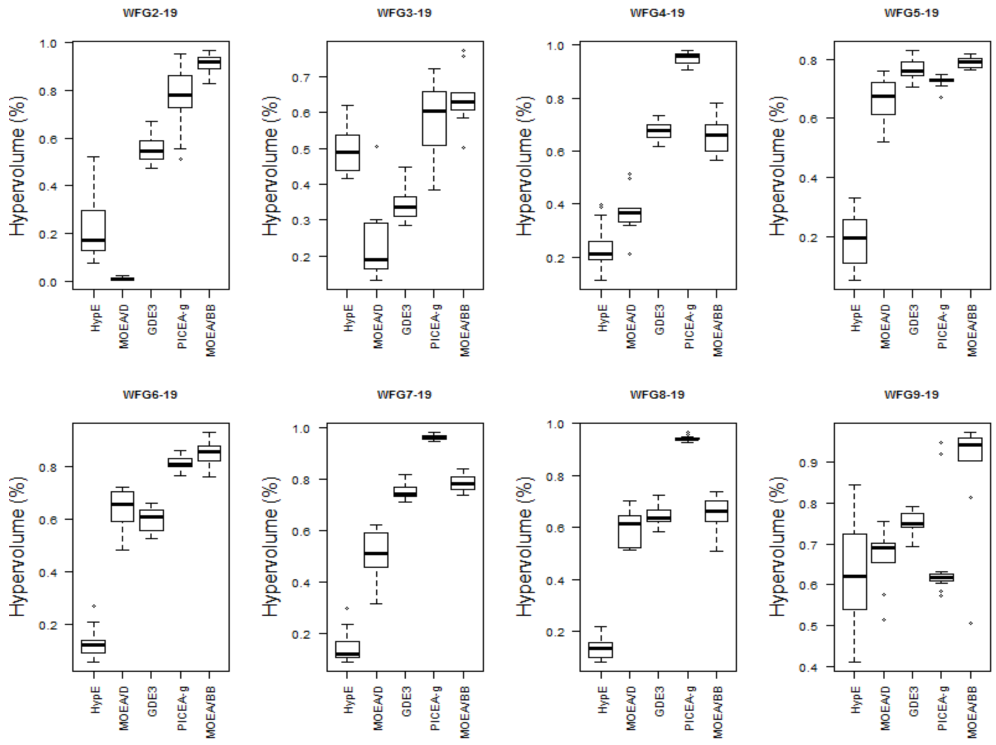

Fig. 4. Hypervolume performance metric comparison for WFG2-WFG9 in 19 objectives

References

[1] R. C. Purshouse and P. J. Fleming, "Conflict, harmony, and independence: Relationships in evolutionary multi-criterion optimisation," in *Evolutionary Multi-Criterion Optimization*. Springer, 2003, pp. 16–30.

[2] M. Farina and P. Amato, "On the optimal solution definition for many-criteria optimization problems," in *Proceedings of the NAFIPS-FLINT international conference*, 2002, pp. 233–238.

[3] S. Yang, M. Li, X. Liu, and J. Zheng, "A grid-based evolutionary algorithm for many-objective optimization." *IEEE Trans. Evolutionary Computation*, vol. 17, no. 5, pp. 721–736, 2013.

[4] F. di Pierro, "Many-objective evolutionary algorithms and applications to water resources engineering," Ph.D. dissertation, University of Exeter, 2006.

[5] J. R. Kasprzyk, P. M. Reed, B. R. Kirsch, and G. W. Characklis, "Managing population and drought risks using many-objective water portfolio planning under uncertainty," *Water Resources Research*, vol. 45, no. 12, 2009.

[6] C. A. Garrett, J. Huang, M. Goltz, and G. Lamont, "Parallel real-valued genetic algorithms for bioremediation optimization of tce-contaminated groundwater," in *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, vol. 3. IEEE, 1999.

[7] A. G. Pereira, G. Munda, and M. Pariccini, "Generating alternatives for siting retail and service facilities using genetic algorithms and multiple criteria devision techniques," *Journal of Retailing and Consumer Services*, vol. 1, no. 2, pp. 40–47, 1994.

