Appl Intell (2016) 45:953–975
DOI 10.1007/s10489-016-0800-4

A diversity control mechanism in many objective optimizations

Kimia Bazargan Lari[1] · Ali Hamzeh[1]

Published online: 18 June 2016

Abstract Multi-objective evolutionary optimization algorithms are among the best optimizers for solving problems in control systems, engineering and industrial planning. The performance of these algorithms degrades severely due to the loss of selection pressure exerted by the Pareto dominance relation which will cause the algorithm to act randomly. Various recent methods tried to provide more selection pressure but this would cause the population to converge to a specific region which is not desirable. Diversity reduction in high dimensional problems which decreases the capabilities of these approaches is a decisive factor in the overall performance of these algorithms. The novelty of this paper is to propose a new diversity measure and a diversity control mechanism which can be used in combination to remedy the mentioned problem. This measure is based on shortest Hamiltonian path for capturing an order of the population in any dimension. In order to control the diversity of population, we designed an adaptive framework which adjusts the selection operator according to diversity variation in the population using different diversity measures as well as our proposed one. This study incorporates the proposed framework in MOEA/D, an efficient widely used evolutionary algorithm. The obtained results validate the motivation on the basis of diversity and performance measures in comparison with the state-of-the-art algorithms and demonstrate the applicability of our algorithm/method in handling many-objective problems. Moreover, an extensive comparison with several diversity measure algorithms reveals the competitiveness of our proposed measure.

Keywords Evolutionary algorithm · Many objective Evolutionary algorithms · Population diversity · Optimization

1 Introduction

Most of engineering and design problems include optimizing a number of objectives, simultaneously [1] known as Multi-Objective Optimization Problems (MOPs) which continue to gain increasing attention among researchers to find efficient methods of handling them. A Multi-Objective Optimization Problem (MOP) can be stated as follows:

$$\begin{aligned} minimize\,\{(f_1\,(\mathbf{x})\,.\,f_2\,(\mathbf{x})\,.\,.\,.\,.\,.\,f_k\,(\mathbf{x}))\} \\ subject\ to\ x\ \in\ S \end{aligned} \tag{1}$$

where f (x) is a vector of k objective functions which are to be optimized and x is a vector of decision variables [2]. Here and in the rest of this paper, without loss of generality, we assume MOPs to be minimization problem. In such problems, there exists no single solution to optimize all objectives. Hence, it is desirable to find a set of incomparable solutions which would be considered as the optimal solutions to the MOP [1, 9]. Many researches have been conducted in solving MOPs ever since their introduction. Recently, a considerable number of studies have been devoted to applying Evolutionary Algorithms (EAs) for solving such problems. Evolutionary Algorithms are suitable for solving MOPs due to their parallel search capability and impressive performance in unknown environments [2–5]. Accordingly, one of the well-known problem solvers for MOP are Multi-Objective Evolutionary Algorithms (MOEAs). The main reason of their popularity is

✉ Ali Hamzeh
ali@cse.shirazu.ac.ir

[1] CSE and IT Department, Shiraz University, Shiraz, Iran

their ability to find multiple trade-off solutions in a single run. As it is said, finding a set of solutions as optimal is desirable in solving an MOP. These solutions are called non-dominant solutions and the set containing all these solutions is called Pareto front (non-dominated vectors are in the objective space). Converging to the true Pareto-front and maximizing the population diversity of non-dominated solutions in the objective space are the two main targets of all MOEA approaches. The formal definition of Pareto domination is: a vector u_2 Pareto dominates another vector u_1 if and only if for all objectives we have [1]:

$$f_i\left(u_1\right) \geq f_i\left(u_2\right) i = 12. \ldots .m \tag{2}$$

and

$$\exists i \epsilon \left\{1.2. \ldots .m\right\} \Rightarrow f_i\left(u_1\right) \succ f_i\left(u_2\right) \tag{3}$$

Research in the field of Multi-Objective Evolutionary Algorithm(MOEAs) has been very active in the last 10 years and many potential algorithms with promising results have been proposed [6, 7], and [8]. Although the existing MOEAs show satisfying performance for MOPs with 2 or 3 objectives, their search ability for problems with more than three objectives often degrades. These problems are referred as Many-objective Optimization Problem (MaOP) [6]. This performance degeneration is due to the loss of selection pressure exerted by the Pareto dominance relation which will cause premature convergence of the algorithms [6–9]. Furthermore, approximating the Pareto front will become difficult, since we need more points to represent a larger space [2]. In high dimensional problems, Pareto dominance relation loses its discrimination capability. In other words, nearly all solutions in the population would become non-dominated and algorithms which use dominance relation directly would lose their selection pressure and accordingly would not converge.

The search effort in MOEAs can be divided into two main parts: converging to the Pareto Front and diversifying the population (representing all regions of the Pareto front). In many objective problems, the regular MOEAs would fail to discriminate between solutions according to their fitness so only the diversity part of their selection would work [6, 7]. Increasing selection pressure by modifying the dominance relation would cause the population to converge to a specific region while paying more attention to diversity would prevent the algorithm from convergence. In this way, convergence and diversity are two contradicting objectives of an MOEA.

In this regard, this paper attempts to adaptively control the population diversity among chromosomes to overcome this dilemma. In the proposed method, a diversity control mechanism is introduced along with a diversity measure which can be used in high dimensional spaces. In this approach, the selection of chromosomes is based on their participation in the preservation of diversity in addition to their fitness and we would adaptively change the importance of diversity in each generation. The proposed diversity metric is based on finding shortest Hamiltonian path. The experimental results confirmed that the proposed mechanism is capable of creating a set of well-distributed and optimal solutions which is competitive with the state-of-the-art methods.

Our contribution in this paper is two folded. First we have proposed a diversity control mechanism which is able to adaptively control the population diversity which consequently controls the amount of selection pressure and thus helps the algorithm to produce a diverse set of solutions close to real Pareto Front. Second, we designed a diversity measure to be used in high dimensional problems. In view of the fact that the proposed diversity control mechanism requires a diversity measure to assess the diversity of each generation and also other known diversity measures have some drawbacks in high dimensions, a new diversity measure is proposed. This diversity measure is based on finding a shortest Hamiltonian path in the complete distance graph of solutions in the objective space. Using this concept, we can capture a proper order in high dimensional spaces which do not have any intrinsic order.

In our experiments, we obtained contradictory results with respect to two well-known performance indicators: Hypervolume and IGD (Inverted Generational Distance); which was not odd according to similar experimental and theoretical studies [37]. In the next step, using known methodologies and arguments [20], we compared the two metrics and revealed that IGD can give a more accurate comparison between various methods. Since our method has obtained relatively smaller values of IGD we can conclude that the proposed framework along with the diversity measure have potential in optimizing many objective problems.

The rest of this paper is organized as follows: Section 2, explains the background and related work. Proposed method is described in Section 3. Section 4 shows the experimental results and analysis. Finally, Section 5 includes conclusion and future work in regard to the proposed method.

2 Related work

In this following section, we introduce the background and the concept of Multi and Many Objective Evolutionary Algorithms.

Many objective problems have been in the frontier of multi-objective optimization research for more than a decade. Since the main issue that hinders the evolution of algorithms is the loss of discrimination ability of the Pareto dominance operator, new algorithms are either using a modified version of this operator or using it in an indirect way

[6]. Using modified versions of dominance operator can be useful for 4-6 objectives but it would fail in higher dimensional problems [6]. There are a variety of evolutionary meta-heuristic algorithms for handling many objective optimization problems. Many studies have been focused on improving the performance of these methods in solving problems [7–9, 16, 17]; here we will explore their offered approach.

By increasing the number of objective functions, in an optimization problem, the amount of non-dominated solutions will increase rapidly, leading to severe loss of pressure in the selection mechanism [6]. In order to address this problem, among many other methods, two classes of algorithms have shown great potential in solving many objective optimization problems: the decomposition based methods and indicator based methods. Decomposition based methods use a set of pre-specified uniformly distributed weight vectors, each of which constructs a single objective optimization problem that ideally would find a point of the true Pareto front. Collectively these sub-problems will find an acceptable representation of the Pareto front [11]. The pioneer method in this family is Multi-Objective Evolutionary Algorithm by Decomposition (MOEA/D) [11]. In order to search effectively, a neighborhood is assigned for each sub-problem which includes other problems that are near to it in the weight space. Mating is restricted within the neighborhoods so the variation operators would not be disruptive. In designing a decomposition based algorithm, one has to choose how to generate the weight vectors. If there is preference information available, it can be incorporated in the distribution of weight vectors. But, in most cases, the weights are generated uniformly using the procedure proposed by Das & Dennis [12]. Another design choice is the scalarization function which is used to combine the objective values and construct the sub-problems. Many different functions have been proposed in the literature [11], but in this paper we used the Tchebichev function.

Various attempts are provided to improve the performance of the MOEA/D. First the SBX was considered as its genetic operator but due to the poor performance of this operator for maintaining population diversity, Zhang et.al introduced MOEA/D-DE using DE operator to perform more efficient search [13]. In order to avoid the problem of high computational complexity, Zhang et al. introduced the idea of dynamic resource allocation in the form of MOEA/D-DRA [14]. This algorithm introduces a kind of MOEA/D that uses a utility function for assigning computational resources to each sub-problem and on the basis of this assignment, the computational load is distributed among all of them. In another work, Zhang et.al introduced MOEA/D-GM which presents two mechanisms to improve MOEA/D. First, in order to take full advantage of the neighborhood information of a sub-problem, it applies targeted mutation operator that uses both global and local information. Then, an update mechanism will apply to improve the algorithm, when there is no uniform distribution of sub problems. This update mechanism is implemented using a priority queue. In 2011, Gong et.al proposed iMOEA/D, this algorithm has an interaction with decision maker and guides the search in the direction of preferred regions in order to enhance the distribution of Pareto front and improve the search mechanism of MOEA/D especially in dealing with many objective optimization problems. Sato proposed a method named inverted PBI scalarization in which an effective scalarization function is used to produce approximation sets which are widely distributed in the objective space [15].

Another Many Objective algorithm that uses decomposition is NSGA-III [16]. NSGA-III is not technically a decomposition based MOEA but it borrows ideas from these algorithms. In order to overcome the loss of diversity in the population, NSGA-III uses a set of predefined points in the objective space which are spread on a hyper-plane and called reference points. All the solutions in the objective space are projected onto this hyper-plane and each one is assigned to the nearest reference point on the plane. For each solution the crowdedness of its corresponding reference point is an indicator of its contribution to the overall diversity. Yuan et al. proposed an improved NSGA-III named Θ-NSGA-III that inherits specific characteristics of NSGA-III including adaptive normalization and diversity maintenance which with the aid of reference points will lead to a well distributed Pareto front in the objective space [17]. However, Θ-NSGA-III differs from NSGA-III in various certain parts. It applies a new dominance relation, Θ-dominance, which assigns solutions to different clusters. These clusters are formed by reference points, which are well distributed throughout the space. Just the solutions that are found in similar clusters are participating in competitive relation. In this competitive relation a penalty based boundary intersection is described and the solutions will perform based on it. During environmental selection, the θ-dominance not only keeps the solutions with better fitness in each cluster but also ensures that the solutions are evenly distributed among all clusters [17].

Another class of algorithms for handling many objective optimization problems includes Indicator based methods. Indicator based algorithms are inspired by the idea of maximizing a performance indicator which would yield a good result in terms of that indicator. The main problem with these algorithms is the computational complexity of indicator measures. Hypervolume is the most popular indicator used in this context. In the following paragraphs we will describe two of the most popular indicator based algorithms briefly.

S-Metric Selection Evolutionary Multi-Objective Algorithm (SMS-EMOA) [18] is a steady-state evolutionary algorithm which uses hypervolume. S-Metric is another name for hypervolume which was more common in the early years of this metric's introduction [18]. Since computing hypervolume is a computationally intense task, SMS-EMOA uses a steady-state population model to reduce the number of hypervolume computations. So in each generation, one new solution is created and added to the population and consequently one solution should be deleted. In each generation, in order to remove one solution, all the individuals in the first non-dominated front are sorted according to their hypervolume contribution. The hypervolume contribution of an individual is defined as the difference between the hypervolume of the population with and without that individual.

