

Regression line shifting mechanism for analyzing evolutionary optimization algorithms

Anupam Biswas¹ · Bhaskar Biswas¹

Published online: 21 September 2016
© Springer-Verlag Berlin Heidelberg 2016

Abstract This work introduces a novel methodology to perform the comparative analysis of evolutionary optimization algorithms. The methodology relies simply on linear regression and quantile–quantile plots. The methodology is extrapolated as the one-to-one comparison, one-to-many comparison and many-to-many comparison of solution quality and convergence rate. Most of the existing approaches utilize both solution quality and convergence rate to perform comparative analysis. However, many-to-many comparison, i.e., ranking of algorithms is done only in terms of solution quality. The proposed method is capable of ranking algorithms in terms of both solution quality and convergence rate. Method is analyzed with well-established algorithms and real data obtained from 25 benchmark functions.

Keywords Evolutionary optimization algorithms · Linear regression · Particle swarm optimization · Differential evolution · Visual analysis

1 Introduction

Evolutionary optimization algorithms (EOAs) are heuristic stochastic processes. EOAs generate next state in each successive step depending on the current state, and the process continues until certain predefined conditions are met. Numerous algorithms have been developed in last decades following

same iterative notion. Most of these algorithms are developed for optimization purpose. Although the notion of these algorithms is similar and designed for the same problem, strategies adopted for generating next states are different. No free lunch theorem (Wolpert and Macready 1997) states that it is not possible to find one algorithm being better for any problem. It has also been noticed that for the same problem, algorithms behave in different ways depending on the degree of knowledge about problem (García et al. 2009). Hence, it is essential to know when algorithm behaves better and which algorithm performs better. With the rapid growth of applications of EOAs as well as algorithms in several application domains, comparative analysis of EOAs has drawn attention in recent years.

Numerous approaches have been developed for evaluating the performance of EOAs. Nonparametric approaches (García et al. 2007; Moreno-Pérez et al. 2007; Derrac et al. 2011) are very popular and widely used to evaluate the performance of EOAs. These approaches simply generate different parameters to solutions obtained over multiple executions of an algorithm, and performances are estimated in terms of those parameters. Rojas et al. (2002) and Czarn et al. (2004) have proposed parametric approaches that infer parameters from the probability distribution of solutions. Shilane et al. (2008) has extended these approaches further for analyzing performance specific to any parameter. Francois and Lavergne (2001) has introduced a statistical methodology to choose efficient parameter settings. The analysis method of EOAs has further grown in several directions such as bootstrapping (Nijssen and Back 2003; Carrano et al. 2008), exploratory landscape analysis (ELA) (Mersmann et al. 2010) and drift analysis (He and Yao 2001). Mathematical approaches (Muhlenbein and Mahnig 2001; Yang 2011; Lockett 2013; He and Chen 2013) and visual analysis approach (Wu et al. 1999; Lutton and Fekete 2011) have drawn attention for

Communicated by A. Di Nola.

✉ Anupam Biswas
anupam.rs.cse13@iitbhu.ac.in

Bhaskar Biswas
bhaskar.cse@iitbhu.ac.in

¹ Department of Computer Science and Engineering, Indian Institute of Technology (BHU), Varanasi, India

evaluating EOAs in recent years. In this paper, a simple mechanism has been developed based on linear regression analysis and quantile–quantile plot to design a methodology for analyzing EOAs visually.

Although many approaches have been developed over the decade, parametric and nonparametric approaches are still treated to be more effective and utilized immensely for analysis of EOAs. Most of these approaches incorporate results obtained in perspective of benchmark functions or some specific problems. Basic statistical measures such as standard deviation, mean, median, minimum and maximum are estimated to determine overall performance. We have proposed a nonparametric approach in which interpretation of parameters is done through visual inspection. Proposed approach furnishes more direct interpretation of original solutions with quantile–quantile plot and simple linear regression analysis. With proposed approach, the performance of EOAs is analyzed in terms of both solution quality and convergence rate. Method has been examined in directives such as one-to-one comparison, one-to-many comparison, and for ranking many-to-many comparison.

A preliminary report of parts of this study was published as a conference paper (Biswas and Biswas 2014). The current version of paper significantly extends earlier work as well as results with more in-depth discussions. The methodology is also extended for convergence rate and analyzed with additional benchmark objective functions.

Rest of the paper is organized as follows: Sect. 2 explains preliminary concepts related to proposed approach with suitable example, Sect. 3 elaborates regression line shifting mechanism and derives important properties, Sect. 4 describes comparative analysis methodology of solution quality and convergence in detail with regression line shifting mechanism, Sect. 5 studies proposed analysis method with 25 benchmark functions and provides thorough analysis with detailed experimental setups, and finally, Sect. 6 concludes with remarks about the advantages and drawbacks of proposed method.

2 Preliminaries

In this section, various background topics such as linear regression analysis, regression line and quantile plot are briefed with suitable examples. Readers already familiar with these topics may skip this section, whereas we suggest novice readers to follow this section to better understand proposed methodology.

2.1 Linear regression

Linear regression attempts to establish the relationship between dependent variable and one or more explanatory

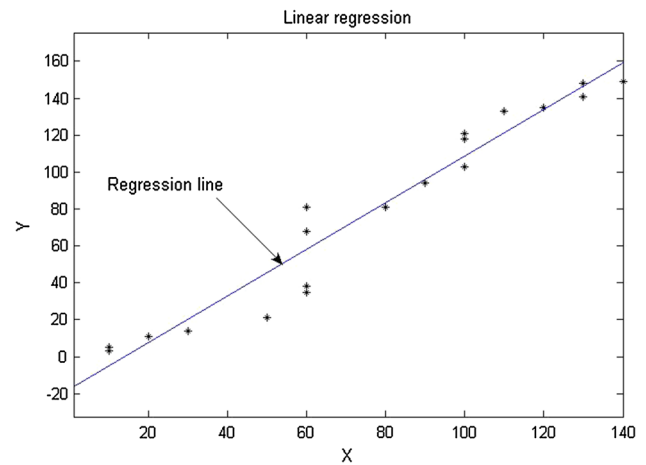


Fig. 1 Linear regression example. A best-fitted line is drawn through the given data points

variables. Involvement of single explanatory variable is referred as simple linear regression (SLR), otherwise referred as multiple linear regression. In this paper, we will deal only with SLR. In SLR, a best-fitting straight line called regression line is derived from the set of points as shown in Fig. 1.

Definition 1 (*Regression line (RL)*) A linear regression line or simply regression line has an equation of the form $Y = mX + c$, where X is the explanatory variable and Y is the dependent variable. The gradient of line and intercept, i.e., m and c , respectively, are computed during SLR analysis of variables X and Y .

Vertical distances between best-fitted line and the points of the data set are referred as residuals. SLR attempts to find a regression line such that sum of squared residuals is minimum. Thus, SLR is nothing but an optimization process that minimizes the sum of squared residuals. This regression line is the representative line of the set of points from where it is being derived. The corresponding value of the dependent variable for any value of the explanatory variable is predictable by referring regression line.

2.2 Quantile plot

A quantile plot is a very simple and effective way of displaying univariate data graphically. It displays all the data for a given attribute. First, the data set is sorted, and then data points are plotted against their respective quantile information. Quantiles are the points taken at a regular interval from the data set. Consider that data set $X = \{x_1, x_2, x_3, \dots, x_N\}$ is sorted in non-decreasing order. Each observation x_i is coupled with a percentage f_i (referred as f -value), which implies approximately $100 \times f_i$ % of data are below x_i . The f -value of i th observation is computed as follows:

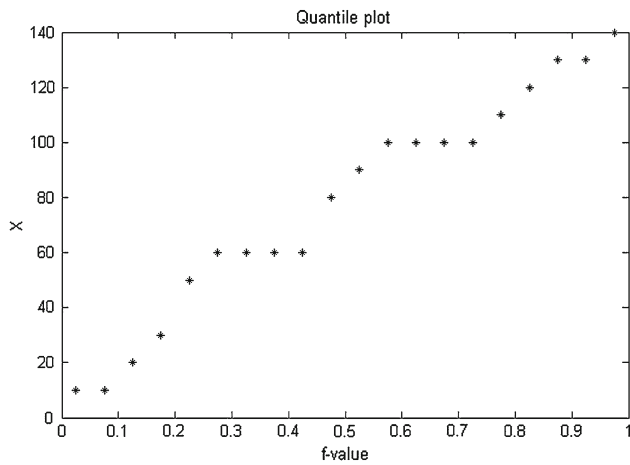


Fig. 2 Quantile plot example. Every value of X is plotted against their corresponding f -values

$$f_i = \frac{(i - 0.5)}{N} \tag{1}$$

where f_i is ranged from $\frac{1}{2N}$ (which is slightly above 0) to $1 - \frac{1}{2N}$ (which is slightly below 1) and it increases with equal step size $\frac{1}{N}$. These f -values are the quantiles corresponding to each observation x_i . In quantile plot, a graph is plotted with x_i versus f_i . An example of quantile plot is shown in Fig. 2.

2.3 Quantile–quantile plot

In the quantile–quantile plot, quantiles of two univariate data sets are plotted one against another. The dominance of one data set over another is visualized with this plot. Unlike quantile plot discussed above, quantile–quantile plot consists of two quantile plots, one with respect to the x -axis and another one with respect to the y -axis. Let $X = \{x_1, x_2, x_3, \dots, x_N\}$ and $Y = \{y_1, y_2, y_3, \dots, y_M\}$ are two data sets, and both are sorted in non-decreasing order. Quantiles of data sets X and Y are plotted one against another. Any point (x_i, y_i) on the plot corresponds to quantile f_i of first data set X (i.e., x -coordinate) against same quantile of second data set Y (i.e., y -coordinate). If $M = N$, then plotting is simply x_i versus y_i . If $(M < N)$ or $(M > N)$ then plotting is little bit difficult. In this case, y_i is plotted against x_i till satisfies $(M < N)$ or $(M > N)$. Remaining observations of the larger data set are plotted through interpolation. In this paper, we will deal only with the first situation, i.e., equal-sized data sets. An example of the quantile–quantile plot is shown in Fig. 3. It is clear from plotting that, for given data sets X and Y , initial portion is dominated by Y , whereas rest of portion is dominated by X . These observations are even more clear with the visualization of plotting in reference to $X = Y$ line.