[8] R. Zebulum, M. A. Pacheco, and M. Vellasco, "Synthesis of cmos operational amplifiers through genetic algorithms," in *Integrated Circuit Design, 1998. Proceedings. XI Brazilian Symposium on*. IEEE, 1998, pp. 125–128.

[9] K.-S. Tang, K.-F. Man, and K.-T. Ko, "Wireless lan design using hierarchical genetic algorithm." in *ICGA*, 1997, pp. 629–635.

[10] Y. Donoso Meisel *et al.*, *Multi-objective optimization scheme for static and dynamic multicast flows*. Universitat de Girona, 2005.

[11] T. Marcu, L. Ferariu, L. Mirea, and P. Frank, "Genetic evolving of dynamic neural networks with application to process fault diagnosis," in *Proceedings of the European Control Conference, Karlsruhe, Germany*, vol. 31, 1999.

[12] P. Schroder, A. Chipperfield, P. Fleming, and N. Grum, "Multi-objective optimisation of distributed active magnetic bearing controllers," 1997.

[13] D. C. Donha, D. Desanj, and M. Katebi, "Genetic algorithm for weight selection in h control design," in *Industrial Applications of Intelligent Control (Digest No: 1997/144), IEE Colloquium on*. IET, 1997, pp. 5–1.

[14] A. Chipperfield and P. Fleming, "Multiobjective gas turbine engine controller design using genetic algorithms," *Industrial Electronics, IEEE Transactions on*, vol. 43, no. 5, pp. 583–587, 1996.

[15] J. Whidborne, D.-W. Gu, and I. Postlethwaite, "Algorithms for the method of inequalities-a comparative study," in *American Control Conference, Proceedings of the 1995*, vol. 5. IEEE, 1995, pp. 3393–3397.

[16] K. C. Tan and Y. Li, "Multi-objective genetic algorithm based time and frequency domain design unification of linear control systems," in *Proceedings of the Artificial Intelligence in Real-Time Control*, 1997, pp. 35–40.

[17] N. Duarte, A. Ruano, C. Fonseca, and P. Fleming, "Accelerating multi-objective control system design using a neuro-genetic approach," 2000.

[18] N. V. Dakev, J. Whidborne, and A. J. Chipperfield, "H design of an ems control system for a maglev vehicle using evolutionary algorithms," 1995.

[19] M. Ortmann and W. Weber, "Multi-criterion optimization of robot trajectories with evolutionary strategies," in *Proceedings of the 2001 Genetic and Evolutionary Computation Conference. Late-Breaking Papers*, 2001, pp. 310–316.

[20] C. A. Coello Coello, A. D. Christiansen, and A. H. Aguirre, "Using a new ga-based multiobjective optimization technique for the design of robot arms," *Robotica*, vol. 16, no. 04, pp. 401–414, 1998.

[21] N. Laumanns, M. Laumanns, and D. Neunzig, "Multi-objective design space exploration of road trains with evolutionary algorithms," in *Evolutionary Multi-Criterion Optimization*. Springer, 2001, pp. 612–623.

[22] M. Qiu, "Prioritising and scheduling road projects by genetic algorithm," *Mathematics and Computers in Simulation*, vol. 43, no. 3, pp. 569–574, 1997.

[23] C. Jarvis, N. Stuart, K. Kelsey, and R. Baker, "Towards a methodology for selecting a characteristic sample from an existing database: An evolutionary approach," in *Third International Conference/Workshop on Integrating GIS and Environmental Modeling*, 1996, pp. 21–26.

[24] J. H. Reynolds, "Multi-criteria assessment of ecological process models using pareto optimization," Ph.D. dissertation, 1997.

[25] R. Groppetti and R. Muscia, "On a genetic multiobjective approach for the integration and optimization of assembly product design and process planning," in *Integrated Design and Manufacturing in Mechanical Engineering*. Springer, 1997, pp. 61–70.

[26] B. Rekiek and A. Delchambre, *Assembly line design: the balancing of mixed-model hybrid assembly lines with genetic algorithms*. Springer, 2006.