Since the main shortcoming of indicator based algorithms is their computational cost, Bader and Zitzler proposed a method that tries to estimate the value of the hypervolume. They designed Hypervolume Estimation algorithm (HypE), which uses Monte Carlo simulation for estimating the hypervolume contributions in an efficient way [19]. Monte Carlo simulation is a well-studied method for approximation of definite integrals and since hypervolume is a kind of integral, good estimations are expected from exploitation of this method.

3 Proposed method

As stated before, many objective optimization problems have introduced new challenges to MOEA community. In high dimensional problems, nearly all solutions in the population would become non-dominated which is caused by loss of selection pressure and accordingly the algorithm would not converge to Pareto front properly. Due to these challenges, finding reasonable approximation of diverse optimal Pareto-sets is so hard and requires a more delicate balance between convergence and diversity [17]. In this regard, common approaches in evolutionary multi-objective optimization either fails to provide enough selection pressure or the selection pressure is too much which causes concentration of the population to a specific region of space [6, 7], and [17]. In order to overcome this dilemma, one obvious direction for improving MOEAs is to add more effective diversity control mechanisms to algorithms with acceptable selection pressure. With the aim of preserving population diversity, this paper proposed a diversity control mechanism and also a new diversity measure which can be used in combination. In order to control the population diversity, we designed an adaptive framework which adjusts the selection operator according to diversity variation in the population. By an increase in the diversity of population the selection operator will change to increase the selection pressure and by a decrease in diversity the selection pressure will decline. Additionally, a new diversity measure which is based on shortest Hamiltonian path is proposed in order to monitor the variation of population diversity during each generation.

In the rest of this section, our contribution is presented in three parts. The first part introduces the diversity maintenance mechanism. After that, in the second part the novel diversity measure will be described and its details will be expounded and finally refinements needed for incorporation of diversity maintenance mechanism in MOEA/D will be explained.

3.1 Diversity maintenance mechanism

Here in this section, we aim to propose a diversity control mechanism to streamline the optimization process. As discussed in many other works, lack of discriminability in dominance relation propels optimizers to modify it so as to increase the selection pressure. Intensifying the selection pressure usually leads to loss of diversity in the population which is not desirable [10]. By using a diversity measure, we can detect any variation in the population diversity at each generation and make decisions based on that. After we identified that the population diversity is decreased we should regulate the selection mechanism in a way to compensate for this reduction. In order to incorporate diversity into the selection mechanism of algorithms in a more controlled way, this paper proposed a diversity maintenance framework. This framework is concisely illustrated in the Flowchart 1 where it uses the diversity measure that will describe in the next subsection. In most of common MOEAs, solutions are selected according to their fitness and diversity is used as a tie breaker [1]. We propose to alter this order adaptively. For instance, there may be situations where the population is converging toward a specific region of the Pareto Front or even a single point. In such a situation, the selection operator should pay more attention to the population diversity than fitness.

There are two important issues in our proposed framework. First, we need to know when to switch the order of selection. Second, we need to determine which chromosome has the highest contribution in the diversity of the population so we give it an advantage in the selection mechanism. For the first concern, we measure the diversity of the population in each generation using accepted diversity measure, and by comparing to the one for the previous generation we can determine whether the population has got or lost diversity. Accordingly, we can set the order in the selection operator, i.e. "Fitness first" or "Diversity first". For each chromosome we should be able to compute its contribution in the population's overall diversity which can be done in different ways. This contribution in the diversity would be used as the second measure of selection for each solution.

In the case of “Fitness first”, the selection mechanism will operate regularly, i.e. mainly based on the fitness of individuals and based on the diversity contribution in case of indifference. But for the “Diversity first”, we first select based on the solution’s diversity contribution and the fitness would be used breaking ties. You must note that by fitness we mean the scalar value assigned to each solution based on its objective values, like the domination rank or in NSGA2 or the scalarized value in MOEAD or distance to reference point in NSGAIII.

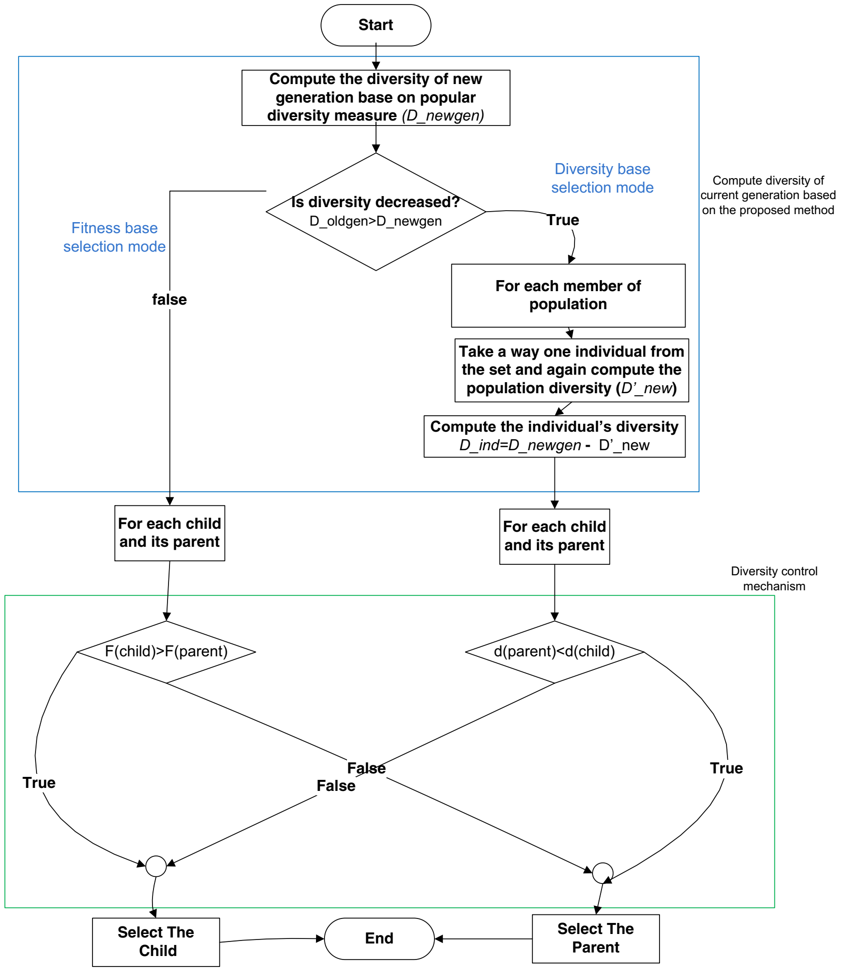
In the next sub sections, we introduce some popular diversity metrics and after exploring their inefficiencies we will propose a novel diversity measure.

3.2 Planning for diversity measure

In the previous sub section, we proposed an adaptive framework which adjusts the selection operator according to the diversity variation in the population. This diversity control mechanism requires to find the population diversity. As it can be seen, the main part of this framework is monitoring and computing the population diversity. In this section, first, we introduce some popular diversity metrics and review their features to find out whether they can be exploited in our framework or not.

Toward understanding the motivation of our proposed diversity metric, some quantitative metrics for multi and many objective optimization are examined. In particular, the quantitative metrics of multi-objective optimization realm are commonly categorized in to four groups [sweing]:

1. *Capacity metrics:* These metrics quantify the number (ratio) of non-dominated solutions in population that satisfies given predefined requirements. In general, a large number of non-dominated solutions in the population is preferred. (e.g. Error Ratio (ER) [21], Ratio of the Reference Points Found in [22, 23], non-dominated Points by Reference Set (C2R) [23], etc.)
2. *Convergence metrics*: These are metrics for measuring the degree of proximity of approximation set S based on the distance between the solutions in S to those in the Pareto Front. Convergence metrics measure proximity. (e.g. Generational Distance (GD) [24, 25], ϵ-indicator [26], ...)


Start
Compute the diversity of new generation base on popular diversity measure (D_newgen)
Is diversity decreased?
D_oldgen>D_newgen
Diversity base selection mode
Compute diversity of current generation based on the proposed method
Fitness base selection mode
True
false
For each member of population
Take a way one individual from the set and again compute the population diversity (D’_new)
Compute the individual’s diversity
D_ind=D_newgen - D’_new
For each child and its parent
For each child and its parent
Diversity control mechanism
F(child)>F(parent)
d(parent)<d(child)
True
False
False
True
Select The Child
End
Select The Parent

Flowchart 1 Flowchart of the proposed diversity maintenance mechanism

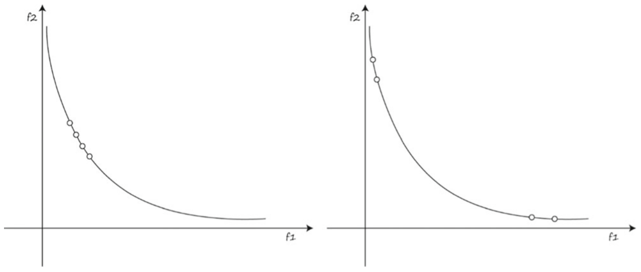
3. *Diversity metrics:* These metrics incorporate two notions: Distribution and spread. Distribution focuses on how evenly the solutions of the population are scattered in the objective space. The Spread quantifies how much of the extreme regions of the Pareto front are covered by the population
4. *Convergence–Diversity metrics:* These metrics are indicating both the convergence and the diversity of solution set S in a single value. (e.g. in [27, 28], Zitzler et al. proposed the popular performance metric Hypervolume (HV) which gives the volume of optimal solution set S in the objective space and the metric Inverted Generational Distance (IGD) in [29] which compute the average distance of the true Pareto front from the obtained optimal solution set S).

In this paper we concentrate on diversity metrics and propose a new measure with the aim of preserving population diversity between generations. Since capacity metrics and convergence metrics are not related to our aim, the interested reader is referred to [20, 29] for further information. More specifically, we investigate three representative groups of diversity metrics including: distribution, spread and distribution-spread.

3.3 Diversity metrics

Most researchers have been trying to define a description that can characterize the concept of population diversity. In the literature, scientists mentioned that Diversity metrics should indicate the distribution and spread in the optimal solution set S [20]. Here, Fig. 1 is presented to convey the concept of distribution and spread.

In this figure a 2-dimentional PF is presented. The bounding solutions of the PF are (0,1) and (1,0) which are known as extreme points. The optimal solution set S with five non-dominated solutions in Fig. 1a shows good distribution and poor spread since S does not contain bounding solutions. Conversely, the five non-dominated solution of optimal solution set *S* in figure (b) have good spread but poor distribution since the solutions in S are not scattered evenly.



(a) Good distribution, Poor spread (b) Poor distribution, Good spread

Fig. 1 Distribution and spread in a set

To have a better comprehensive investigation, this part express three classes of diversity metrics. As mentioned before, diversity metrics are aimed to measure the distribution and spread of solutions in the optimal solution set S [20]. So researchers categorize diversity metrics in to three classes:

1- diversity metrics which focus on distribution
2- diversity metrics which focus on spread
3- diversity metrics which focus on both distribution and spread

3.4 Diversity metrics which focus on distribution

As mentioned before Distribution focuses on how evenly the solutions of an approximation set are scattered in the objective space [20]. Here we have investigated five representative diversity metrics which focus on distribution including: Δ' [30], M_3^* [31], M_2^* [31], Uniform Distribution(UD) [32] and Spacing(SP) [33].

1) Δ' : This metric is proposed by Deb et al. which compares all the consecutive distances of solutions' with the average distance and is computed in the following way:

$$\Delta'(S) = \sum_{i=1}^{|S|-1} \frac{(d_i - \bar{d})}{|S|-1} \tag{4}$$

where S is a solution set and d_i is the Euclidean distance between consecutive solutions in S, and $\bar{d}$ is the average of d_i. The best distribution would produce a zero value when all the solutions have equal distance to their successor. The consecutive solutions are defined to be the next solution in the lexicographic ranking of the population [30]. As you can see Δ' is a non-parametric metric with computational complexity of $O(m|S|^2)$.

2) M_3^* : It is a metric which is similar to Δ' but instead of average distance, this metric considers the maximum distance instead [31]. So M_3^* is also a non-parametric metric with computational complexity of $\mathrm{O(m|S|^2)}$.
3) M_2^*: The M_2^* metric is proposed by Ziztler and Deb et al. [31] which operates with niche radius(σ) parameter and its formulation is as follow:

$$M_2^*(S) = \frac{\sum_{\vec{s} \in S} |\{\vec{s}_2 \in S| |\vec{s}_1 - \vec{s}_2| | < \sigma\}|}{|S|-1} \tag{5}$$

For each solution this metrics compute the average number of solutions in its local neighborhood defined by the niche radius parameter. As you see M_2^* is designed with user-specified parameter (σ) and its computational complexity is $O(m|S|^2)$.