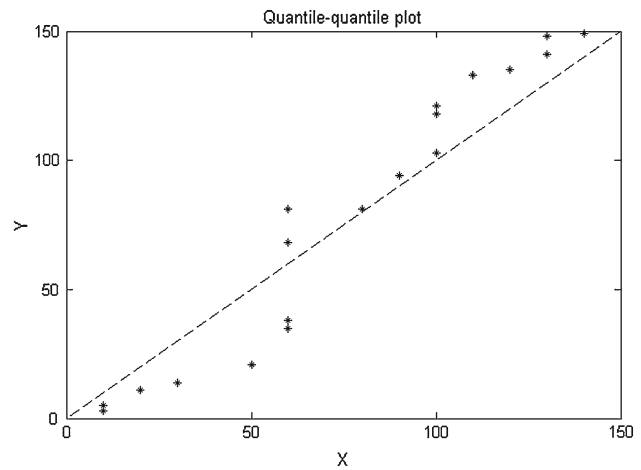


Fig. 3 Quantile–quantile plot example. Data sets X and Y are of equal size. Quantiles of X and Y are plotted against each other

3 Regression line shifting

In this section, point dominance and line dominance of two data sets are discussed. Shifting mechanism of the regression line is discussed with suitable examples, particularly positive shift and negative shift. Some important properties are derived.

3.1 Point dominance

The dominance of a data set defines how good are the data values of the data set in comparison with another data set. Evaluation of dominance of a data set over other is a complex task. Most data values of the set are good with respect to some portion but bad for remaining portion of data values of other data sets. Quantile–quantile plot eases evaluation of dominance of one data set over another set as discussed in the previous section. We refer data set whose dominance is to be evaluated as *actor* and data set with respect to whom dominance is evaluated as *competitor*. Let $D_a = \{a_i | i = 1, 2, 3, \dots, r\}$ and $D_c = \{c_i | i = 1, 2, 3, \dots, r\}$ are actor data set and competitor data set, respectively, sorted in non-decreasing order. We consider equal-sized actor data set and competitor data set, so quantile–quantile plot is simply a plot of D_a versus D_c . We specify x -axis for plotting actor and y -axis for plotting competitor. With these prespecifications, both actor and competitor sets are presented in the quantile–quantile plot as shown in Fig. 4. Each point (a_i, c_i) in plot is a pair of $a_i \in D_a$ and $c_i \in D_c$. We define point dominance at any point (a_i, c_i) in reference to neutral line.

Definition 2 (Neutral line (NL)) A line $Y = mX + c$ is referred as NL if it goes through the origin (i.e., intercept $c = 0$) and gradient $m = 1$. The line is neutral in the sense

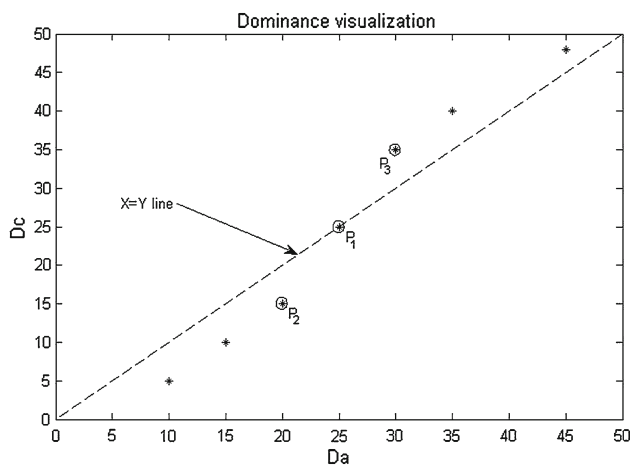


Fig. 4 An illustrative example of point dominance. Data sets of comparing algorithm (D_a) and competitor algorithm (D_c) are plotted in quantile–quantile plot. Three points P_1 , P_2 and P_3 are considered as on, below and above the $X = Y$ line, respectively

that at every point (x, y) in the NL, $x = y$. Thus, simply we say $X = Y$ line is NL.

We interpret smaller value implies higher dominance because this work deals with minimization problems during result analysis. In reference to NL, we define three kinds of possible dominance at any point (a_i, c_i) in plot as follows:

Non-dominance: Dominance of actor set over competitor set at any point (a_i, c_i) that lies on the NL is referred as non-dominance since $a_i = c_i$, i.e., neither a_i dominates c_i nor c_i dominates a_i .

Actor-dominance: Dominance of actor set over competitor set at any point (a_i, c_i) that lies below the NL is referred as actor-dominance since $a_i < c_i$, i.e., a_i dominates c_i .

Competitor-dominance: Dominance of actor set over competitor set at any point (a_i, c_i) that lies above the NL is referred as competitor-dominance since $a_i > c_i$, i.e., c_i dominates a_i .

In Fig. 4, points P_1 , P_2 and P_3 are examples of non-dominance, actor-dominance and competitor-dominance, respectively. Note that dominance at any point (a_i, c_i) is nothing but operator that fits the comparison between a_i and c_i . Thus, we have three operators =, < and > for defining dominance on quantile–quantile. Based on the dominance of different points on quantile–quantile plot following situations may arise for actor set and competitor set. Suppose, all points in quantile–quantile plots of D_a and D_c are non-dominated, i.e., we have condition $a_i = c_i, \forall a_i \in D_a$ and $\forall c_i \in D_c$. If initial consecutive points are competitor-dominated, while remaining points are non-dominated then we have condition $a_i > c_i, i = 1, 2, \dots, m$ and $a_i =$

$c_i, i = m + 1, m + 2, \dots, r$. In first case, points are summarized in a single partition comprising all non-dominated points. However, in second case, points are partitioned into two sets one for competitor-dominated and another one for non-dominated points. In this way, we can have multiple conditions depending on partitioning of points with respect to operators (=, <, >) in quantile–quantile plot.

Proposition 1 With three operators (=, <, >) and k partitions 3^k number of conditions are possible.

Proof By permutations with repetition. \square

Proposition 2 With three operators (=, <, >) and $k > 1$ partitions $\frac{(3^k+3)}{2}$ number of distinct conditions are possible.

Proof For $k = 1$

1. $a_i = c_i, \forall a_i \in D_a$ and $\forall c_i \in D_c$
2. $a_i < c_i, \forall a_i \in D_a$ and $\forall c_i \in D_c$
3. $a_i > c_i, \forall a_i \in D_a$ and $\forall c_i \in D_c$

For $k = 2$

1. $a_i = c_i, i = 1, 2, \dots, m$ and $a_i < c_i, i = m + 1, m + 2, \dots, r$
2. $a_i = c_i, i = 1, 2, \dots, m$ and $a_i > c_i, i = m + 1, m + 2, \dots, r$
3. $a_i < c_i, i = 1, 2, \dots, m$ and $a_i = c_i, i = m + 1, m + 2, \dots, r$
4. $a_i > c_i, i = 1, 2, \dots, m$ and $a_i = c_i, i = m + 1, m + 2, \dots, r$
5. $a_i < c_i, i = 1, 2, \dots, m$ and $a_i > c_i, i = m + 1, m + 2, \dots, r$
6. $a_i > c_i, i = 1, 2, \dots, m$ and $a_i < c_i, i = m + 1, m + 2, \dots, r$
7. $a_i = c_i, i = 1, 2, \dots, m$ and $a_i = c_i, i = m + 1, m + 2, \dots, r$
8. $a_i < c_i, i = 1, 2, \dots, m$ and $a_i < c_i, i = m + 1, m + 2, \dots, r$
9. $a_i > c_i, i = 1, 2, \dots, m$ and $a_i > c_i, i = m + 1, m + 2, \dots, r$

In the above situation, although for two partitions we can have 3^2 conditions, but note that conditions 7, 8 and 9 actually represent the same situations as in the case of $k = 1$, i.e., the conditions 1, 2 and 3, respectively. Hence, we actually have $3^2 - 3 = 6$ conditions out of 9 conditions. In this way, we can have $3^k - (3^{(k-1)} + 3^{(k-2)} + \dots + 3^1) = \frac{1}{2} \left\{ 3^k - (3 \times (3^{(k-1)} - 1)) \right\} = \frac{(3^k+3)}{2}$ conditions for k possible partitions, where $k > 1$. \square

As shown in Propositions 1 and 2, we have to deal with several conditions that arise with the partitioning of points based

on their dominance in quantile–quantile plot. Therefore, evaluation of overall dominance of a data set over other becomes complicated with quantile–quantile plot since we have to accumulate meaning of all conditions to express overall dominance of data set. In following subsections, mechanisms for accumulating the meaning of various conditions, those arise with the partitioning of points in the quantile–quantile plot, are discussed.

3.2 Regression line shifting and dominance

Simple linear regression (SLR) analysis derives a linear relationship between two variables of the bivariate data set. Regression line (RL) is the representative for the relationship drawn between two variables of data set with SLR. Point dominance discussed in previous subsection incorporates two data sets, where actor and competitor data sets are presented in a quantile–quantile plot. However, the accumulation of overall dominance was difficult due to various conditions that arise with the partitioning of points in the quantile–quantile plot. SLR analysis on quantiles of data points in quantile–quantile plot will accumulate all the point dominance by simply deriving a linear relationship. The SLR analysis on quantiles is different from ordinary SLR analysis on scatter plot explained in Sect. 2. In case of SLR analysis on scatter plot, one of the two distinct variables of a single bivariate data set is considered as explanatory variable and other one as dependent variable. Corresponding data values for both the variables are considered directly during SLR analysis as they appeared in the data set. On contrary, in case of SLR analysis on quantile–quantile plot, two univariate data sets of same variable are considered. Actor data set and competitor data set are considered as explanatory and dependent variable, respectively. Data values are considered with respect to quantiles of each variable during SLR analysis.