[27] B. Rekiek, P. De Lit, F. Pellichero, T. LEglise, E. Falkenauer, and A. Delchambre, "Dealing with users preferences in hybrid assembly lines design," in *Proceedings of the MCPL2000 Conference*, 2000.

[28] J.-H. Chen and S.-Y. Ho, "Multi-objective evolutionary optimization of flexible manufacturing systems."

[29] B. Paechter, R. Rankin, A. Cumming, and T. C. Fogarty, "Timetabling the classes of an entire university with an evolutionary algorithm," in *Parallel Problem Solving from NaturePPSN V*. Springer, 1998, pp. 865–874.

[30] P. Poulos, G. Rigatos, S. Tzafestas, and A. Koukos, "A pareto-optimal genetic algorithm for warehouse multi-objective optimization," *Engineering Applications of Artificial Intelligence*, vol. 14, no. 6, pp. 737–749, 2001.

[31] R. Bellman, *Dynamic Programming*, ser. Dover Books on Computer Science Series. Dover Publications, 2003. [Online]. Available: http://books.google.com.bd/books?id=fyVtp3EMxasC

[32] K. Miettinen, *Nonlinear multiobjective optimization*. Springer, 1999, vol. 12.

[33] A. Inselberg and B. Dimsdale, *Parallel coordinates for visualizing multi-dimensional geometry*. Springer, 1987.

[34] ——, "Parallel coordinates: a tool for visualizing multi-dimensional geometry," in *Visualization, 1990. Visualization '90., Proceedings of the First IEEE Conference on*, Oct 1990, pp. 361–378.

[35] C. A. C. Coello, "A comprehensive survey of evolutionary-based multiobjective optimization techniques," *Knowledge and Information systems*, vol. 1, no. 3, pp. 269–308, 1999.

[36] C. M. Fonseca, P. J. Fleming *et al.*, "Genetic algorithms for multiobjective optimization: Formulationdiscussion and generalization." in *ICGA*, vol. 93, 1993, pp. 416–423.

[37] N. Srinivas and K. Deb, "Muiltiobjective optimization using nondominated sorting in genetic algorithms," *Evolutionary computation*, vol. 2, no. 3, pp. 221–248, 1994.

[38] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *Evolutionary Computation, IEEE Transactions on*, vol. 6, no. 2, pp. 182–197, 2002.

[39] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach," *Evolutionary Computation, IEEE Transactions on*, vol. 3, no. 4, pp. 257–271, 1999.

[40] E. Zitzler, M. Laumanns, L. Thiele, E. Zitzler, E. Zitzler, L. Thiele, and L. Thiele, "Spea2: Improving the strength pareto evolutionary algorithm," 2001.

[41] R. C. Purshouse and P. J. Fleming, "Evolutionary many-objective optimisation: An exploratory analysis," in *Evolutionary Computation, 2003. CEC'03. The 2003 Congress on*, vol. 3. IEEE, 2003, pp. 2066–2073.

[42] E. J. Hughes, "Evolutionary many-objective optimisation: many once or one many?" in *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, vol. 1. IEEE, 2005, pp. 222–227.

[43] T. Wagner, N. Beume, and B. Naujoks, "Pareto-, aggregation-, and indicator-based methods in many-objective optimization," in *Evolutionary multi-criterion optimization*. Springer, 2007, pp. 742–756.

[44] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: A short review." in *IEEE Congress on Evolutionary Computation*, 2008, pp. 2419–2426.

[45] H. Ishibuchi, N. Tsukamoto, Y. Hitotsuyanagi, and Y. Nojima, "Effectiveness of scalability improvement attempts on the performance of nsga-ii for many-objective problems," in *Proceedings of the 10th annual conference on Genetic and evolutionary computation*. ACM, 2008, pp. 649–656.

[46] M. Farina and P. Amato, "Fuzzy optimality and evolutionary multiobjective optimization," in *Evolutionary Multi-Criterion Optimization*. Springer, 2003, pp. 58–72.

[47] J. Knowles and D. Corne, "The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation," in *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, vol. 1. IEEE, 1999.

[48] T. Hanne, "On the convergence of multiobjective evolutionary algorithms," *European Journal of Operational Research*, vol. 117, no. 3, pp. 553–564, 1999.