4) Uniform Distribution(UD): Uniform Distribution metric reveals the degree of uniformity on a given distribution. This measure computes the standard deviation of the number of neighbors for each solution. These neighbors are determined using a radius parameter. For a given population S the uniform distribution would be computed with the following formula [32]:

$$UD(S) = \frac{1}{1 + D_{nc}} \tag{6}$$

,

$$D_{nc} = \sqrt{\sum_{\vec{s}_i \in S} \left(nc(\vec{s}_i) - \overline{nc}(\vec{s})\right)^2 / (|S| - 1)} \tag{7}$$

,

$$nc(\vec{s}_i) = \left|\left\{\vec{s}_j \in S \middle| \left|\vec{s}_i - \vec{s}_j\right| \right| < \sigma\right\}\right| - 1 \tag{8}$$

In this equation $\overline{nc}(\vec{s})$ is the average of $nc(\vec{s}_i)$ This metric is designed with a user-specified parameter which determines the neighbors of a solution and its computational complexity is $O(m|S|^2)$.

4) Spacing (SP): In [33], the Spacing (SP) metric is defined as:

$$SP(S) = \sqrt{\frac{\sum_{i=1}^{|S|-1}(d_i - \bar{d})}{|s| - 1}} \tag{9}$$

$$d_i = \min_{\vec{s} \in S\ \vec{s}_i, \neq \vec{s}_j} \left|\left|F(\vec{s}_i) - F(\vec{s}_j)\right|\right| \, and \ s_i \in S \tag{10}$$

In order to compute mutual distances this metric finds closest distance instead of lexicographic order. SP is a non-parametric metric with computational complexity of $O(m|S|^2)$.

3.5 Diversity metrics which focus on spread

The spread property of a solution set S quantifies how much the solutions of S cover the PF completely or in other words how much the extreme points are explored. The Overall Pareto Spread (OS) is a famous metric for assessing the spread of a set which is proposed by J. Wu in [34] and expressed as:

$$OS(S, P_{Good}, P_{Bad}) = \prod_{k=1}^{m} \frac{\left|\max_{\vec{s} \in S} f_k(\vec{s}) - \min_{\vec{s} \in S} f_k(\vec{s})\right|}{|f_k(P_{Bad}) - f_k(P_{Good})|} \tag{11}$$

where $\max_{\vec{s} \in s} f_k(\vec{s}) f_k(\vec{s})$ are the maximum and minimum values of the k^{th} objective in S, respectively. The metric OS has a computational complexity of $O(m|S|)$. In this metric we need to know to points in the objective space: a good point and a bad point. P_{Bad} is a point in the objective space that is dominated by all the solutions of the Pareto front and P_{Good} is a point that dominates all the point of the true Pareto front. Note that the good point would not be feasible at all but here it is used to find the bounds of Pareto front. In this paper we apply this measure for our comparison as a nominee of this group and the good and bad points are approximated using the offline archive of the algorithm.

3.6 Diversity metrics which focus on both distribution and spread

This group of metrics considers both the distribution and the spread of a solution set S simultaneously. Here we investigate five representative diversity metrics which focus on both distribution and spread of a set which are: Δ [35], Generalized Spread (Δ*) [20, 36], NDC_μ and CL_μ [20, 37], Metrics σ and $\bar{\sigma}$ [20, 38] and The Entropy-based metric [39]

1) Δ : The metric Δ is introduced by Deb et al. [35], which is defined as follows:

$$\Delta(S, P) = \frac{d_f + d_l + \sum_{i=1}^{|S|-1} |d_i - \bar{d}|}{d_f + d_l + (|S| - 1)\bar{d}} \tag{12}$$

where d_i is the Euclidean distance to the consecutive solution of i'th solution and $\bar{d}$ is the average of d_i. The terms d_f and d_l are the minimum Euclidean distances from solutions in S to the extreme (bounding) solutions of the PF (P).

2) The Generalized Spread (Δ*): Requiring consecutive sorting in metric Δ restricts its application to 2-dimensional PFs only. Zhou et al. [36] introduced the Generalized Spread (Δ*) as an extension of Δ, which takes the following form and can be exploited in high dimensional spaces.

$$\Delta^*(S, P) = \frac{\sum_{k=1}^{m} d(\vec{e}_k, S) + \sum_{i=1}^{|S|} |d_i - \bar{d}|}{\sum_{k=1}^{m} d(\vec{e}_k, S) + (|S|)\bar{d}} \tag{13}$$

,

$$d(\vec{e}_k, S) = \min_{\vec{s} \in S} ||\mathrm{F}(\vec{e}_k) - \mathrm{F}(\vec{s})|| \tag{14}$$

,

$$d_i = \min_{\vec{s}_j \in S, \vec{s}_i \neq \vec{s}_j} \left|\left|F(\vec{s}_i) - F(\vec{s}_j)\right|\right| \, and \ s_i \in S \tag{15}$$

Where $\vec{e}_k \in P$ is the extreme solutions on the kth objective and d_i is to identify the closest pairwise solutions in S, and $\bar{d}$ is the average of d_i.

These two measures (Δ and Δ*) require true PF for comparison set and both of them have the computational complexity of $O((m|S|^2) + m\,|S|\,|P|)$.

3) Metrics NDC$_\mu$ and CL$_\mu$: The value of these metrics is obtained by partitioning the objective space in to $\left(\frac{1}{\mu}\right)^{\mathrm{m}}$ grids where μ is a variable that can be varied between 0 and 1 [20, 37]. The number of grids that contains solution are take a part in these metrics. Since these metrics construct a grid structure their computational complexity exponentially increases with the dimension of space. These metrics also demand subtle parameter settings. (number of grids)
4) σ and $\bar{\sigma}$: Using a set of lines passing from the origin, the objective space would be divided into equal angels. The metric value is based on the number of reference lines. The computational complexity of these two metrics is also exponential and also the values of these metrics relay on the parameter settings (number of reference points, sub-regions or angles and etc.) [20, 38].
5) Entropy-based metrics: This metric also divides the objective space into some sub-regions and grids. By employing an influence function (Gaussian in this paper), it would compute a density value for each cell of the grid. Then the Entropy-based metric calculates the Shannon entropy of these density values [39].

As you see the main part of our proposed framework is monitoring and computing the population diversity. Here, we want to explore the possibility of these popular diversity metrics being incorporated in our mechanism. In this study we picked one candidate from each category, in order to compare with our proposed measure.

For the first category which is focused on the distribution of solutions we picked Spacing (SP). Note that Δ' and M_3^*. use consecutive distances and require the lexicography order of solutions in the population. Also M_2^* and Uniform Distribution (UD) are depended on user-specified parameters σ and the neighbors of a solution). For the second category, which is focused on spread, we selected Overall Pareto Spread (OS) and finally for the third category which considers both distribution and spread we chose Entropy-based metric. In this category Δ and Δ^* require the True Pareto Front for comparison and it is not applicable for our purpose. The computational complexity of other Metrics of this group (CL_μ and σ , $\overline{\sigma}$ and Entropy) are high but in this work we picked Entropy-based metrics as an example of this category.

All these metrics are either computationally expensive or are not able to capture the diversity of a solution set in high dimensions. Also many of these metrics relies on specific knowledge of the true Pareto front which is not available in practice and this would prevent their exploitation in the mechanism of evolutionary algorithms. We have designed a parameter free diversity measure which can be effectively incorporated in evolutionary algorithms and also has a polynomial computational complexity. The new diversity measure is based on shortest Hamiltonian path and supposed to effectively evaluate the diversity of populations in high dimensional spaces. A summary of all these classes of diversity measure is revealed in Table 1.

3.7 Proposed Diversity Measure

To explain our measure, let us assume that we are given two approximation sets or generally two sets of points in the objective space. For each set, we compose a complete weighted graph using the points in the set and their mutual distances. In such a graph, corresponding to each point we have a node which is connected to all other nodes. Then, we compute mutual distances between all the points and set them as a weight for the corresponding edge. We need to capture the distance between nearby points and the best tool for this purpose is a Hamiltonian path. A path is called Hamiltonian if it passes through all the points once and only once. Here, we need to find a Hamiltonian path with shortest length. It is worth noting that this effort is different from finding nearest point for each solution, which has been utilized in other measures like generalized spread [36]. Finding

Table 1 A summary of different classes of diversity measure

Classes of diversity metric	Metrics complexity	Parametric on set	Computational	Comparis
Distribution	Δ',M$_3^*$,SP	No	$O(m\|S\|^2)$	None
	M$_2^*$,UD	Niche radius (σ)	$O(m\|S\|^2)$	None
Spread	OS	No	$O(m\|S\|)$	P$_{good}$,P$_{bad}$
Distribution & Spread	Entropy, NDC$_\mu$ and CL$_\mu$ σ and $\bar{\sigma}$ Δ, Δ^*	# of grids, # of reference points, sub-regions, angles, μ	exponential	None
	No	$O((m\|S\|^2)+m\|S\|.\|P\|)$		Pareto front

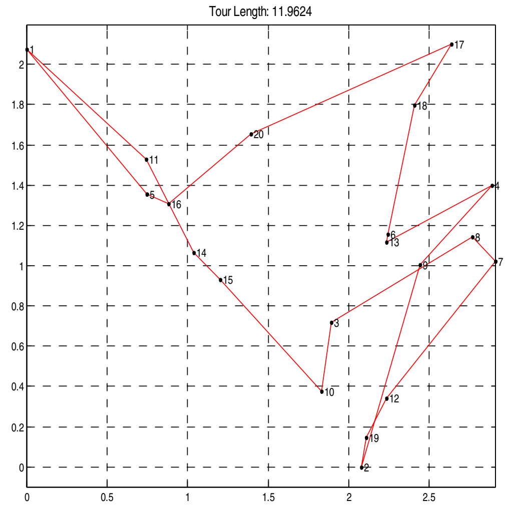


Fig. 2 Graph of a sample solution set

nearest point can be ineffective and there are usually situations that a distance would be missed while other distances might be considered multiple times, for more details see [1, 20].

The problem of finding shortest Hamiltonian path is NP-complete [40, 41], but there exist several heuristic approaches with polynomial complexity that yields an acceptable approximation to the optimal path. In this study, we used 2-opt algorithm which has a proven complexity of $(n^{1.2})$ [41]. This algorithm starts by a random Hamiltonian path in the graph and searches for a locally optimal solution. In each step a change involving two edges is applied to the path and any improvement would be applied in a greedy fashion. The algorithm continues until no change, involving two edges, would improve the path. These paths are identified in order to quantitatively compare the diversity of their corresponding sets of points. We propose to treat these paths as time series using which we can better compare any change in the density of the sets. In this regard, we need to define and find segments in each path.

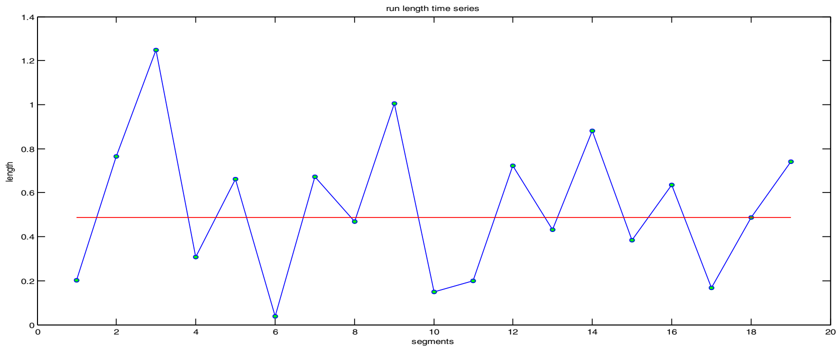
Each edge in the optimal Hamiltonian path is called a segment. So, for a graph with n nodes we will have n-1 segments. We assume a natural order for the segments, the first segment is the edge between first and second nodes of the optimal path. The length of a segment is the weight of its corresponding edge and the length of a path is the total sum of the edge weights. As an example, we have constructed a graph from a sample set of points whose corresponding optimal path is illustrated in Fig. 2.