The RL obtained with SLR analysis on quantile–quantile plot preserves overall dominance of actor data set over competitor data set. If point dominance changes, i.e., dominance of some points changes from one dominance to another (e.g., actor-dominance to competitor-dominance) then accordingly RL also shifts to new position. The shifting mechanism of RL depending on various conditions appeared in the points of quantile–quantile plot are discussed as follows

For instance, if the partitioning implies condition 1 for $k = 1$, i.e., $a_i = c_i, \forall a_i \in D_a$ and $\forall c_i \in D_c$. At condition 1, all points lie on the NL so the RL also lies on NL. Similarly, at condition 3 for $k = 1$, all values of D_a are dominated by D_c , so RL appears below and parallel to the NL. Suppose, first m values of D_a and D_c are changed in such a way that $\{a_i \leq c_i, i = 1, 2, \dots, m\}$. It means that condition 1 and condition 3 for $k = 1$ changed to condition 3 for $k = 2$, i.e., $\{a_i < c_i, i = 1, 2, \dots, m$ and $a_i = c_i, i = m + 1, m + 2, \dots, r\}$ and condition 5 for $k = 2$, i.e., $\{a_i < c_i, i = 1, 2, \dots, m$

and $a_i > c_i, i = m + 1, m + 2, \dots, r\}$, respectively. New conditions imply the portion from the beginning to m to lie on or above the NL, i.e., the new RL no longer remains parallel to NL. Initial portion of the RL shifts toward NL, and it intersects somewhere at the starting portion. This shifting follows a clockwise rotation of the RL with respect to previous position of the RL. If such kinds of changing in D_a and D_c continue further up to last point (i.e., $m = r$), the RL also rotates with respect to previous position, and it shifts more toward the NL. Various instances of RL with incremental changing in the portions of D_a and D_c are shown in Fig. 5. One can notice that with rotational shifting of RL, intersecting point on NL also moves ahead toward end. When $m = r$, i.e., becomes condition 2, the RL again becomes parallel to NL as well as above the NL. Hence at any point, the dominance relation of D_a and D_c is determined by the angle and position of intersecting point between RL and NL. Depending on dominance of actor data set over competitor data set, we divide the shifting of RL into two categories. If the shifting of RL implies increment in dominance of actor data set over competitor data set, we refer it as positive shift, and otherwise, we refer it as negative shift.

3.2.1 Positive shift

The positive shifting of RL achieved in two ways as follows. Changing condition 3 for $k = 1$ to either condition 2 or condition 5 for $k = 2$ as explained above actually implies improvement in D_a so as the dominance of actor data set. If the change continued up to r , the improvement in D_a becomes worst to best. During the changes in D_a and D_c from worst dominance of actor data set to best dominance, the RL corresponding to that situation also undergoes clockwise rotational shift toward the NL. This rotational positive shift is with respect to the last point in the clockwise direction, so it is referred as clockwise positive shift (CPS). Similarly with respect to starting point, we will also have positive shift if we change values from last point to m such that $\{a_i \leq c_i, i = m, m + 1, \dots, r\}$, i.e., changed to either condition 4 or 6 for $k = 2$. In this case, the positive shifting of RL is in the anticlockwise direction, and hence, it is referred as anticlockwise positive shift (APS). The CPS and APS of different instances of RLs are presented in Figs. 5 and 6.

3.2.2 Negative shift

Similar to positive shift we also have negative shift if we do similar changing on condition 2 for $k = 1$ in D_a and D_c . The condition 2 for $k = 1$ means all values of D_a are dominated by D_c , so the RL appears above and parallel to the NL. However, in the context of minimization problems, actually all values of D_c are dominated by D_a , which means dominance of actor data set over competitor data set is in the

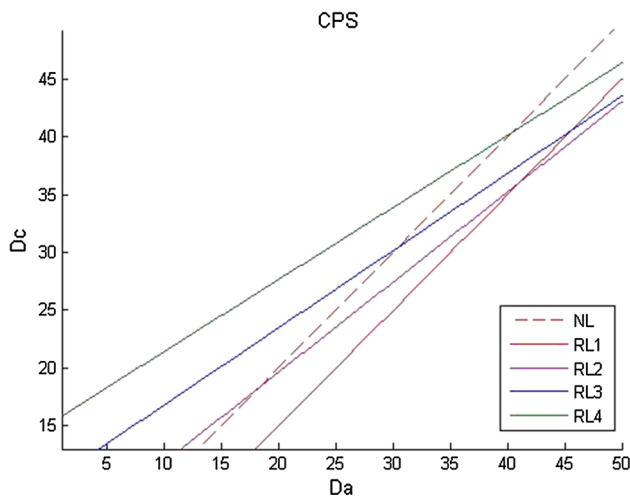


Fig. 5 Clockwise positive shift. Here, NL is the neutral line. RL1, RL2, RL3 and RL4 are the various instances of regression line after successive incremental modification of first m values of D_a and D_c

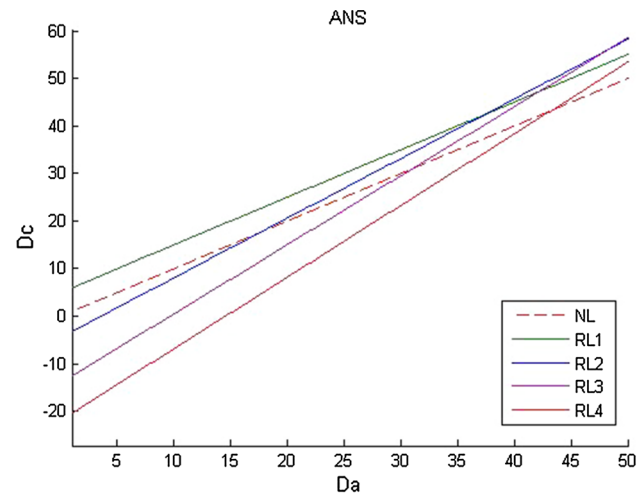


Fig. 7 Anticlockwise negative shift. Here, NL is the neutral line. RL1, RL2, RL3 and RL4 are the various instances of regression line after successive incremental modification of first m values of D_a and D_c

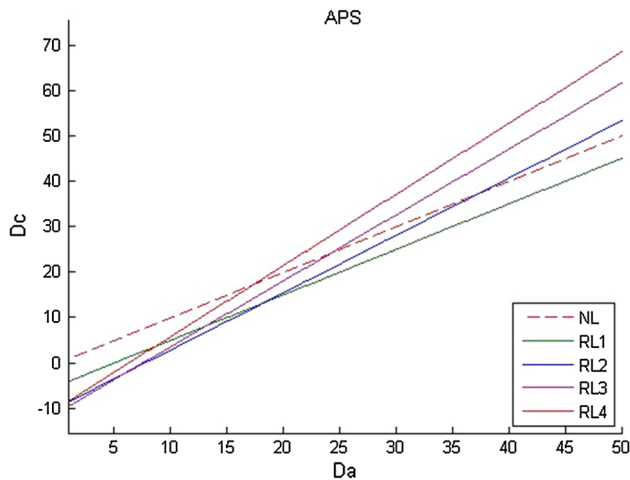


Fig. 6 Anticlockwise positive shift. Here, NL is the neutral line. RL1, RL2, RL3 and RL4 are the various instances of regression line after successive incremental modification of last m values of D_a and D_c

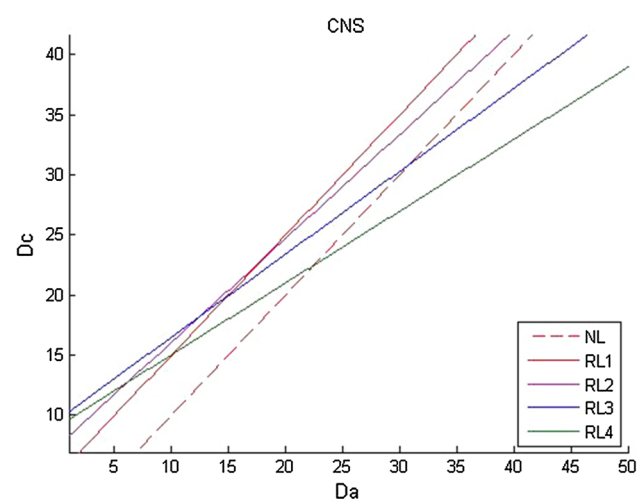


Fig. 8 Clockwise negative shift. Here, NL is the neutral line. RL1, RL2, RL3 and RL4 are the various instances of regression line after successive incremental modification of last m values of D_a and D_c

best situation. If we change first m values of D_a and D_c to become either condition 1 or condition 6 for $k = 2$, it results in anticlockwise rotational shifting of RL as shown in Fig. 7. Similarly, if we change last m values of D_a and D_c then condition 2 for $k = 1$ changes to either condition 3 or condition 5 for $k = 2$, which results in clockwise rotational shifting of RL as shown in Fig. 8. From the perspective of minimization problem, these shifts indicate degradation in dominance of actor data set. Therefore, these shifts are referred as negative shift. The clockwise and anticlockwise rotational shift that results negative effect is referred as clockwise negative shift (CNS) and anticlockwise negative shift (ANS), respectively.

Theorem 1 For minimization problems, the CPS or CNS can always have greater than 315° angle between RL and NL.

Proof Let us consider one RL that has the angle with the NL less than 315° for the purpose of contradiction. Note that when the angle between RL and NL is 315° , at this time RL is parallel to y-axis as shown in Fig. 9. Now, consider two points p and q in the RL, above and below the intersecting point on NL. It is clear that for these two points, the conditions $\{a_p < c_p\}$, $\{a_q > c_p\}$, $\{c_p > c_q\}$ and $\{a_p < a_q\}$ are satisfied. We have plotted D_a and D_c which are sorted in increasing order. Therefore, the condition $c_p > c_q$ contradicts, as in an increasing ordered list $D_c = c_1, c_2, \dots, c_r$ it is not possible to have an ordered pair (c_i, c_j) such that $c_i > c_j$ and $i < j$. The

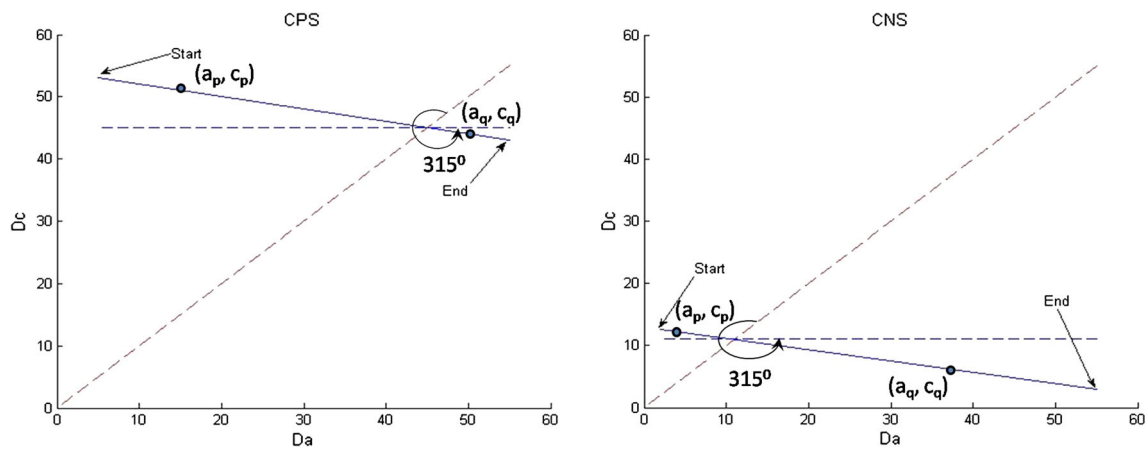


Fig. 9 Angular displacement in CPS and CNS. The *Start* and *End* indicate the starting and ending point of regression line, respectively. Coordinates of points p and q are (a_p, c_p) and (a_q, c_q) , respectively