[49] L. A. Zadeh, "Fuzzy sets," *Information and control*, vol. 8, no. 3, pp. 338–353, 1965.

[50] H. Sato, H. E. Aguirre, and K. Tanaka, "Controlling dominance area of solutions and its impact on the performance of moeas," in *Evolutionary Multi-Criterion Optimization*. Springer, 2007, pp. 5–20.

[51] M. Köppen, R. Vicente-Garcia, and B. Nickolay, "Fuzzy-pareto-dominance and its application in evolutionary multi-objective optimization," in *Evolutionary Multi-Criterion Optimization*. Springer, 2005, pp. 399–412.

[52] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler, "Combining convergence and diversity in evolutionary multiobjective optimization," *Evolutionary computation*, vol. 10, no. 3, pp. 263–282, 2002.

[53] L. S. Batista, F. Campelo, F. G. Guimaraes, and J. A. Ramírez, "A comparison of dominance criteria in many-objective optimization problems," in *Evolutionary Computation (CEC), 2011 IEEE Congress on*. IEEE, 2011, pp. 2359–2366.

[54] N. Drechsler, R. Drechsler, and B. Becker, "Multi-objective optimisation based on relation favour," in *Evolutionary multi-criterion optimization*. Springer, 2001, pp. 154–166.

[55] A. Sülflow, N. Drechsler, and R. Drechsler, "Robust multi-objective optimization in high dimensional spaces," in *Evolutionary multi-criterion optimization*. Springer, 2007, pp. 715–726.

[56] S. Kukkonen and J. Lampinen, "Ranking-dominance and many-objective optimization," in *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*. IEEE, 2007, pp. 3983–3990.

[57] K. Maneeratana, K. Boonlong, and N. Chaiyaratana, "Compressed-objective genetic algorithm," in *Parallel Problem Solving from Nature-PPSN IX*. Springer, 2006, pp. 473–482.

[58] D. W. Corne and J. D. Knowles, "Techniques for highly multiobjectiveoptimisation: some nondominated points are better than others," in *Proceedings of the 9th annual conference on Genetic and evolutionary computation*. ACM, 2007, pp. 773–780.

[59] J. Knowles and D. Corne, "On metrics for comparing nondominated sets," in *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on*, vol. 1. IEEE, 2002, pp. 711–716.

[60] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Da Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *Evolutionary Computation, IEEE Transactions on*, vol. 7, no. 2, pp. 117–132, 2003.

[61] T. Okabe, Y. Jin, and B. Sendhoff, "A critical survey of performance indices for multi-objective optimisation," in *Evolutionary Computation, 2003. CEC'03. The 2003 Congress on*, vol. 2. IEEE, 2003, pp. 878–885.

[62] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithmsa comparative case study," in *Parallel problem solving from naturePPSN V*. Springer, 1998, pp. 292–301.

[63] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *Parallel Problem Solving from Nature-PPSN VIII*. Springer, 2004, pp. 832–842.

[64] J. Bader and E. Zitzler, "Hype: An algorithm for fast hypervolume-based many-objective optimization," *Evolutionary Computation*, vol. 19, no. 1, pp. 45–76, 2011.

[65] L. While, L. Bradstreet, and L. Barone, "A fast way of calculating exact hypervolumes," *Evolutionary Computation, IEEE Transactions on*, vol. 16, no. 1, pp. 86–95, 2012.

[66] K. Deb and D. K. Saxena, "On finding pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems," *Kangal report*, vol. 2005011, 2005.

[67] D. Brockhoff and E. Zitzler, "Objective reduction in evolutionary multiobjective optimization: Theory and applications," *Evolutionary Computation*, vol. 17, no. 2, pp. 135–166, 2009.

[68] ——, "Are all objectives necessary? on dimensionality reduction in evolutionary multiobjective optimization," in *Parallel Problem Solving from Nature-PPSN IX*. Springer, 2006, pp. 533–542.

[69] E. J. Hughes, "Multiple single objective pareto sampling," in *Evolutionary Computation, 2003. CEC'03. The 2003 Congress on*, vol. 4. IEEE, 2003, pp. 2678–2684.