In order to compare the obtained paths, one can simply compare the sum of distances for each path but this can be ineffective in high dimensional spaces. Known metrics tend to lose their granularity in higher dimensions due to the infamous “curse of dimensionality” [20]. Beside, avoiding aforementioned problem, the main advantage of interpreting these paths as time series is that we can compare each set by their acquired time series and there exists a profound history of time series research that we can exploit.

After determining the segments, a time series is constructed using these segment lengths. We visualize this time series in a 2D graph where the x-axis shows the segments in their order and the y-axis shows their corresponding length. Figure 3 illustrates the time series corresponding to the example shown in Fig. 2. The median segment length is indicated by a horizontal line (red line in Fig. 3). In order to compare two time series, we borrow a concept from statistics, named run statistic. Each set of consecutive segments which all have a length below or above the median, is called a run [42]. In other words, if we travel on the time series’ plot, the set of all segments that we pass by, before crossing the median line, is called a run. After crossing the median line, we enter a new run. When a run is below the median it contains points that are close to each other which implies that the area represented by this run is dense; on the other hand, when it is above the median line, it contains points that are scattered. We can incorporate this knowledge obtained from time series into diversity measure.

By considering the number and the length of these runs we can compare the diversity of two sets without directly comparing the distances between points. In order to compare these time series precisely, a numerical value should be

Fig. 3 Time series for the found path in Fig. 2



extracted for each of them. We propose to use the following equation in diversity comparison mechanisms:

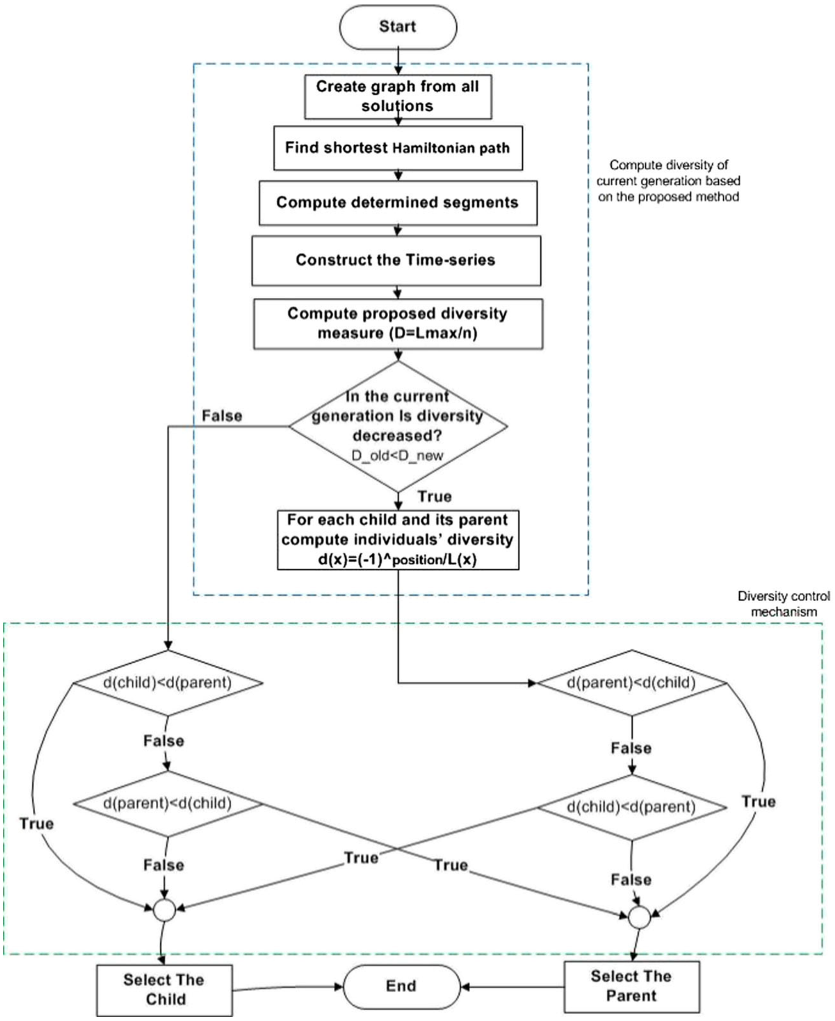
$$D = \frac{L_{max}}{n_l} \tag{16}$$

In this equation L_{max} is the length of the largest run below the median and n_l is the number of runs below the median. Here for more simplicity, the runs located below the median are called sub-median and the runs located above the median are called super-median. Since sub-median runs are indicator of how close the points are to each other, we decided to incorporate these runs into the proposed diversity measure. A set with a large sub-median run contains an area of dense points so it is not considered as a diverse set. In an ideal set, each point of the time series should constitute a single run and half of them would be sub-median. So, smaller values of this quantity is desired and lower values of D correspond to more diverse sets. Our proposed diversity method is suitable for high-dimensional problems since its complexity is polynomial in terms of number of objectives and also it uses a graph structure to find the order of solutions in the space which is more effective or efficient in higher dimensions than using direct techniques [41]. In the next sub-section utilization of this quality measure will be explained.

3.8 Diversity maintenance mechanism by means of proposed diversity measure

Here in this section, we aim to utilize the proposed metric into the evolution of algorithms and will propose a diversity control mechanism to streamline the optimization process. As stated in many other works, there is a dilemma between controlling diversity and converging to the Pareto front in MOEAs which is directly related to amount of selection pressure applied by the algorithm. In this framework we want to make a balance between these two contradicting objective by maintaining a steady selection pressure in the survival selection operator. In the previous sub section, we proposed a diversity metric which is supposed to effectively

Flowchart 2 Flowchart of the teamwork of proposed diversity measure in proposed maintenance mechanism


Start
Create graph from all solutions
Find shortest Hamiltonian path
Compute determined segments
Construct the Time-series
Compute proposed diversity measure (D=Lmax/n)
Compute diversity of current generation based on the proposed method
In the current generation Is diversity decreased? D_old<D_new
False
True
For each child and its parent compute individuals' diversity d(x)=(-1)^position/L(x)
Diversity control mechanism
d(child)<d(parent)
False
d(parent)<d(child)
True
False
True
d(parent)<d(child)
False
d(child)<d(parent)
True
True
False
Select The Child
End
Select The Parent

evaluate the diversity of populations in high dimensional spaces. Using this measure, we can detect any variation in the population diversity at each generation. After we identified that the population diversity is decreased we should regulate the selection mechanism in a way to compensate for this reduction. In order to incorporate diversity into the selection mechanism of algorithms, in a more controlled way, we proposed a diversity maintenance framework. This framework is concisely illustrated in the Flowchart 2 where it uses the diversity measure described in the previous subsection. In most of common MOEAs, solutions are selected according to their fitness and diversity is used as a tie breaker [1]. We propose to alter this order adaptively. For instance, there may be situations where the population is converging toward a specific region of the front or even a single point. In such a situation, the selection operator should pay more attention to the population diversity than fitness.

As stated before the two important issues in our proposed framework was finding a schedule for switching the order of selection and determining the chromosome which has the highest contribution in the diversity of the population to give it an advantage in the selection mechanism. For the former we compare the diversity of each generation to its predecessor and change the selection operator accordingly. Assume that, using the proposed diversity measure, we have acknowledged that diversity is decreased. So, we need to change the selection mechanism in favor of diversity control. In this situation solutions are selected according to their contribution to the diversity of the population rather than their fitness. While choosing between a child and its parent, the parent will be selected if it is located in a super-median run and the child resides in a sub-median run, since it is in a more scattered area of the space. And reversely the child will be chosen if it belongs to a super-median segment and the parent belongs to a sub-median one. If they both are in either above or below of the median line, the selection will be based on the run length. Different cases are illustrated in Table 2. Note that when they both are below the median we are selecting between two dense areas so the one with shorter run length is in a less crowded region, i.e. it contains fewer points. In the same way, when solutions are above the median, they are both in a sparse area and we select the one with larger run length which means less diverse.

Table 2 Different possible selection states

Parent	Child	Selection
Above	Below	Parent
Below	Above	Child
Below	Below	Smaller length
Above	Above	Larger length

The whole procedure of selection according to diversity can be integrated into the following formula:

$$d(x) = \frac{(-1)^{[xisabovethemedium]}}{L(x)} \tag{17}$$

In this equation we used Iverson bracket, here it returns one if x is above the median and returns zero otherwise. Also note that L(x) is length of the run containing x. Points with smaller d values are in more sparse regions of the approximation set. By this equation we determine which chromosome has the highest contribution in the diversity of the population so we give it an advantage in the selection mechanism.

To better understand the selection, consider the plots in Fig. 4 that correspond a hypothetical offspring population and its parent population set. We are to select between parents or their offspring.

First of all, note that the selection is according to diversity. When comparing either A and D or B and C the selection is based on their position relative to median line. So, for these comparison the solutions below the median line, namely A and C, will be selected. But when the solutions are on the same side of median line we must consider

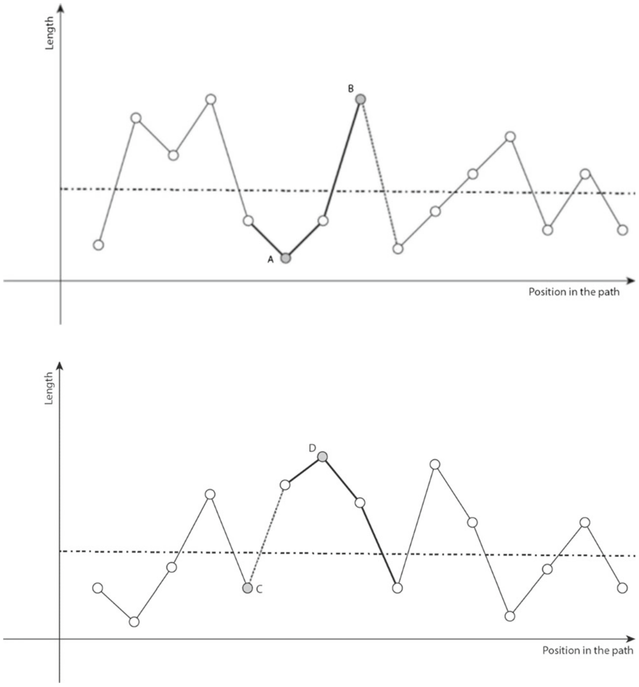


Fig. 4 These plots represent the time series constructed from two hypothetical populations. The horizontal dashed line indicates the median length and two runs are highlighted in each plot

more information. For instance, when selecting between *A* or *C* since both are below median we select the one which resides in a shorter run, in this case *C*. Note that *C* is in a run with one segments while *A* is inside a run with three segments. When comparing *B* and *B* again we need to look at the run lengths since they both are above median. Here we select the one which is in the longer run. As highlighted in the figure, B is in a run with one segment while D is in a run with three segments so D will be selected.

3.9 Incorporation in MOEAD

So far, we proposed a measure to diagnose the reduction of population diversity along the generations in the high dimensional spaces and using this knowledge repaired and reformed the mechanism of selection based on chromosomes' participation in either fitness or diversity. In this section, in order to investigate the capabilities of both the proposed diversity measure and the proposed framework we have made few modifications in the MOEA/D [10] algorithm. Actually the selection mechanism of MOEA/D is modified to use our method which is illustrated in Fig. 3. Despite the fact that MOEA/D is a well-known method in the MOEA community for sake of completeness, we have provided a pseudo-code of the modified version of it in Algorithm 1. In this algorithm the selection function (line 10) is the procedure described in Section 3.3, which adaptively changes the selection priority according to the difference in the computed diversity of current population in comparison to the previous one. Also, in line 5 for computing the diversity of current population we use our devised diversity metric which was described in details in Section 3.2.