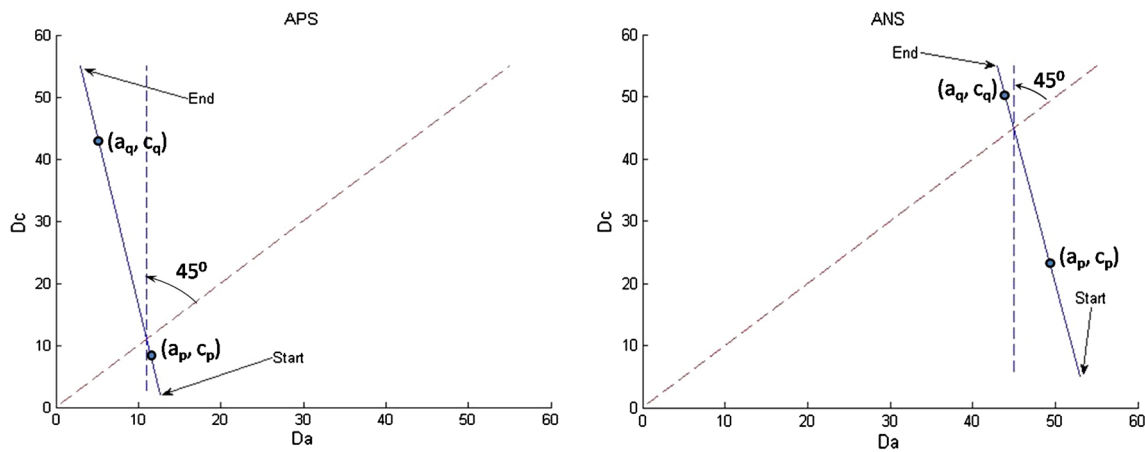


Fig. 10 Angular displacement in APS and ANS. *Start* and *End* indicates starting and ending point of regression line, respectively. Coordinates of points p and q are (a_p, c_p) and (a_q, c_q) , respectively

RL represents the dominance relationship between D_a and D_c . Hence, if any point in the RL has to satisfy $c_p > c_q$, it implies that there have to be some points (a_i, c_i) and (a_j, c_j) where we have a pair (c_i, c_j) such that $c_i > c_j$ and $i < j$, which is a contradiction. Hence, it is proved that for CPS or CNS angle between RL and NL is always greater than 315° . □

Theorem 2 For minimization problems, the APS or ANS can always have less than 45° angle between RL and NL.

Proof Let us consider a RL that has angle greater than 45° with NL. Note that when angle between RL and NL is 45° , at this time the RL is parallel to x -axis as shown in Fig. 10. Now, consider two point p and q in the RL below and above the intersecting point in NL. It is clear that for these two points, the conditions $\{a_p > c_p\}$, $\{a_q < c_p\}$, $\{c_p < c_q\}$ and $\{a_p > a_q\}$ are satisfied. Similar to Theorem 1, it can be shown that the condition $a_p > a_q$ contradicts. If any point

in the RL has to satisfy $a_p > a_q$, it implies that there have to be some points (a_i, c_i) and (a_j, c_j) where we have a pair (a_i, a_j) such that $a_i > a_j$ and $i < j$. Hence, it is proved that for APS or ANS angle between RL and NL angle less than 45° . □

Theorem 3 The maximum positive or negative rotational shift of RL is 90° , and rotational area covers half of 1st quadrant and half of 4th quadrant with respect to NL.

Proof By Theorem 1, it is clear that the RL can attain maximum angle of 360° up to minimum angle of 315° and takes place in 4th quadrant with respect to NL. Thus, maximum possible rotation would be $360^\circ - 315^\circ = 45^\circ$. Similarly from Theorem 2, it is also clear that maximum possible rotation would be 45° , i.e., in 1st quadrant, as the RL rotates from 0° to 45° with respect to NL. Hence, we have 45° in 1st and 45° in 4th quadrant totaling 90° shift of RL. □

4 The analysis methodology

This section elaborates the implication detail of regression line shifting mechanism with respect to NL for comparing performance of EOAs. Solution quality and convergence rate are the two major aspects in evaluation of performance of EOAs. Shifting mechanism is extrapolated to both solution quality- and convergence-based performance comparison. With the same notion as discussed above, we refer as *acting algorithm* whose performance has to be evaluated, and we refer as *competing algorithm* with respect to whom the performance of acting algorithm is evaluated. To evaluate performance of acting algorithm- over competing algorithm-based solution quality, best solutions generated by the algorithms in multiple trials are considered. However, for convergence analysis central tendency of population per iteration of the algorithm is considered. Dominance of acting algorithm over competing algorithms is expressed with clockwise shifting or anticlockwise shifting of RL with respect to NL. Different types of shifting and their significance in terms of performance of EOAs are discussed in following subsections.

4.1 Quality comparison

Solution quality of EOAs is evaluated in following three ways: first, the one-to-one comparison, where performance of acting algorithm is compared with only one competing algorithm at a time; second, the one-to-many comparison, where performance of acting algorithm is compared with multiple competing algorithms; at the last, many-to-many comparison, where ranking of acting algorithm is done based on performance with other competing algorithms.

4.1.1 One-to-one comparison

To compare performance of an acting algorithm A over competing algorithm C in terms of best solutions obtained over t trials are plotted in quantile–quantile plot. Assume that the objective function is a minimization problem. SLR analysis on quantile–quantile plot generates one RL with the methodology explained above. Note that we have single RL for this instance. We have already noticed that same instance can be attained through either positive shift or negative shift. Thus, it is unpredictable for a single instance of RL whether the instance is attained through positive shift or negative shift. However, with Theorem 2 we have anticlockwise shifting (i.e., APS or ANS) angle between RL and NL which is less than equal to 45° . Again, with Theorem 1 we have clockwise shifting (i.e., CPS or CNS) with angle which is greater than equal to 315° . Therefore, simply by visualizing the angle, it can be determined whether it is a clockwise shift or anticlockwise shift. From the perspective of performance,

if detected shift is clockwise then smaller angle indicates acting algorithm A performs better than competing algorithm C . If detected shift is anticlockwise then larger angle indicates better performance of A . Along with the angle between RL and NL, another significant aspect of visual inspection is the position of intersection RL on NL. For clockwise shifting, the intersecting point toward the end indicates better performance of A , while the intersecting point toward the beginning indicates better performance of A for anticlockwise shift.

4.1.2 One-to-many comparison

The one-to-many comparison plays important role in deciding whether any algorithm is better than all other algorithms. Particularly when any new algorithm is introduced, it has to be compared with existing state-of-the-art algorithms to prove its superiority. Let us consider, any new algorithm A , i.e., the acting algorithm whose performance has to be evaluated with respect to other n competing algorithms $C_1, C_2, C_3, \dots, C_n$. As discussed above, best solutions obtained over t trials on an objective function (minimization say) are plotted in quantile–quantile plot for each (A, C_i) pair and performed SLR analysis. Thus, we have n different RLs for all pairs of acting and competing algorithms. Each RL represents dominance of acting algorithm A over respective competing algorithm C_i . To decide on dominance of acting algorithm A over any competing algorithm C_i , the intersecting point of RL on NL and angle with NL is noted. Depending on the significance of these two observations for the RL (discussed earlier), the performance of the acting algorithm A with respect to any competing algorithm C_i is decided.

4.1.3 Many-to-many comparison

One challenging task is to compare performance of n algorithms with each other and rank them. It is challenging in the sense that performance of all algorithms varies with respect to other algorithms. One algorithm may perform better with respect to r algorithms and perform worst with respect to remaining $(n - r - 1)$ algorithms. In this way, we may obtain different sets of best-performing and worst-performing algorithms for each algorithm. Therefore, it is difficult to determine from the multiple possibilities that which of the algorithm exactly performs best or worst. Proposed methodology provides a very simplistic way to tackle this task.

The many-to-many comparison of n algorithms simply imply n one-to-many comparisons, where each of the algorithm is considered as acting algorithm, and remaining algorithms are competing algorithms. On the other hand, one-to-many comparison of n algorithms imply n one-to-one comparison. When one-to-one comparison is performed, the influence of competing algorithm in dominance relation-

ship of acting algorithm is actually determined through RL. In that sense, to get combined influence of all n competing algorithms the influence of all algorithms has to be added. Thus, one-to-many comparison of n algorithms reduced to single one-to-one comparison, where one algorithm is considered as acting algorithm, and remaining $n - 1$ algorithms are collectively considered as one single competing algorithm. Therefore, we have n one-to-one comparisons instead of one-to-many comparisons for many-to-many comparisons of n algorithms. For each of the n one-to-one comparisons, one of the n algorithms is considered as acting algorithm, and remaining $n - 1$ algorithms are collectively considered as one comparing algorithm. The collective influence of all $n - 1$ competing algorithms is obtained as follows. Let us consider, we have n algorithms $A_1, A_2, A_3, \dots, A_n$. Let A_m is the acting algorithm and all remaining $A_i, i \neq m$ algorithms are competing algorithm. Let we have the data sets $D = \{D_i | i = 1, 2, \dots, n\}$ comprising the sets of best solutions of all algorithms for t is the number of trials, where $D_i = \{d_{ij} | j = 1, 2, \dots, t\}$ arranged in non-decreasing order. Influence of all $n - 1$ competing algorithms with respect to acting algorithm A_m is computed as follows:

$$D_c = \left\{ \sum_{i=1 \& i \neq m}^{i=n} d_{ij} | j = 1, 2, \dots, t \right\} \quad (2)$$

The set D_c represents the collective influences of all $n - 1$ algorithms, and it is considered as a single competing algorithm. For the acting algorithm A_m we have the set of best solutions for t is the number of trials $D_m = \{d_{mj} | j = 1, 2, \dots, t\}$. The D_m and D_c are plotted in the quantile–quantile plot and with SLR analysis obtained one RL. Besides this RL for A_m , we will have $n - 1$ RLs for remaining $n - 1$ algorithms.

With this methodology, the RL of best-performing algorithm automatically acquires the highest position in terms of RL position. Similarly, the RL of the second best algorithm acquires the position below the best and so on up to the RL of the worst algorithm. The acquiring of RL positions depending on algorithms' performance can be understood with the following example. For instance, if algorithm A_m is the best algorithm that means most values in D_m will be dominated by the values of $D_i, i = \{1, 2, \dots, n \& i \neq m\}$. This means that all D_i contains larger values than D_m , and hence, their sum D_c will definitely contain the set of highest values. The values in D_c will produce the highest domination to D_m , attaining the highest possible position of RL.