[70] ——, "Msops-ii: A general-purpose many-objective optimiser," in *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*. IEEE, 2007, pp. 3944–3951.

[71] Q. Zhang and H. Li, "Moea/d: A multiobjective evolutionary algorithm based on decomposition," *Evolutionary Computation, IEEE Transactions on*, vol. 11, no. 6, pp. 712–731, 2007.

[72] P. J. Fleming, R. C. Purshouse, and R. J. Lygoe, "Many-objective optimization: An engineering design perspective," in *Evolutionary multi-criterion optimization*. Springer, 2005, pp. 14–32.

[73] K. Deb, J. Sundar, N. UdayaBhaskara Rao, and S. Chaudhuri, "Reference point based multi-objective optimization using evolutionary algorithms," *International Journal of Computational Intelligence Research*, vol. 2, no. 3, pp. 273–286, 2006.

[74] L. Thiele, K. Miettinen, P. J. Korhonen, and J. Molina, "A preference-based interactive evolutionary algorithm for multiobjective optimization," 2007.

[75] R. Wang, R. C. Purshouse, and P. J. Fleming, "Preference-inspired co-evolutionary algorithms for many-objective optimization," *Evolutionary Computation, IEEE Transactions on*, vol. 17, no. 4, pp. 474–494, 2013.

[76] M. P. Kleeman and G. B. Lamont, "Coevolutionary multi-objective eas: The next frontier?" in *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*. IEEE, 2006, pp. 1726–1735.

[77] R. C. Purshouse, C. Jalbă, and P. J. Fleming, "Preference-driven co-evolutionary algorithms show promise for many-objective optimisation," in *Evolutionary Multi-Criterion Optimization*. Springer, 2011, pp. 136–150.

[78] M. Farina and P. Amato, "A fuzzy definition of" optimality" for many-criteria optimization problems," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 34, no. 3, pp. 315–326, 2004.

[79] M. Nasir, A. K. Mondal, S. Sengupta, S. Das, and A. Abraham, "An improved multiobjective evolutionary algorithm based on decomposition with fuzzy dominance," in *Evolutionary Computation (CEC), 2011 IEEE Congress on*, June 2011, pp. 765–772.

[80] M. Köppen, K. Franke, and B. Nickolay, "Fuzzy-pareto-dominance driven multi-objective genetic algorithm," in *Proc. 10th International Fuzzy Systems Assotiation World Congress (IFSA), Istanbul, Turkey*, vol. 2003, 2003, pp. 450–453.

[81] Z. He and G. G. Yen, "A new fitness evaluation method based on fuzzy logic in multiobjective evolutionary algorithms," in *Evolutionary Computation (CEC), 2012 IEEE Congress on*. IEEE, 2012, pp. 1–8.

[82] Z. He, G. G. Yen, and J. Zhang, "Fuzzy-based pareto optimality for many-objective evolutionary algorithms," *Evolutionary Computation, IEEE Transactions on*, vol. 18, no. 2, pp. 269–285, 2014.

[83] L. Rachmawati and D. Srinivasan, "Incorporation of imprecise goal vectors into evolutionary multi-objective optimization," in *Evolutionary Computation (CEC), 2010 IEEE Congress on*. IEEE, 2010, pp. 1–8.

[84] H. Zimmermann, *Fuzzy Set Theory and Its Applications Second, Revised Edition*. Springer, 1992.

[85] D. E. Knuth, *The Art of Computer Programming, Volume 2 (3rd Ed.): Seminumerical Algorithms*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1997.

[86] S. Kukkonen and J. Lampinen, "Gde3: The third evolution step of generalized differential evolution," in *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, vol. 1. IEEE, 2005, pp. 443–450.

[87] J. Lampinen *et al.*, "Des selection rule for multiobjective optimization."

[88] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *Evolutionary Computation, IEEE Transactions on*, vol. 10, no. 5, pp. 477–506, 2006.

[89] C. A. C. Coello, D. A. Van Veldhuizen, and G. B. Lamont, *Evolutionary algorithms for solving multi-objective problems*, vol. 242.

[90] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *Evolutionary Computation, IEEE Transactions on*, vol. 1, no. 1, pp. 67–82, 1997.