Algorithm 1 Incorporation of proposed selection mechanism in MOEA/D.

```
1. Initialization {
2. generate uniform weights
3. find neighbors of each weight
4. P := create initial random population
5. Dp := Diversity(P)                        // Algorithm 3
   and Equation (16)
6. FitnessFirst := True
7. }
8. while not finished do
9. Q := Variation()
10. P := Selection(P, Q, FitnessFirst)       // Flowchart1
11. Dq := Diversity(P)                       // Algorithm 3
    and Equation (16)
12. FitnessFirst := $D_p < D_q$
13. $D_p$ := $D_q$
14. end while
```

Algorithm 2 The algorithmic structure of proposed method

```
function Selection(P,Q, FitnessFirst)
R = []    //empty population
for i = 1 to |P|
    if (FitnessFirst == true)
        if (P[i].fitness > Q[i].fitness || d(P[i]) >
        d(Q[i]))
            R[i] = P[i]
        else if (d(P[i]) < d(Q[i]))
            R[i] = Q[i]
        else
            R[i] = Random Q[i] or P[i]
    else
    if (d(P[i]) > d(Q[i]) || P[i].fitness >
    Q[i].fitness)
            R[i] = P[i]
        else if (P[i].fitness < Q[i].fitness)
            R[i] = Q[i]
        else
            R[i] = Random Q[i] or P[i]
end for
return R
```

4 The Experiments

This section describes the computational results of the evaluation of the proposed method that was introduced in the previous section. This section outlines three parts. The first part covers the experimental setup for the proposed method. The second part will explain our assessment methodology and two well-known evaluation metrics and finally the third part reports results of the application of the proposed method on a set of well-known artificial problems to test the performance of the proposed algorithm.

4.1 Experimental setup

To prove the potential of our method, we aim at comparing and demonstrating its behavior with known competing algorithms to well-known criteria in terms of diversity and convergence. For this target, we used MOEA/D as a baseline for comparison. Since the proposed mechanism is presumed to be an improvement over MOEA/D this comparison is acceptable. In order to expose the prospects of our method in solving many objective optimization problems, we have also included NSGA-III [16] in our comparisons.

Many test problems have been proposed in the literature where we decided to utilize the most comprehensive and fairly recent WFG problem family [43], since it covers various types of fronts and different form of hindering

Table 3 Main characteristics of WFG problems (NS: non-separable, S: separable, U: uni-modal, M: multimodal, D: deceptive, PD: parameter dependent bias)

Problem	Feature
WFG2	NS, (U/M), Convex, Disconnected
WFG4	S, M, Concave
WFG5	S, D, Concave
WFG6	NS, U, Concave
WFG7	S, U, Concave
WFG8	NS, U, PD, Concave
WFG9	NS, M, D, PD, Concave

attributes. WFG Toolkit is designed in a systematic way that can introduce many different attributes to a test problem. Using this toolkit, they have created a set of 9 test problems. The essential properties of each problem in the WFG family are listed in Table 3. In a separable problem the Pareto front obtained from optimization using all variables would be equal to union of all Pareto fronts obtained from optimization using each single variable. The modality stated in Table 2 is for the function used for transforming the decision space to the objective space. For more information on these properties one can refer to [43].

WFG problems are scalable in terms of number of objectives and we used problems with 2, 3, 4, 6 and 8 objectives. In our experiments, we used an open source Java library named jMetal, which is a well-known framework for implementing evolutionary algorithms [44, 45]. JMetal contains original implantations of WFG problem family as well as MOEA/D and NSGA-II algorithms.

There are a few parameters that must be set in both algorithms and problems, which are summarized in Table 4. These parameters are either selected empirically or chosen the same as the values proposed by the original authors. The invoked variation operators are polynomial mutation and simulated binary crossover with the same parameterization as mentioned in jMetal framework. Other necessary evolutionary parameters are also configured with the default setting of jMetal.

Table 4 Parameter settings

Parameter	Value
Population Size (N)	100
Goal Population Size (N_{goal})	100 * # objectives
Maximum Iterations	200
Tournament Size	2
SBX distribution index	15
PM distribution index	20
Crossover rate	1.0
Permutation rate	1/30
Position parameter, k	#objectives 1
Distance parameter, l	10

4.2 Performance measures

There are various performance measures used for evaluation of MOEAs. Two of the most commonly used measures are Inverted Generational Distance (IGD) [29] and Hypervolume [27, 28]. These two measures are widely accepted among the researchers in this field and we have employed them in our study. Both measures are capable of evaluating diversity as well as convergence to the Pareto front [20]. These measures are described in the following paragraphs. Hypervolume computes the volume dominated by the points in the approximation set. In order to compute this value one needs a point or a set of points as a reference to limit the volume with it. It can be defined formally in the following way [27, 28].

$$S(X, R) = \left(\int \bigcup_{x \in X} \{x' | x \preceq x' \preceq R\} \right) \tag{18}$$

Here X is the approximation set we want to evaluate and R is the set of reference points. It is shown that hypervolume is the only unary indicator that is compliant with the Pareto operator [26].

A poor choice of reference point can significantly affect the performance of this metric [31]. We normalized all the values in the [0, 1] interval and chose 1 as the reference point for all the problems. Computation of hypervolume is exponential in number of objective and computing hypervolume for problems more than 6 objectives is impractical [19, 26, 31]. In this study we used a Monte Carlo simulation procedure to approximate the volumes for problems with more than 6 objectives [19]. In order to compute inverted generational distance, we need the true Pareto front or a representative subset of it. IGD is computed as the mean distance from each point in the true PF to the nearest point in the approximation set [29].

$$IGD(P^*.P) = \frac{\sum_{vep^*} d(v.P)}{|P^*|} \tag{19}$$

Here, P is the approximation set we are evaluating and P^* is the true Pareto front. Also, $d(v, P)$ is the distance between v and the nearest point in P.

Table 5 Obtained result for 2D-3D-4D-6D-8D convex problem

		Hypervolume			IGD		
	# Objs	Proposed	MOEAD	NSGAIII	Proposed	MOEAD	NSGAIII
WFG2	2	0.5458020	0.5526240	0.5465730	0.0024320	0.0021420	0.0023980
	3	0.8854370	0.8802010	0.8832352	0.0021530	0.0027310	0.0024292
	4	0.8790730	0.8764339	0.8824604	0.0036010	0.0054720	0.0040103
	6	0.8504630	0.8349710	0.8510533	0.0040140	0.0048960	0.0043229
	8	0.9139860	0.8903880	0.9028583	0.0043590	0.0049880	0.0045429

4.3 Results

To have a satisfactory and comprehensive analysis, this paper has evaluated the proposed method using both performance metrics (Hypervolume and the IGD). In the following tables the values of HV and IGD are reported for the algorithms on the selected WFG problems.

Table 6 Obtained hypervolume values for 2,3,4,6, and 8 objective concave problems

		Hypervolume			IGD		
	# Objs	Proposed	MOEAD	NSGAIII	Proposed	MOEAD	NSGAIII
WFG4	2	0.2059640	0.2129911	0.2091233	0.0002481	0.0001219	0.0001240
	3	0.3460447	0.3916662	0.3813246	0.0010391	0.0010129	0.0010280
	4	0.3897091	0.3409792	0.3409792	0.0020820	0.0042571	0.0028381
	6	0.3032500	0.2523533	0.2843258	0.0031410	0.0055255	0.0036428
	8	0.2684160	0.3133470	0.3234750	0.0035010	0.0045855	0.0040402
WFG5	2	0.1856145	0.1890576	0.1842582	0.0003833	0.0003619	0.0003739
	3	0.3345456	0.3568027	0.3437914	0.0009962	0.0008614	0.0008642
	4	0.3579368	0.3271197	0.3370538	0.0025528	0.0035105	0.0003418
	6	0.3129230	0.3300450	0.3156360	0.0040390	0.0053696	0.0051715
	8	0.3400830	0.3811290	0.3763447	0.0041030	0.0052256	0.0052561
WFG6	2	0.2055719	0.2129355	0.2100539	0.0001877	0.0000270	0.0000257
	3	0.3851618	0.4023562	0.3977418	0.0009152	0.0011171	0.0009889
	4	0.3596272	0.4261959	0.4065304	0.0030080	0.0036563	0.0039456
	6	0.2796310	0.3259030	0.3357770	0.0052710	0.0059357	0.0057249
	8	0.3177050	0.3609460	0.3707799	0.0052843	0.0059550	0.0053605
WFG7	2	0.2067482	0.2131609	0.2214676	0.0006532	0.0003960	0.0003128
	3	0.3586452	0.4031093	0.4194321	0.0009241	0.0008830	0.0007752
	4	0.3676873	0.4292885	0.4314389	0.0028611	0.0034743	0.0031266
	6	0.3949300	0.4358010	0.4234555	0.0038320	0.0063718	0.0069135
	8	0.3243510	0.4226980	0.4047363	0.0050660	0.0062044	0.0067976
WFG8	2	0.4150068	0.4212617	0.4224958	0.0025378	0.0046303	0.0046303
	3	0.5965659	0.6469618	0.6503757	0.0030006	0.0039740	0.0039740
	4	0.5760403	0.7070111	0.7156532	0.0030209	0.0045937	0.0045937
	6	0.5386450	0.4483630	0.5223528	0.0036750	0.0090459	0.0083864
	8	0.5158290	0.4786990	0.5131469	0.0049960	0.0084600	0.0062698
WFG9	2	0.2161048	0.2249586	0.2133657	0.0001049	0.0000591	0.0000895
	3	0.3592907	0.3874917	0.3908650	0.0009848	0.0010696	0.0010197
	4	0.3663834	0.3264170	0.3695242	0.0022786	0.0042197	0.0028248
	6	0.2491770	0.1879360	0.2382277	0.0035850	0.0061390	0.0056837
	8	0.1949910	0.2473860	0.2593254	0.0057860	0.0056510	0.0057786

Tables 5 and 6 illustrates the average of 30 independent outcomes for each algorithm for concave and convex WFG problems respectively. Here, several test problems with various numbers of objectives (2-8) are used to demonstrate the prospective performance of the proposed method. In these tables, the numbers in each cell is the average of both HV and IGD values of 30 runs.

The evaluation of the proposed method on the benchmark test problems had shown a significant improvement in the solution space diversity. Since higher values for Hypervolume and lower values for IGD signify better approximation of Pareto front. These results show that the proposed method performs fine in high dimensional spaces. In order to determine the significance of these results we have used Kruskal-Wallis non-parametric statistical test [46]. The bolded numbers in each row are statistical different from others with 98 % confidence. For better comparison of the results, we have included box plot for all the problems which are depicted in the Figs. 5 and 6. These pictures show that the true median of the proposed method is higher (better) than its competitor.

Results depicted in Tables 5 and 6 indicate that, generally our method has better values of IGD while MOEAD shows better performance in terms of HV. This distinction becomes more obvious as the number of objectives increases. The relative performance of the two algorithms is different for convex problems in comparison with concave problems. In concave problems the proposed method performs better in terms of both assessment measures while in convex problem it shows superiority only in terms of IGD. But the question is which of the two measures are the true indicator of algorithms' performance. We answer this question in the next section.