4.2 Convergence comparison

Convergence rate is very crucial for an evolutionary optimization algorithm. Most of these algorithms are population

based. EOAs are stochastic in nature and follows unsupervised learning process. At each generation, the algorithm learns the solution space and improves the solutions of the population. Thus for any algorithm, it is expected that the population moves ahead toward the optimal solution by improving previous states. Overall improvement of the population is measured in terms of central tendency of the population. The central tendency of any data set is measured with the mean value or median value of the data set. Improvement in the mean value indicates the entire population is converging toward the optimal value. Often, the mean value of the population is plotted against each generation to visualize the rate at which the algorithm converges. Sometimes best value obtained in each generation also plotted against the generation to visualize convergence. Similar procedure is followed for comparing convergence of one algorithm with other competing algorithms. However, with the existing approaches, the dominance of EOAs in terms of convergence rate cannot be measured.

Similar to the quality comparison, convergence comparison is also divided into three types: one-to-one comparison, one-to-many comparison and many-to-many comparison. A more direct comparative analysis of convergence is performed with the quantile–quantile plot. With the quantile–quantile plot, the dominance of acting algorithm over competing algorithms in terms of convergence rate is expressed very easily. Unlike the quality comparison where data sets are sorted before quantile plot and performed SLR analysis, here data sets are not sorted and SLR analysis is not performed. Main reason for the unsorted quantile–quantile plotting is to observe the dominance in convergence at every generation in the same order as they had appeared during the execution of the algorithm. Another key difference with the quality comparison is that the data sets considered in convergence comparison are not the best values obtained in various trials; instead, it contains values obtained at each generation for a single trial. To determine the dominance of an acting algorithm over competing algorithms, the quantile–quantile plot is observed with reference to the NL. The portions that are below the NL indicate actor-dominance, portions above the NL indicate competitor-dominance, and the portions on the NL indicate non-dominance. The level of dominance is determined based on how far the actor-dominating or competitor-dominating portions are from the NL.

5 Experimental evaluation and discussion

Empirical analysis is performed with 25 benchmark functions. Well-established algorithms are evaluated with the proposed methodology and verified with the existing studies. The details of empirical analysis such as benchmark

functions, algorithms and experimental setup are briefed as follows.

5.1 CEC-2005 benchmark functions

We have selected the set of 25 benchmark test problems that appeared in the CEC-2005 special session on real parameter optimization (Suganthan et al. 2005). All these optimization problems are minimization problems. The set contains following standard functions:

- 5 unimodal functions
 - f1: shifted sphere function.
 - f2: shifted Schwefel's problem 1.2.
 - f3: shifted rotated high-conditioned elliptic function.
 - f4: shifted Schwefel's problem 1.2 with noise in fitness.
 - f5: Schwefel's problem 2.6 with global optimum on bounds.
- 20 multimodal functions
 - 7 basic functions
 - f6: shifted Rosenbrock's function.
 - f7: shifted rotated Griewank function without bounds.
 - f8: shifted rotated Ackley's function with global optimum on bounds.
 - f9: shifted Rastrigin's function.
 - f10: shifted rotated Rastrigin's function.
 - f11: shifted rotated Weierstrass function.
 - f12: Schwefel's problem 2.13.
 - 2 expanded functions
 - f13: expanded extended Griewank's plus Rosenbrock's (Ef8f2)
 - f14: shifted rotated expanded Scaffers F6.
 - 11 hybrid functions (f15–f25). Each of these functions has been defined through compositions of 10 out of the 14 previous functions.

The optima of all the functions have been displaced from the origin or from the previous position to ensure that the optimal solutions can never be obtained at the center of the solution space. This displacement mechanism has made it difficult for the algorithms that have central tendency to converge toward the optimal solution. Hybridization has added more difficulty to the problem so that algorithm unable to follow certain patterns to reach the optimal solution.

5.2 Algorithms considered

We have considered three well-established EOAs for studying the proposed methodology for analyzing performance,

which includes two variants of PSO (Biswas et al. 2015) and one most efficient variant of DE (Veček et al. 2014). Numerous new EOAs are developed in recent years. However, the applicability of those algorithms has to be studied with variety of applications. The algorithms considered are very popular and already applied to wide range of applications of various domains. Besides this fact, another reason for considering these three algorithms is for a clear understanding of the methodology and verify with the existing studies. The algorithms are briefed as follows:

SADE: Self-Adaptive Differential Evolution (Qin and Suganthan 2005) is an extension to the differential evolution model. SADE can adapt parameters CR and F, suitable to corresponding situation.

PSO-TVIW: PSO-Time-Varying Inertia Weight (Shi and Eberhart 1999) is an instance of particle swarm optimization (PSO). In PSO-TVIW, linearly decrement of the weight parameter implies global search ability at the beginning and local search ability at the end of execution. The weight (ω) is decreased from $\omega_{\max} = 0.9$ to $\omega_{\min} = 0.4$.

PSO-TVAC: PSO-Time Varying Acceleration Coefficients (Ratnaweera et al. 2004) is an extension of PSO-TVIW, where linearly varying acceleration coefficients C_1 and C_2 are considered in addition to ω . This additional linearity to PSO-TVIW became more effective to global search during execution of the algorithm and to local search at the end. C_1 is varied from 2.5 to 0.5, C_2 from 0.5 to 2.5 and ω from 0.9 to 0.4.

5.3 Experimental setup

All experiments are carried out with population size 100. Dimensions 50 and 70 are considered for quality comparison and convergence comparison, respectively. For quality comparison, all algorithms are carried out 50 trials for observing best values. Each run stops when total functional evaluation exceeds 300,000. For convergence comparison, each algorithm is observed up to 1000 generations and noted the best value achieved at each generation. Functions f1–f14 are considered for quality comparison, which includes 5 unimodal functions, and rest of the functions are multimodal functions. Among these 14 functions f1–f4, f5–f12 and f6 & f8–f14 are considered for one-to-one, one-to-many and many-to-many quality comparison, respectively. Functions f15–f25 are considered for convergence comparison, all of them are hybrid multimodal functions. Among these 11 functions f15–f17, f18–f21 and f22–f25 are considered for the one-to-one, one-to-many and many-to-many convergence comparison, respectively.