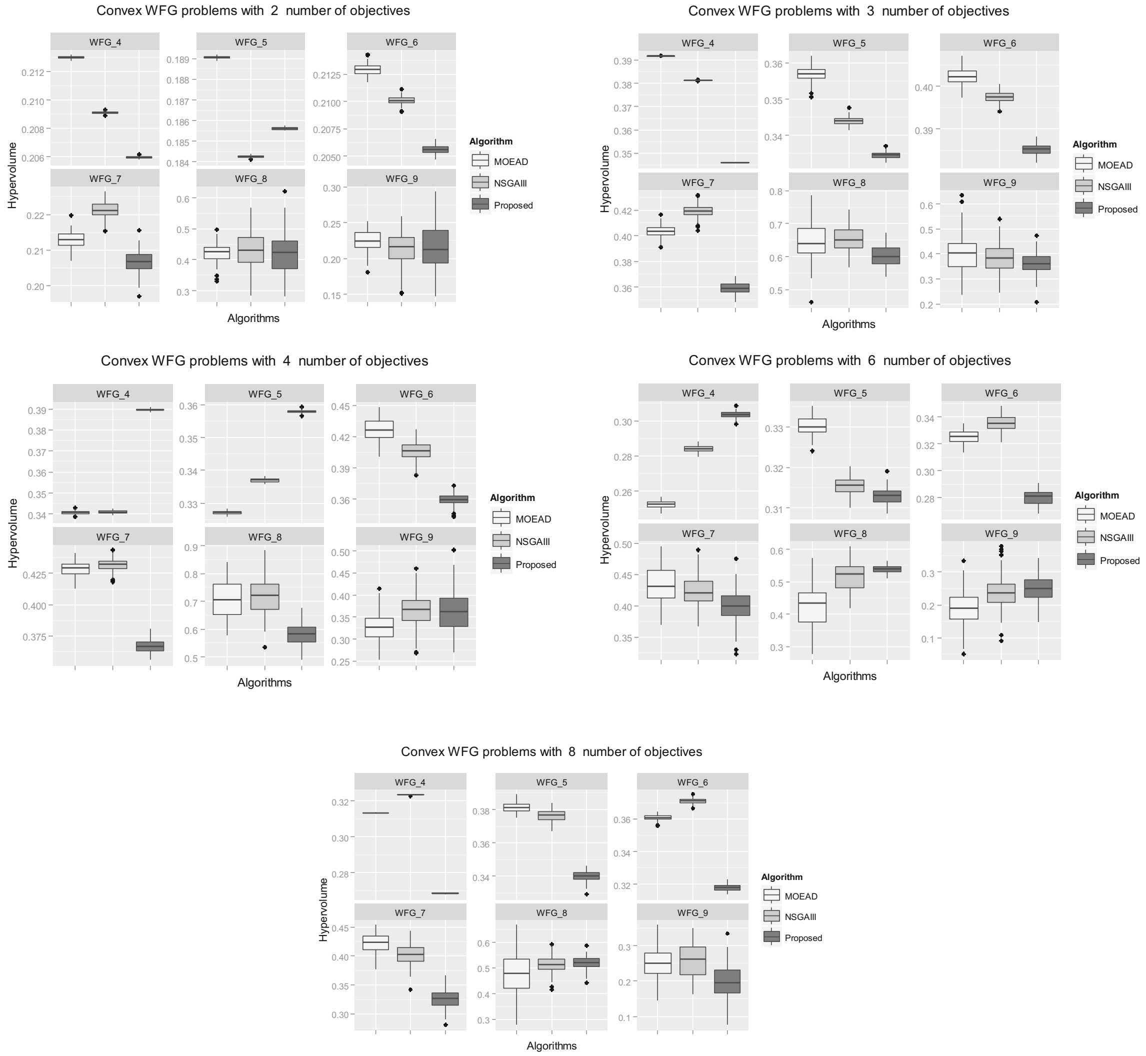


Fig. 5 Box plots comparing hypervolume of each algorithm for 30 independent runs

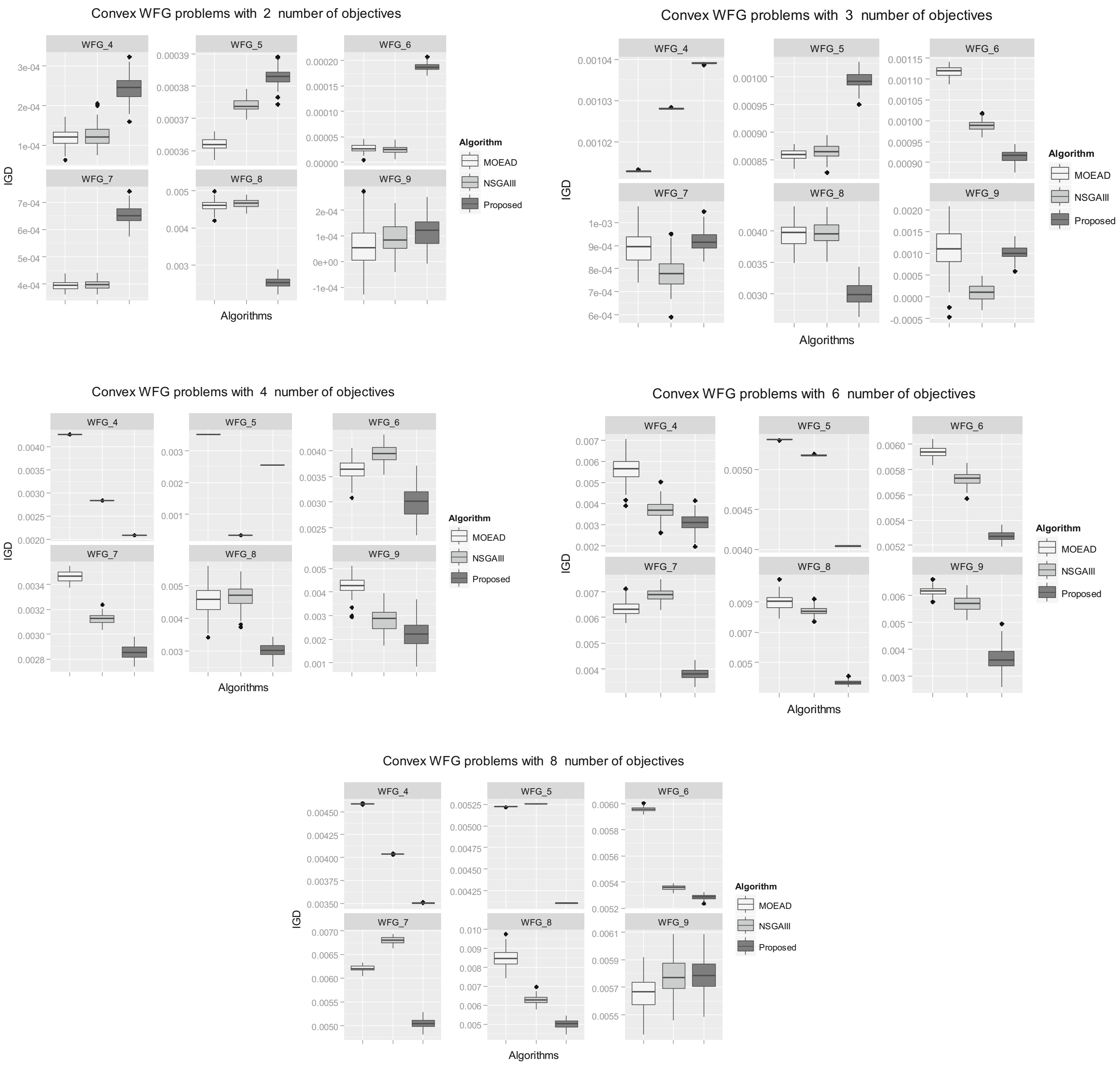


Fig. 6 Box plots comparing IGD values of each algorithm for 30 independent runs

4.4 Comparison with other classes of optimizers

In this section, we compare our algorithm with other prominent classes of multi-objective optimizers. As claimed in previous studies [49, 50] other types of population based optimizers can perform better than common multi-objective evolutionary algorithms in certain problem settings; in order to indicate the potential of our method in comparison with these methods we performed this experiment. Differential evolution and particle swarm optimization are two important population based optimization methods which have been generalized for solving multi-objective problems as well [47, 49].

There are many multi-objective particle swarm optimizers (MOPSO) in the literature which are reviewed in the survey by Durillo et al. [51]. One of the effective design choices of an MOPSO is the method of selecting global and personal best locations. Various methods for choosing these locations in each generation is reviewed in the work of Padhye et al. [48]. In order to compare the performance of our algorithm to this class we have selected Speed-constrained Multiobjective PSO (SMPSO) to be included in our experiments [50].

Differential evolution is a special class of evolutionary algorithms suited for real-valued optimization. There are various versions of these algorithms designed for solving multi-objective optimization problems among which Generalized Differential Evolution (GDE3) has shown promising results [49]. We included this algorithm in our experimental comparison to show the competitiveness of our method relative to this class.

Table 7 presents the Hypervolume and IGD values of these two methods along with the corresponding values for our proposed method in solving concave WFG problems. As it can be seen in the table the proposed method has shown superior results in most of the test problems.

Table 7 presents the average of 20 independent outcomes for three algorithms (the proposed method, SMPSO, and GDE3) on concave WFG problems. These test problems have dimensions in range of 2-8 to demonstrate the prospective performance of the proposed method. This evaluation confirms that as the number of objectives in each test function increased, the higher values for Hypervolume and lower values for IGD obtained, comparing to the other two methods, which signifies that the proposed method has better approximation of Pareto front. These results show that the proposed method performs fine in high dimensional spaces. In order to determine the significance of these

Table 7 Comparing hypervolume and IGD values of Different classes of optimizers for concave WFG problems with 2,3,4,6, and 8 objectives

		Hypervolume			IGD		
	# Objs	Proposed	SMPSO	GDE3	Proposed	SMPSO	GDE3
WFG4	2	0.205964	0.1790086	0.2067185	0.0002481	0.0003375	0.0000792
	3	0.3460447	0.3035097	0.3775838	0.0010391	0.0011040	0.0007785
	4	0.3897091	0.3056854	0.3643534	0.002082	0.0025109	0.0023573
	6	0.30325	0.2072907	0.2432389	0.003141	0.0037442	0.0038569
	8	0.268416	0.1451884	0.2278573	0.003501	0.0042793	0.0043368
WFG5	2	0.1856145	0.1870646	0.1886871	0.0003833	0.0003668	0.0003587
	3	0.3345456	0.323611	0.3616125	0.0009962	0.0011220	0.0008430
	4	0.3579368	0.2547447	0.3454159	0.0025528	0.0027813	0.0037543
	6	0.312923	0.2200564	0.3224525	0.004039	0.0047235	0.0043536
	8	0.340083	0.2145077	0.33694	0.004103	0.0050507	0.0047470
WFG6	2	0.2055719	0.195974	0.1839296	0.0001877	0.0001866	0.0003207
	3	0.3851618	0.3031983	0.3348627	0.0009152	0.0013969	0.0010393
	4	0.3596272	0.3178983	0.3557865	0.003008	0.0025698	0.0022131
	6	0.279631	0.2454878	0.2664257	0.005271	0.0058785	0.0059181
	8	0.317705	0.2225549	0.2330374	0.0052843	0.0061269	0.0057033
WFG7	2	0.2067482	0.2017039	0.2116647	0.0006532	0.0001420	0.0000562
	3	0.3586452	0.2758958	0.3710991	0.0009241	0.0014721	0.0009248
	4	0.3676873	0.2869787	0.3800807	0.0028611	0.0026998	0.0022016
	6	0.39493	0.242296	0.2976645	0.003832	0.0042594	0.0044190
	8	0.324351	0.1739119	0.2060591	0.005066	0.0057297	0.0060875
WFG82		0.4150068	0.3661964	0.3691485	0.0025378	0.0038328	0.0038523
	3	0.5965659	0.4563235	0.5708926	0.0030006	0.0041885	0.0040750
	4	0.5760403	0.468932	0.6627424	0.0030209	0.0053372	0.0047031
	6	0.538645	0.417896	0.50806	0.003675	0.0051026	0.0067168
	8	0.515829	0.377654	0.491508	0.004996	0.0072734	0.0069289
WFG9	2	0.2161048	0.213743	0.2148464	0.0001049	0.0001095	0.0000977
	3	0.3592907	0.3253762	0.358357	0.0009848	0.0010571	0.0009102
	4	0.3663834	0.293585	0.3412209	0.0022786	0.0032115	0.0031233
	6	0.249177	0.1572768	0.2023494	0.003585	0.0044994	0.0042507
	8	0.194991	0.1028697	0.1193904	0.005786	0.0065411	0.0067046

results we have used Kruskal-Wallis non-parametric statistical test [46]. The bolded numbers in each row are statistical different from others with 98 % confidence.

5 Discussions

In this section we are going to discuss about the performance of our framework and compare the proposed diversity measure with other diversity metrics. In the first part the workflow of our algorithm is analyzed to find out whether it was able to maintain diversity beside convergence. In the second part we compared the obtained results and provided evidence for the superior performance of our algorithm. Finally, in the last part we considered using other diversity metrics in our framework and compared the results with the main approach.

5.1 Diversity Maintenance

It is widely accepted that the main aggravating factor in evolutionary many objective optimization is diversity reduction and here we will show that how our approach can handle this impediment. To better understand the effect of the proposed framework we have studied the behavior of the algorithm by plotting the populations diversity in all generations which is depicted in Fig. 7 for different test problems. The behavior of the algorithm is similar for different number of objectives so to make the paper brief we only presented the plots for 4 dimensional problems. This figure shows the values of the proposed diversity measure in different generations of the algorithm. As it is illustrated in the Fig. 7, whenever the diversity of a population was decreased, our approach detected this failure and the application of its framework had increased it in the next generation. So, over the course of the algorithm the diversity of the algorithm is maintained at a certain level while the population is being pushed toward Pareto front.

5.2 Outcome analyzing

Results depicted in Tables 5 and 6 indicate that our approach performs significantly better than their original counterparts in terms of IGD; but this is not true for hypervolume

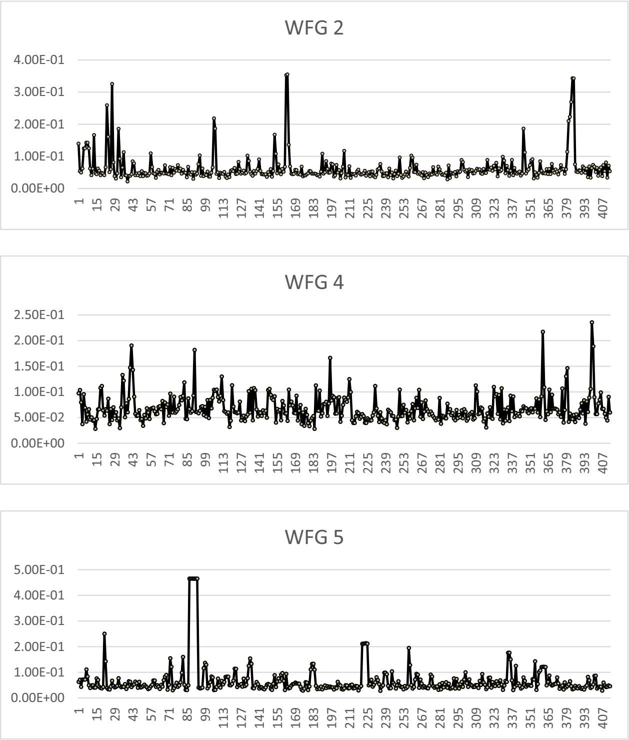


Fig. 7 Diversity plot of generations for each problem, Horizontal axis is the number of generations and vertical axis is the diversity measure computed by (4)

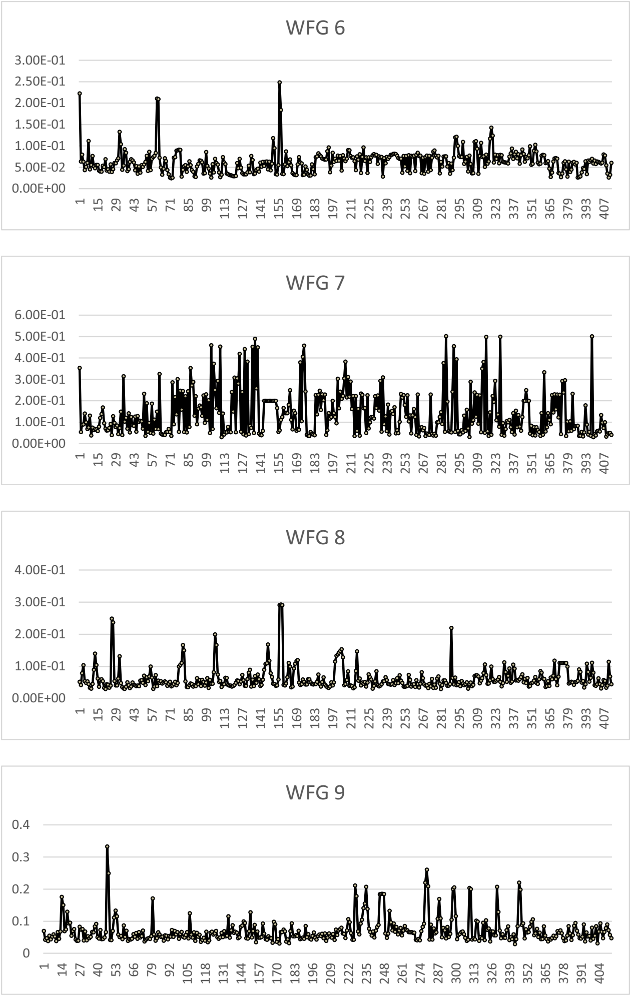
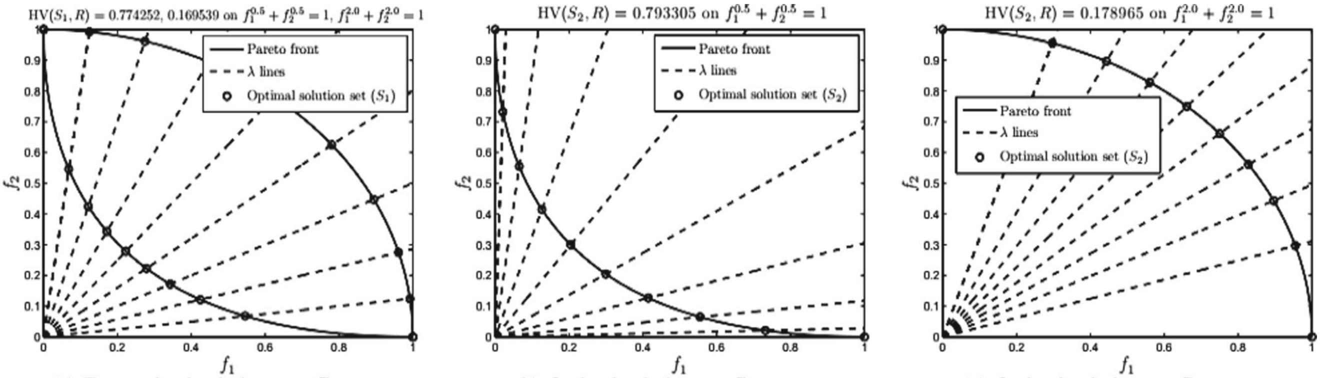


Fig. 7 (continued)



(a) Two optimal solution sets S_1 (b) Optimal solution set S_2 (c) Optimal solution set S_2

Fig. 8 The optimal solution sets S1 and S2 for two dimensions. These figures are generated using the source code provided by the authors of [20]

metric. So the main question is that how can we justify this contradiction? There have been many works on comparison between IGD and HV and their capabilities in the MOEA literature. All of them proved experimentally or theoretically that these measures show contradictory results in problems with concave Pareto front, especially in high dimensional objective spaces [20].

In order to evaluate the proposed method, in this study, we have applied our approach to both concave and convex methods. For convex problems, as you can see in the Table 4, our proposed method shows improvements with regard to both quality measures, IGD and hypervolume. But, in solving concave problems our method performed better only according to the IGD measure and there was no significant improvement in hypervolume values. In order to comprehend which method is showing the correct performance we refer to a comparative review of performance measures used in MEOA by Jiang et al. [20]. They proved that IGD and hypervolume measures act in contradiction to each other when dealing with concave problems. In the following, we will illustrate their point.

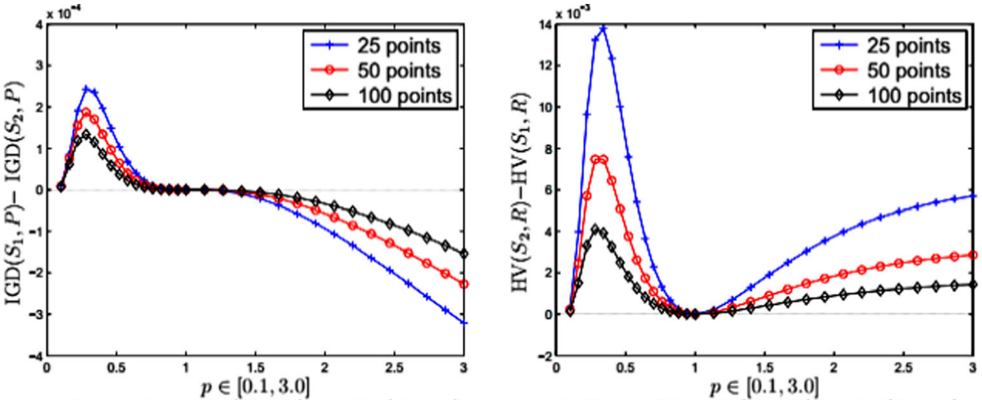
In their study they focused on problems with symmetric and continuous Pareto fronts which may have concave or convex shape. They define these types of fronts using the following equation:

$$f_1^p + f_2^p + \cdots + f_m^p = 1 \tag{20}$$

In this equation the objective values are normalized in [0.1] and p is a non-negative real number. The parameter p controls the geometry of the Pareto front. If p is smaller than 1, the Pareto front will be convex and if it is greater than 1 the Pareto front will be concave. According to the instructions given by Jiang et al. [20], we have created two sets of points, S1 and S2, on a Pareto front as they did which had different characteristics. In convex problems, the solutions in S2 are distributed around the extreme points of the front and are considered to be more diverse than the points in S1. This situation is illustrated in Fig. 7, where the sets are created for a 2-dimensional space.

On the contrary, S2 has better diversity than S1 in concave problems. For more details on generating S2 and S1 the interested readers may refer to [20].

As described in Section 4.2, smaller values of IGD and larger values of HV are desired. Figure 8 compares IGD and Hypervolume values for the previously defined sets. In this figure, the vertical axis indicates the difference of measures



(e) Conv.–Div. IGD(S_1, P) − IGD(S_2, P) **(f) Conv.–Div.** HV(S_2, R) − HV(S_1, R)

Fig. 9 The figures in the section were generated using the source code provided by the authors of [20]

in a way that a positive value means S2 is better than S1 and a negative value means vice versa. The sets have been designed in a way that for convex problems, $0 < p < 1$, S2 would be better than S1 and as we can see in Fig. 9, both measures acknowledge this fact. But in concave problems, $1 < p < \infty$, we know that S1 is better than S2. In this case, while IGD correctly prefers S1, hypervolume favors S2 which is not true. In conclusion, when we have contradiction between IGD and hypervolume in concave problems, IGD results are more reliable. In fact, this distinction is compatible with the definitions of these metrics. Since IGD is fed with a subset of the true Pareto front it have a knowledge of the extreme points of the true front but Hypervolume does not have this information, So IGD can exploit this extra details and yield more reliable comparison.

5.3 Comparison with other diversity metrics

In order to experimentally compare our proposed diversity metric with other measures we have selected three diversity metrics to incorporate in the proposed framework (Section 3.3). These metrics are spacing, overall spread, and the entropy based method proposed in [20] (for more simplicity we call it the entropy metric). These metrics are briefly introduced in the Section 3.1 yet for further information we would refer readers to the original papers. To incorporate a diversity metric in our framework we need to assess the diversity of a population as well as computing the contribution of a single solution in diversity of a set. The former is straightforward since all the measures operate on a set of solutions. But for a solution we used a simple procedure

Table 8 The performance results obtained for various WFG problems in different dimensions

		IGD			
	#objs	Proposed	Entropy	Overall spread	Spacing
WFG4	2	0.000248	0.000638	0.000558	0.000389
	3	0.001039	0.001510	0.001795	0.002193
	4	0.002082	0.002408	0.003486	0.004318
	6	0.003141	0.003627	0.004005	0.004642
	8	0.003501	0.003791	0.004982	0.006005
WFG5	2	0.000383	0.000391	0.000395	0.000397
	3	0.000996	0.001110	0.001141	0.001182
	4	0.002553	0.002163	0.003428	0.004091
	6	0.004039	0.003245	0.004487	0.005942
	8	0.004103	0.004686	0.005281	0.005930
WFG6	2	0.000188	0.000288	0.000360	0.000407
	3	0.000915	0.001319	0.001200	0.001094
	4	0.003008	0.003593	0.003688	0.003817
	6	0.005271	0.005496	0.006241	0.007332
	8	0.005284	0.005853	0.005640	0.005692
WFG7	2	0.000653	0.000639	0.000872	0.001198
	3	0.000924	0.001138	0.001503	0.001776
	4	0.002861	0.003338	0.004043	0.004558
	6	0.003832	0.004860	0.006031	0.007128
	8	0.005066	0.006223	0.006680	0.006996
WFG8	2	0.002538	0.003155	0.003844	0.004526
	3	0.003001	0.002924	0.003943	0.003565
	4	0.003021	0.002885	0.004006	0.004959
	6	0.003675	0.005214	0.006517	0.008791
	8	0.004996	0.005433	0.007413	0.009338
WFG9	2	0.000105	0.000160	0.000124	0.000083
	3	0.000985	0.002229	0.002095	0.001941
	4	0.002279	0.002258	0.003056	0.003817
	6	0.003585	0.005040	0.004953	0.004879
	8	0.005786	0.005975	0.005685	0.005898

Columns are the proposed framework with different diversity metrics in the MOEAD algorithm and the desired contribution is computed as the difference between the population's diversity with and without the considered solution. This method would have extra computations but one should note that this is only for analysis and assessment of our diversity measure.