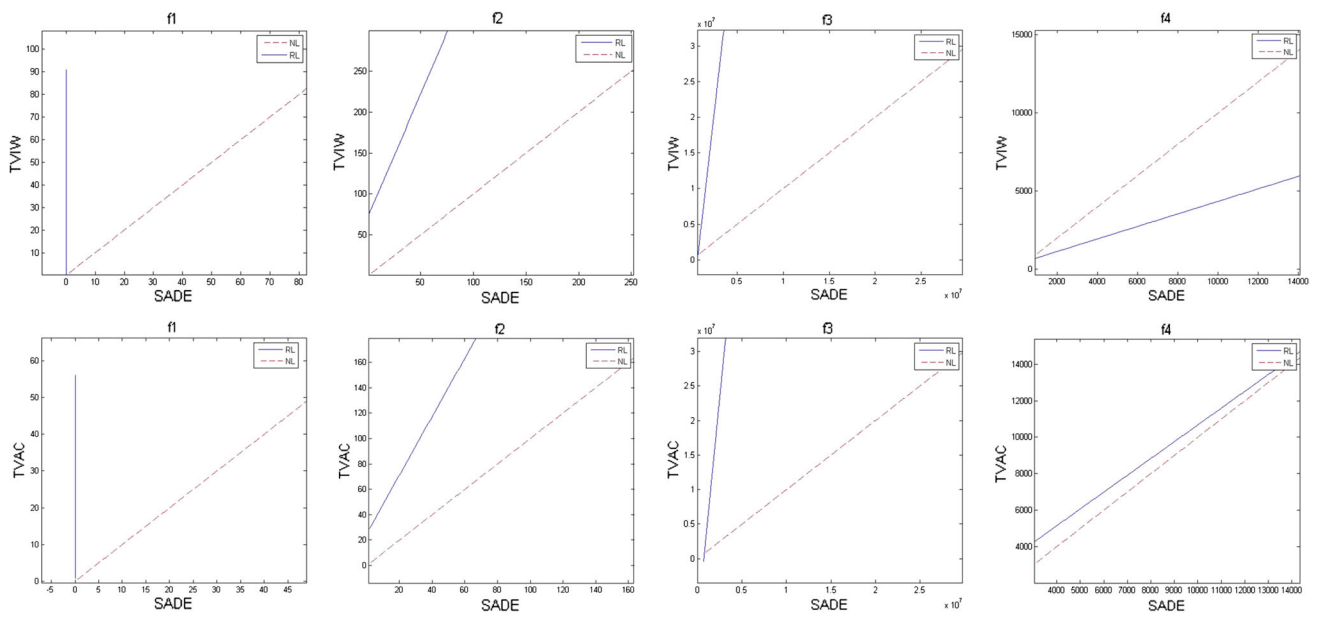


Fig. 11 One-to-one solution quality comparison in CEC2005 functions f1–f4

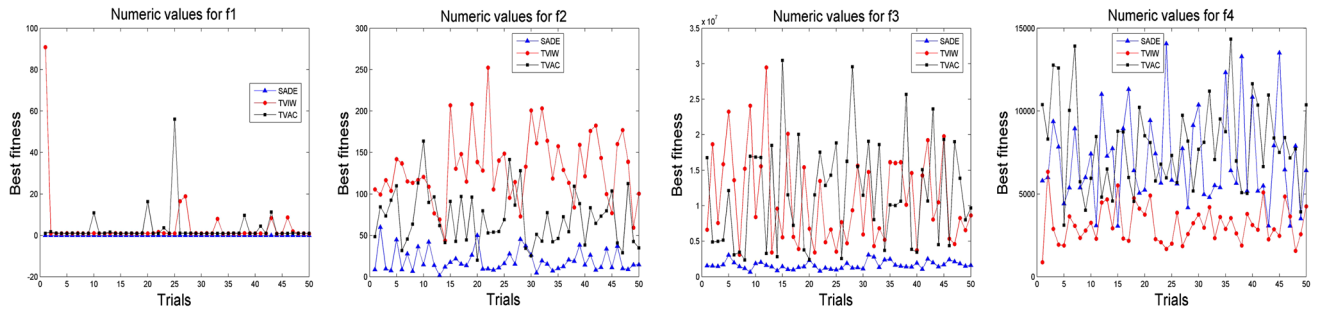


Fig. 12 Numeric values used for quality comparison in CEC2005 functions f1–f4

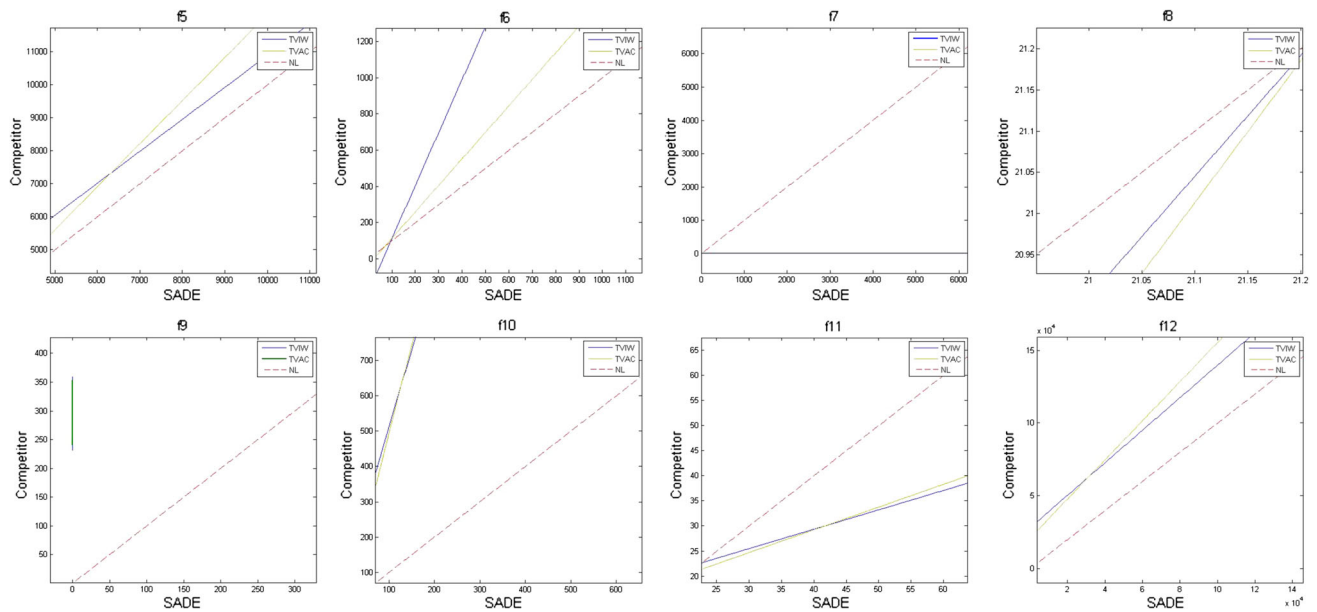


Fig. 13 One-to-many solution quality comparison in CEC2005 functions f5–f12

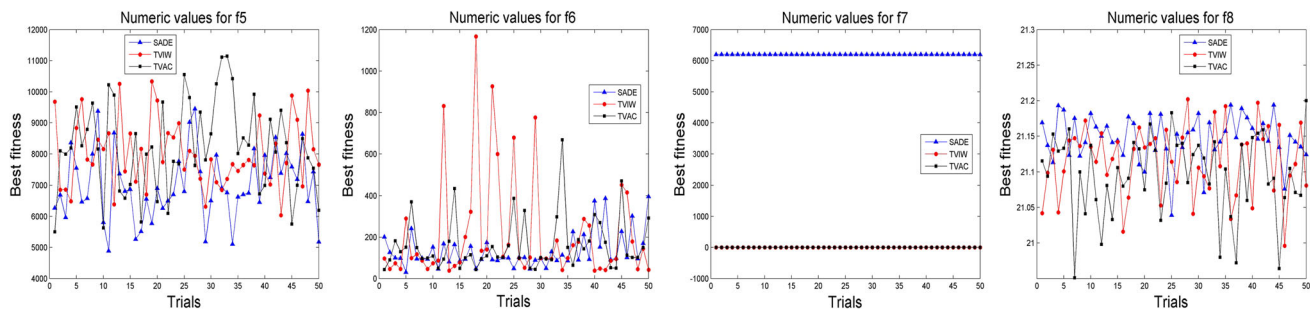


Fig. 14 Numeric values used for quality comparison in CEC2005 functions f5–f8

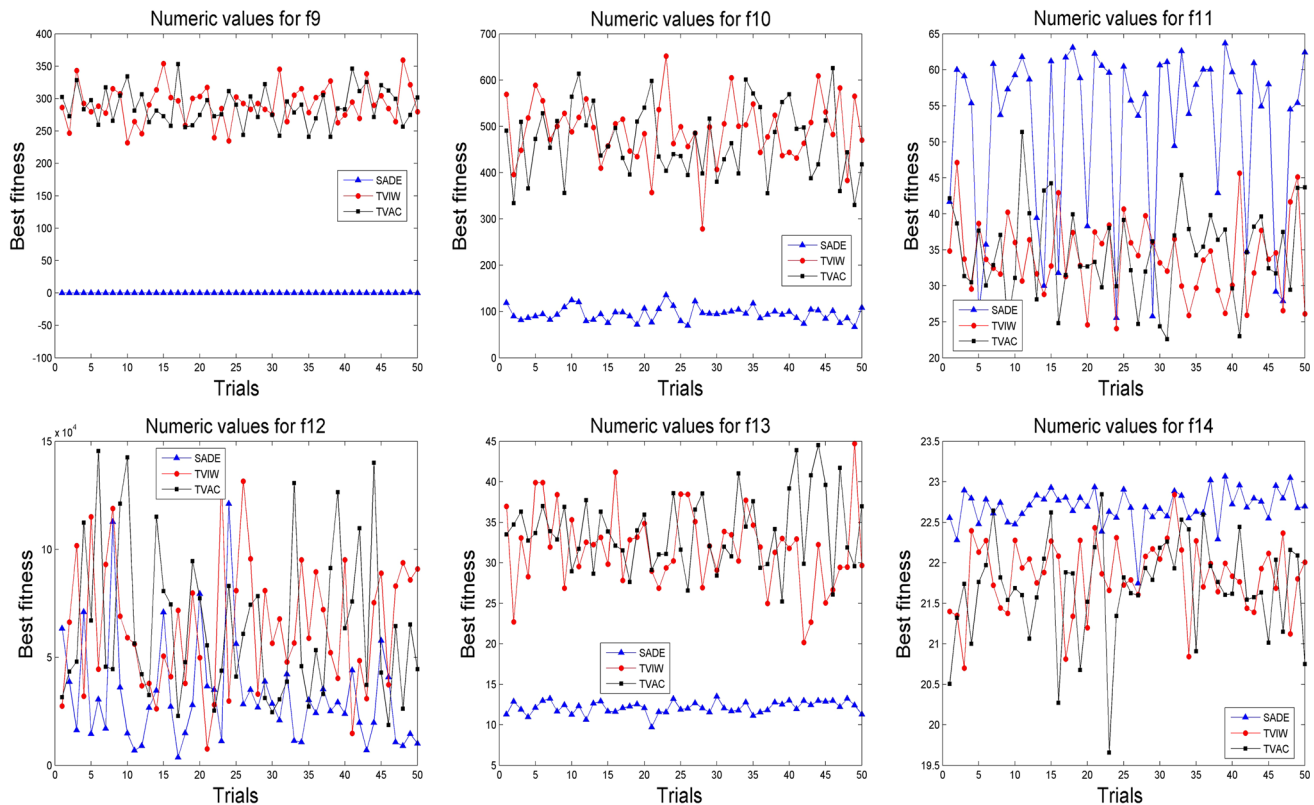


Fig. 15 Numeric values used for quality comparison in CEC2005 functions f9–f14

5.4 Experimental analysis

5.4.1 Quality analysis

For one-to-one and one-to-many comparison, SADE is considered as the acting algorithm, whose performance has to be evaluated with respect to other two competing algorithms. For many-to-many comparison, each of the three algorithms is considered as acting algorithm (comparing algorithm) during its turn and other two are considered as competing algorithms (counter competitor).

In Fig. 11, one-to-one comparison of SADE with TVIW (shorted PSO-TVIW) and TVAC (shorted PSO-TVAC) is presented. The interpretation of visual results is discussed

in Sect. 4, particularly the significance of the angle between NL and RL, and the position of RL with respect to NL is detailed. The visual results are analyzed based on the angle and position of RL with reference to NL as follows. For f1–f3, the RL is above and produce high angle (nearly the maximum angle 45°) with the NL. This implies SADE outperforms over both TVIW and TVAC. However, in the case of f4, solution quality of SADE is poor in comparison with TVIW, as RL lies below the NL. The numeric values (best fitness) that are used to generate the graphs in Fig. 11 are presented in Fig. 12. Clearly, most of the numeric values of SADE for f1–f3 are better than both TVIW and TVAC. The visual results also indicated the same. Similarly, in the case of f4, numeric values are clearly better than SADE. Visual results also indicated

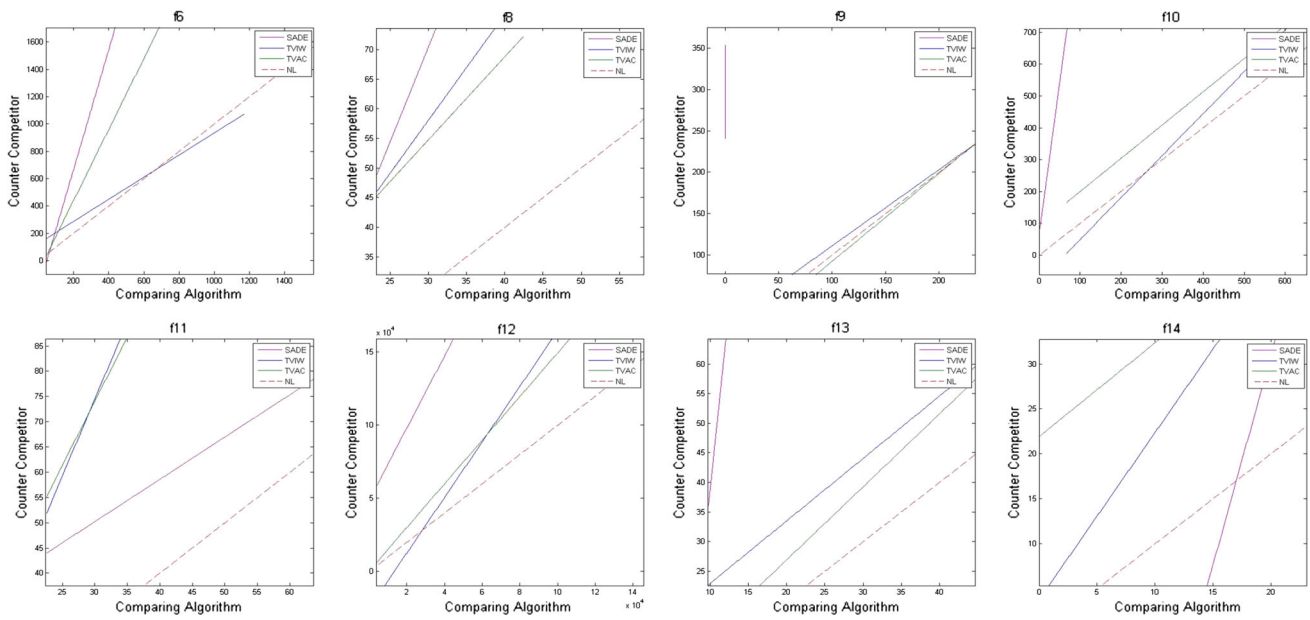


Fig. 16 Many-to-many solution quality comparison in CEC2005 functions f6 and f8–f14