Table 7 presents the IGD for the diversity control framework with different metrics applied to WFG problems. As you can see our proposed diversity measure performed better in most of the test problems. Since the only difference in the algorithms listed in the columns of this table is the exploited diversity measure, we can claim that our proposed diversity measure based on Hamiltonian path can better capture a population's diversity Table 8.

6 Conclusion

In this research, we tried to alleviate the sorts of difficulties that have been ailing multi-objective optimization of high-dimensional problems. Lack of selection pressure propels optimizers to modify selection operators accordingly. These modifications usually lead to loss of diversity in the population which is not desirable and considered as a low performance. Here we proposed a diversity measure and a diversity control mechanism which can be used in combination to remedy this problem.

To control the diversity of population, we used an adaptive framework which changes the selection operator according to changes in the diversity of the population. By an increase in the population diversity, the selection operator will change to increase the selection pressure and by a decrease in diversity the selection pressure will decline. In order to find the variation of diversity this paper proposed a diversity measure based on shortest Hamiltonian path to monitor the diversity variation in each generation.

With the intention of assessing the proposed method, we incorporated it into MOEA/D a well know multi-objective evolutionary algorithm. The only diversity control facility used in MOEA/D is the initial decomposition of the multi-objective optimization problem into a set of uniformly distributed sub-problems. In high dimensions, this decomposition is not coarse enough so it will not be as effective as expected. We exchanged the MOEA/D selection mechanism with our devised operator and used this new MOEA/D algorithm in the experiments. For experiments we used WFG problem family which are the most comprehensive and recent group of test problems in the literature. To evaluate and compare the obtained results two performance measures were used, IGD and hypervolume. Applying both methods to concave and convex problems revealed a contradiction between the two performance metrics. While the modified MOEA/D performed better in terms of IGD in convex problems, the original MOEA/D performed better in terms of Hypervolume. In order to ascertain which performance measure is right, we provided two types of solution sets with known diversity contrast and proved that IGD yields more accurate comparison. Consequently, our method is of a valuable potential and can be extended and refined to perform even better.

References

1. Deb K (2001) Multi-objective optimization using evolutionary algorithms, vol 16. Wiley
2. Garza-Fabre M, Toscano-Pulido G, Coello Coello C, Rodriguez-Tello E (2011) Effective ranking+ speciation= many-objective optimization. In: 2011 IEEE Congress on Evolutionary Computation (CEC). IEEE, pp 2115–2122
3. Schaffer JD (1985) Multiple objective optimization with vector evaluated genetic algorithms. In: proceedings of the 1st international Conference on Genetic Algorithms. L. Erlbaum Associates Inc., pp 93–100
4. Fonseca CM, Fleming PJ (1993, June) Genetic algorithms for multiobjective optimization: Formulationdiscussion and generalization. In: ICGA, vol 93, pp 416–423
5. Horn J, Nafpliotis N, Goldberg DE (1994) A niched Pareto genetic algorithm for multiobjective optimization. In: Proceedings of the First IEEE Conference on Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence. IEEE, pp 82–87
6. Ishibuchi H, Tsukamoto N, Nojima Y (2008) Evolutionary many-objective optimization: a short review. In: IEEE congress on evolutionary computation, pp 2419–2426
7. Ishibuchi H, Akedo N, Ohyanagi H, Nojima Y (2011) Behavior of EMO algorithms on many-objective optimization problems with correlated objectives. In: 2011 IEEE Congress on In Evolutionary Computation (CEC). IEEE, pp 1465–1472
8. Deb K, Jain H (2012) Handling many-objective problems using an improved NSGA-II procedure. In: 2012 IEEE congress on Evolutionary computation (CEC). IEEE, pp 1–8
9. Padhye N, Mittal P, Deb K (2015) Feasibility preserving constraint-handling strategies for real parameter evolutionary optimization. Comput Optim Appl 62.3:851–890
10. Chen B, Lin Y, Zeng W, Zhang D, Si YW (2015) Modified differential evolution algorithm using a new diversity maintenance strategy for multi-objective optimization problems. Appl Intell 43(1):49–73
11. Zhang Q, Li H (2007) MOEA/D: A multiobjective evolutionary algorithm based on decomposition. IEEE Trans Evol Comput 11(6):712–731
12. Das I, Dennis JE (1998) Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. SIAM J Optim 8(3):631–657
13. Li H, Zhang Q (2009) Multiobjective optimization problems with complicated Pareto sets, MOEA/d and NSGA-II. IEEE Trans Evol Comput 13(2):284–302
14. Zhang Q, Liu W, Li H (2009) The performance of a new version of MOEA/d on CEC09 unconstrained MOP test instances. In: IEEE Congress on Evolutionary Computation, vol 1, pp 203–208
15. Gong M, Liu F, Zhang W, Jiao L, Zhang Q (2011) Interactive MOEA/D for multi-objective decision making. In: Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation. ACM, pp 721-728
16. Deb K, Jain H (2014) An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting

approach, part I: Solving problems with box constraints. IEEE Trans Evol Comput 18(4):577–601

17. Yuan Y, Xu H, Wang B (2014) An improved NSGA-III procedure for evolutionary many-objective optimization. In: Proceedings of the 2014 Conference on Genetic and Evolutionary Computation. ACM, pp 661-668
18. Beume N, Naujoks B, Emmerich M (2007) SMS-EMOA: Multiobjective Selection based on dominated hypervolume. Eur J Oper Res 181(3):1653–1669
19. Bader J, Zitzler E (2011) Hype: An algorithm for fast hypervolume-based many-objective optimization. Evol Comput 19(1):45–76
20. Jiang S, Ong YS, Zhang J, Feng L (2014) Consistencies and contradictions of performance metrics in multiobjective optimization. IEEE Trans Cybern 44(12):2391–2404
21. Van Veldhuizen DA (1999) Multiobjective evolutionary algorithms: classifications, analyses, and new innovations (No. AFIT/DS/ENG/99-01). Air Force Institute of Tech Wright-Pattersonafb oh School of Engineering
22. Czyzżk P, Jaszkiewicz A (1998) Pareto simulated annealing–a metaheuristic technique for multiple-objective combinatorial optimization. J Multi-Criteria Decis Anal 7(1):34–47
23. Hansen MP, Jaszkiewicz A (1998) Evaluating the quality of approximations to the non-dominated set. IMM, Department of Mathematical Modelling, Technical Universityof Denmark
24. Van Veldhuizen DA, Lamont GB (1998) Evolutionary computation and convergence to a pareto front. In: Late Breaking Papers at the Genetic Programming 1998 Conference. Stanford University, pp 221–228
25. Van Veldhuizen DA, Lamont GB (1999) Multiobjective evolutionary algorithm test suites. In: Proceedings of the 1999 ACM Symposium on Applied Computing. ACM, pp 351-357
26. Zitzler E, Thiele L, Laumanns M, Fonseca CM, Da Fonseca VG (2003) Performance assessment of multiobjective optimizers: an analysis and review. IEEE Trans Evol Comput 7(2):117–132
27. Zitzler E, Thiele L (1998) Multiobjective optimization using evolutionary algorithms–a comparative case study. In: Parallel Problem Solving from Nature–PPSN V. Springer Berlin Heidelberg, pp 292–301
28. Zitzler E, Thiele L (1999) Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. IEEE Trans Evol Comput 3(4):257–271
29. Li H, Zhang Q (2009) Multiobjective optimization problems with complicated Pareto sets, MOEA/d and NSGA-II. IEEE Trans Evol Comput 13(2):284–302
30. Deb K, Agrawal S, Pratap A, Meyarivan T (2000) A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In: Parallel problem solving from nature PPSN VI. Springer Berlin Heidelberg, pp 849–858
31. Zitzler E, Deb K, Thiele L (2000) Comparison of multiobjective evolutionary algorithms: Empirical results. Evol Comput 8(2):173–195
32. Tan KC, Lee TH, Khor EF (2002) Evolutionary algorithms for multi-objective optimization: performance assessments and comparisons. Artif Intell Rev 17(4):251–290
33. Schott J. R (1995) Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization (No. AFIT/CI/CIA-95-039). Air Force Institute of Technology Wright-Patterson AFB OH
34. Wu J, Azarm S (2001) Metrics for quality assessment of a multiobjective design optimization solution set. J Mech Des 123(1):18–25
35. Deb K, Pratap A, Agarwal S, Meyarivan TAMT (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6(2):182–197
36. Zhou A, Jin Y, Zhang Q, Sendhoff B, Tsang E (2006) Combining model-based and genetics-based offspring generation for multi-objective optimization using a convergence criterion. In: IEEE Congress on Evolutionary Computation, 2006. CEC 2006. IEEE, pp 892–899
37. Li M, Yang S, Liu X, Shen R (2013) A comparative study on evolutionary algorithms for many-objective optimization. In: Evolutionary Multi-Criterion Optimization. Springer Berlin Heidelberg, pp 261-275
38. Mostaghim S, Teich J (2005) A new approach on many objective diversity measurement. In: Dagstuhl Seminar Proceedings. Schloss Dagstuhl-Leibniz-Zentrum für Informatik
39. Farhang-Mehr A, Azarm S (2002) Diversity assessment of Pareto optimal solution sets: an entropy approach. In: WCCI. IEEE, pp 723-728
40. Petrie A, Willemain TR (2010) The snake for visualizing and for counting clusters in multivariate data. Stat Anal Data Min 3(4):236–252
41. Johnson DS, McGeoch LA (1997) The traveling salesman problem: a case study in local optimization. Local Search Comb Optim 1:215–310
42. Petrie A, Willemain TR (2013) An empirical study of tests for uniformity in multidimensional data. Comput Stat Data Anal 64:253–268
43. Huband S, Hingston P, Barone L, While L (2006) A review of multiobjective test problems and a scalable test problem toolkit. IEEE Trans Evol Comput 10(5):477–506
44. Durillo JJ, Nebro AJ, Alba E (2010) The jmetal framework for multi-objective optimization: Design and architecture. In: 2010 IEEE Congress on Evolutionary Computation (CEC). IEEE, pp 1-8
45. Durillo JJ, Nebro AJ, Luna F, Dorronsoro B, Alba E (2006) JMetal: A java framework for developing multi-objective optimization metaheuristics. Departamento de Lenguajes y Ciencias de la Computación, University of Málaga, ETSI Informática, Campus de Teatinos, Technical Report ITI-2006-10
46. Conover WJ (1980) Practical nonparametric statistics
47. Mostaghim S, Teich J (2003) Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO). In: Proceedings of the 2003 IEEE Swarm Intelligence Symposium, 2003. SIS'03. IEEE, pp 26–33
48. Padhye N, Branke J, Mostaghim S (2009) Empirical comparison of mopso methods-guide selection and diversity preservation. In: IEEE Congress on Evolutionary Computation, 2009. CEC'09. IEEE, pp 2516–2523
49. Kukkonen S, Lampinen J (2005) GDE3: The third evolution step of generalized differential evolution. In: 2005. The 2005 IEEE Congress on Evolutionary Computation, vol 1. IEEE, pp 443–450
50. Nebro AJ, Durillo JJ, Garcia-Nieto J, Coello C, Luna F, Alba E (2009) Smpso: A new pso-based metaheuristic for multi-objective optimization. In: IEEE symposium on Computational intelligence in miulti-criteria decision-making, 2009. mcdm'09. IEEE, pp 66–73
51. Durillo JJ, García-Nieto J, Nebro AJ, Coello CAC, Luna F, Alba E (2009) Multi-objective particle swarm optimizers: An experimental comparison. In: Evolutionary Multi-Criterion Optimization. Springer Berlin Heidelberg, pp 495–509

Kimia Bazargan Lari University in Iran (2016). She received her artificial intelligence from Shiraz University in Iran in has been a faculty member of computer engineering in Apadana University from 2013. research domains include evolutionary computation, optimization algorithms and swarm intelligence.

Ali Hamzeh received his B.S. and M.Sc. degrees in computer engineering and artificial intelligence from Shiraz University in Iran. Then he received his Ph.D. degrees in artificial intelligence from Iran University of Science and Technology, 2007. He has been an assi intelligence at Shiraz University from 2007. His research interests include evolutionary computation, machine learning and social networks. Also, he has published over 90 peer areas.