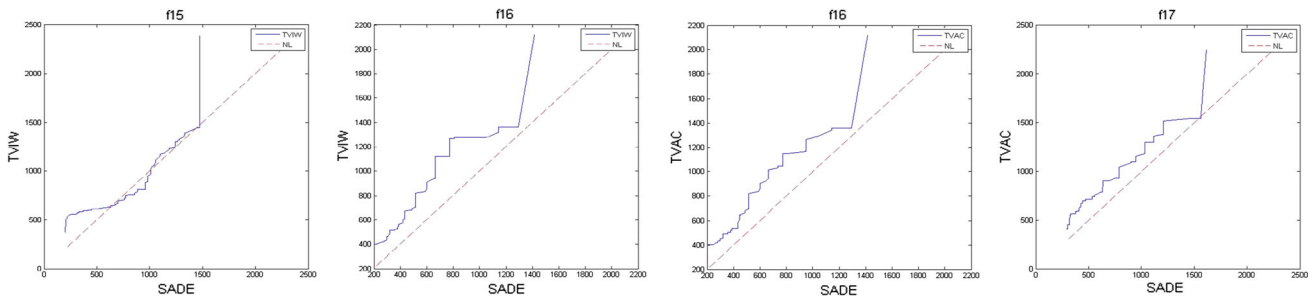


Fig. 17 One-to-one convergence comparison in CEC-2005 functions f15–f17

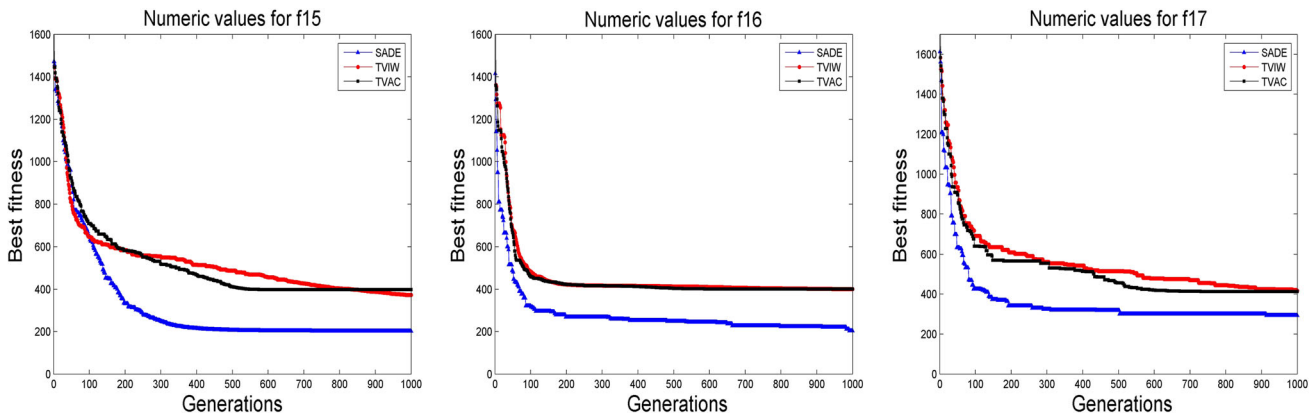


Fig. 18 Numeric values used for convergence comparison in CEC2005 functions f15–f17

the poor performance of SADE in comparison with TVIW. EOs always have the tendency to perform better in some functions, whereas overall they may perform worst (Wolpert and Macready 1997). Generally, TVAC performs better than TVIW (Biswas et al. 2015) and the SADE is best-performing

DE variant (Veček et al. 2014). Thus, better performance of SADE than TVAC justifies the poor performance than TVIW. Nevertheless, SADE performs better in other functions f1–f3.

In Fig. 13, visual results of the one-to-many comparison of SADE with respect to TVIW and TVAC are presented.

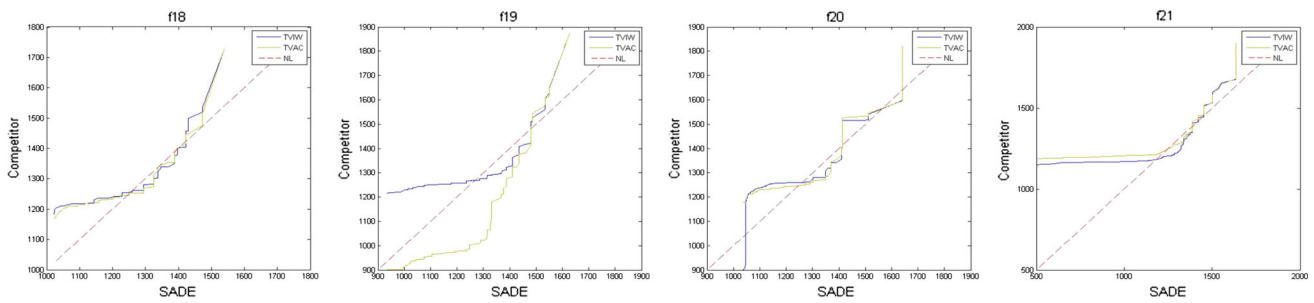


Fig. 19 One-to-many convergence comparison in CEC-2005 functions f18–f21

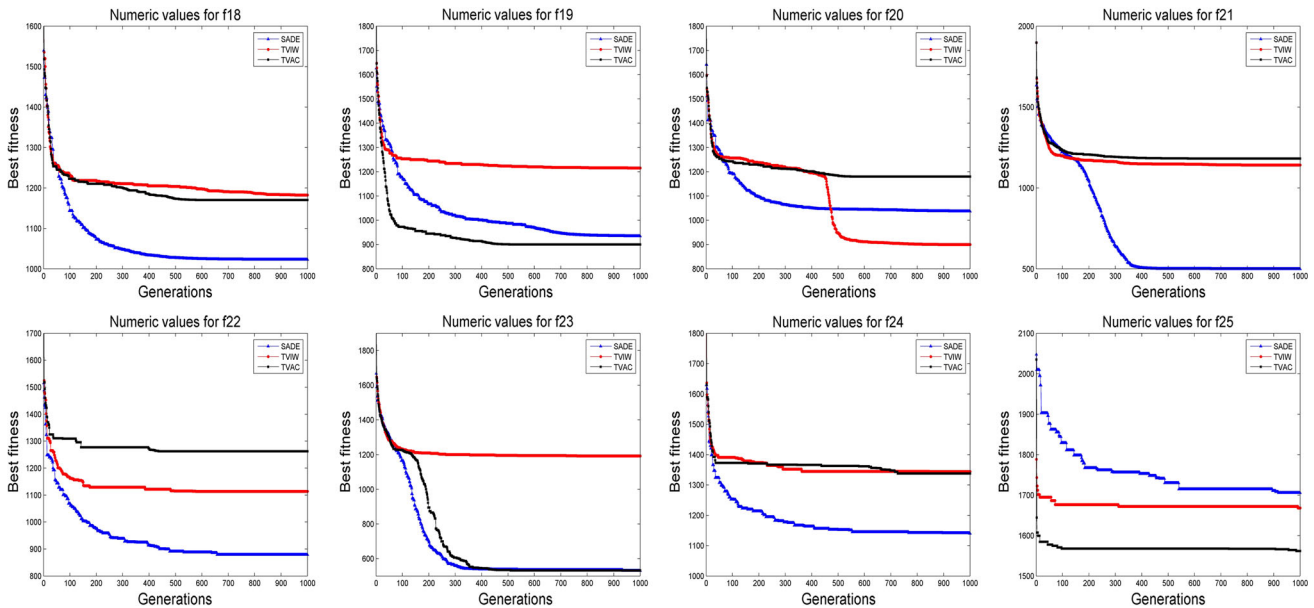


Fig. 20 Numeric values used for convergence comparison in CEC2005 functions f18–f25

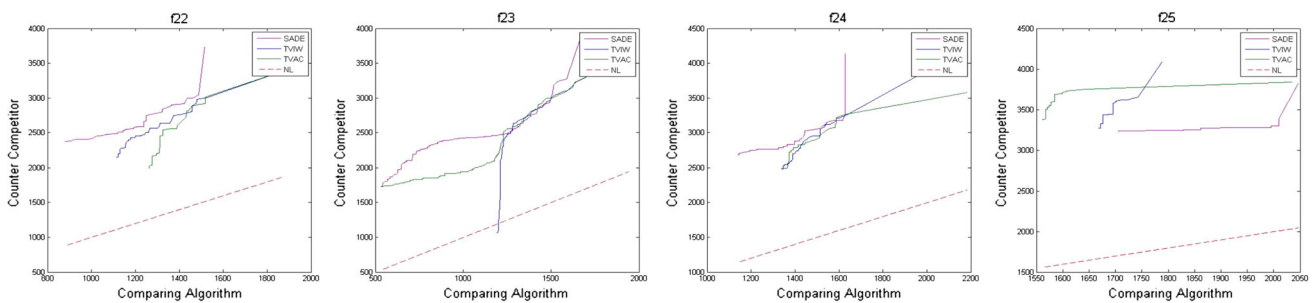


Fig. 21 Many-to-many convergence comparison in CEC-2005 functions f22–f25

Clearly, SADE’s performance is better in f5, f6, f9, f10 and f12 as RL corresponding to TVIW and TVAC lies above producing higher angle with NL. The numeric results for respective functions presented in Figs. 14 and 15 also indicated the same. In the visual results for f7, f8 and f11, RLs lie below the NL, which indicates poor solution quality of SADE in contrast to TVIW and TVAC. Clearly, one can notice the worst numeric values of SADE for f7, f8 and f11. However, overall SADE performs better than both TVIW and TVAC

since five out of eight functions SADE performs better than both the algorithms.

The visual results of many-to-many comparison, i.e., ranking of SADE, TVIW and TVAC are presented in Fig. 16. The RL representing SADE in f6, f8–f10, f12 and f13 lies above the NL, and it also lies above the RLs of both TVIW and TVAC. Therefore, SADE is ranked as best for these functions. Moreover, the angle between RL and NL for SADE is high in these functions, which indicates the SADE produces

best solutions than TVIW and TVAC. The numeric results for respective functions presented in Figs. 14 and 15 also indicated the same for these functions; particularly, one can notice the consistent best numeric values of SADE for the functions f9, f10 and f13. Now, the next best would be one of the two, i.e., either TVIW or TVAC for these functions. For the functions f6, f10 and f12, TVAC is the second best, while TVIW is the second best for rest of the functions. In case of f11, TVIW is ranked as best since angle between RL and NL is largest. In case of f14, although SADE has largest angle between RL and NL, SADE is not the best because its RL is not above the NL and intersect almost the middle of RL and NL. In this case, TVAC is best since its RL lies above all RLs and NL. One can notice in Fig. 15 that for both f11 and f14, SADE shows the worst numeric values in comparison with TVIW and TVAC.

5.4.2 Convergence analysis

Similar to the quality comparison, SADE is considered as an acting algorithm for one-to-one and one-to-many comparison to evaluate performance in terms of convergence rate. For many-to-many comparison, each of the three algorithms is considered as acting algorithm (comparing algorithm) during its turn and other two are considered as competing algorithms (counter competitor).

In Fig. 17, one-to-one convergence comparison is presented. In all f15–f17, SADE's convergence rate is higher than TVIW and TVAC except f15, since the entire portion of the plotting lies above the NL. In f15, initially, SADE's convergence is almost similar to TVIW, as the portion of plotting almost coincides with the NL. However, that middle portion lies below the NL indicates slight degradation in the convergence of SADE and at the end again shows improvement. The numeric values presented in Fig. 18 also show similar characteristics.

In Fig. 19, one-to-many convergence comparison is presented. In all f18–f21, convergence rate is almost similar for SADE with respect to both competitors, and in the middle of execution, convergence rate slightly degrades and again improves at the end. Numeric results presented in Fig. 20 also indicate same. In f19, although with respect to TVIW the rate improves at the end but with respect to TVAC degradation continues until the end. Most of the parts of the plotting of TVIW are above the NL, which indicates SADE's convergence rate is better in those portions. One can notice in Fig. 20 that the numeric values of SADE are also better in those portions, especially at the end of the plotting. This shows the plotting in visual results remains above NL if the convergence rate of SADE is better, which is valid for all the remaining functions as well.

In Fig. 21, visual results of many-to-many convergence comparison are presented. In f22–f24, the convergence rate

of SADE is higher than that of both counter competitors, i.e., TVIW and TVAC; the plotting also lies above both. Therefore, SADE is ranked as best in terms of convergence rate for the functions f22–f24. Accordingly, one can notice the best numeric values of SADE for f22–f24 in Fig. 20. In f25, TVAC's convergence rate is best with respect to both SADE and TVIW, and numeric values also indicate the same. It is clear that the second best convergence rate is TVIW for the functions f22, f24 and f25. In f23, TVAC is ranked as the second best algorithm in terms of convergence rate.

6 Conclusion

A visual analysis method is proposed to evaluate the performance of EOAs in terms of both solution quality and convergence rate. The methodology is designed based on the concepts of quantile–quantile plot and simple regression analysis. The quantile–quantile plot of data sets ensures involvement of each individual data in the evaluation process. In the case of quality comparison, the dominance of an algorithm is determined simply by observing the angle between RL and NL and position of the intersection of RL and NL. In the case of convergence comparison, the involvement of individual data is clearer since unsorted data are directly plotted in the quantile–quantile plot. The dominance of an algorithm in convergence comparison is simply determined by observing the distance of quantile–quantile plotting and NL. The proposed methodology is studied with three well-established algorithms: PSO-TVIW, PSO-TVAC and SADE. Data obtained on CEC-2005 benchmark functions are considered for the analysis. Clear shifting angles between RL and NL and intersecting points on NL are visualized through one-to-one, one-to-many and many-to-many in the case of quality comparisons. For convergence comparison, the dominance in the quantile–quantile plots is visualized with respect to NL.

The major advantage of the proposed method is that the underlying concepts are relatively simple and well known to most of the researchers in this domain. The methodology relies simply on linear regression and quantile–quantile plots to illustrate the dominance of one algorithm over another (or a set of algorithms). Another advantage is that the ranking of EOAs in terms of convergence rate is done very easily with proposed methodology, which cannot be done with the existing methods including widely used parametric approach and other approaches such as ELA or drift analysis. Moreover, the ranking of EOAs based on solution quality is also very simple, just needs to observe the position of the RLs of respective algorithms. However, for very large number of algorithms the visualization of RLs may become a little bit difficult because of overlapping of multiple lines. Other than this limitation, the proposed methodology has several

advantages, especially easy ranking of EOAs in terms of both solution quality and convergence rate.

Acknowledgements Part of the work has been appeared at ISCBI 2014 (Biswas and Biswas 2014). Proper citations have been included for the same in the above work for the purpose transparency.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Informed consent Also informed consent was obtained from all individual participants included in the study.

Human and animal rights This article does not contain any studies with human participants or animals performed by any of the authors.

References

- Biswas A, Biswas B (2014) Visual analysis of evolutionary optimization algorithms. In: 2014 2nd International symposium on computational and business intelligence (ISCBI), pp 81–84
- Biswas A, Gupta P, Modi M, Biswas B (2015) An empirical study of some particle swarm optimizer variants for community detection. In: El-Alfy E-SM, Thampi SM, Takagi H, Piramuthu S, Hanne T. (eds) *Advances in intelligent informatics*. Springer, Berlin, pp 511–520
- Carrano EG, Takahashi RH, Wanner EF (2008) An enhanced statistical approach for evolutionary algorithm comparison. In: Proceedings of the 10th annual conference on genetic and evolutionary computation (GECCO '08). ACM, New York, NY, USA, pp 897–904
- Czarn A, MacNish C, Vijayan K, Turlach B, Gupta R (2004) Statistical exploratory analysis of genetic algorithms. *IEEE Trans Evol Comput* 8(4):405–421
- Derrac J, García S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol Comput* 1(1):3–18
- Francois O, Lavergne C (2001) Design of evolutionary algorithms—a statistical perspective. *IEEE Trans Evol Comput* 5(2):129–148
- García S, Molina D, Lozano M, and Herrera F (2007) An experimental study on the use of non-parametric tests for analyzing the behaviour of evolutionary algorithms in optimization problems. In: Proceedings of the Spanish congress on metaheuristics, evolutionary and bioinspired algorithms (MAEB2007), pp 275–285
- García S, Molina D, Lozano M, Herrera F (2009) A study on the use of non-parametric tests for analyzing the evolutionary algorithms behaviour: a case study on the cec2005 special session on real parameter optimization. *J Heuristics* 15(6):617–644
- He J, Chen T (2013) Novel analysis of population scalability in evolutionary algorithms. CoRR abs/1108.4531. <http://arxiv.org/abs/1108.4531>
- He J, Yao X (2001) Drift analysis and average time complexity of evolutionary algorithms. *Artif Intell* 127(1):57–85
- Lockett A (2013) Measure-theoretic analysis of performance in evolutionary algorithms. In: 2013 IEEE congress on evolutionary computation (CEC), pp 2012–2019
- Lutton E, Fekete J-D (2011) Visual analytics and experimental analysis of evolutionary algorithms. Research Report RR-7605, INRIA. <http://hal.inria.fr/inria-00587170>
- Mersmann O, Preuss M, Trautmann H (2010) Benchmarking evolutionary algorithms: towards exploratory landscape analysis. Springer, Berlin
- Muhlenbein H, Mahnig T (2001) Mathematical analysis of evolutionary algorithms for optimization. In: Proceedings of the third international symposium on adaptive systems. La Havana, pp 166–185
- Moreno-Pérez J, Campos-Rodríguez C, Laguna M (2007) On the comparison of metaheuristics through non-parametric statistical techniques. In: Proceedings of the Spanish congress on metaheuristics, evolutionary and bioinspired algorithms (MAEB2007), pp 286–293
- Nijssen S, Back T (2003) An analysis of the behavior of simplified evolutionary algorithms on trap functions. *IEEE Trans Evol Comput* 7(1):11–22
- Qin A, Suganthan P (2005) Self-adaptive differential evolution algorithm for numerical optimization. In: The 2005 IEEE congress on evolutionary computation, vol 2, pp 1785–1791
- Ratnaweera A, Halgamuge S, Watson H (2004) Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. *IEEE Trans Evol Comput* 8(3):240–255
- Rojas I, Gonzalez J, Pomares H, Merelo J, Castillo P, Romero G (2002) Statistical analysis of the main parameters involved in the design of a genetic algorithm. *IEEE Trans Syst Man Cybern C Appl Rev* 32(1):31–37
- Shi Y, Eberhart R (1999) Empirical study of particle swarm optimization. In: Proceedings of the 1999 congress on evolutionary computation, 1999 (CEC 99), vol 3, p 1950
- Shilane D, Martikainen J, Dudoit S, Ovaska SJ (2008) A general framework for statistical performance comparison of evolutionary computation algorithms. *Inf Sci* 178(14):2870–2879
- Suganthan PN, Hansen N, Liang JJ, Deb K, Chen Y-P, Auger A, Tiwari S (2005) Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization. KanGAL report 2005005
- Veček N, Mernik M, Črepinšek M (2014) A chess rating system for evolutionary algorithms: a new method for the comparison and ranking of evolutionary algorithms. *Inf Sci* 277:656–679
- Wolpert D, Macready W (1997) No free lunch theorems for optimization. *IEEE Trans Evol Comput* 1(1):67–82
- Wu A, De Jong K, Burke D, Grefenstette J, Loggia Ramsey C (1999) Visual analysis of evolutionary algorithms. In: Proceedings of the 1999 congress on evolutionary computation, 1999 (CEC 99), vol 2, p 1425
- Yang X-S (2011) Metaheuristic optimization: algorithm analysis and open problems. In: Pardalos P, Rebennack S (eds) *Experimental algorithms*, vol 6630, lecture notes in computer science. Springer, Berlin, pp 21–32