JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 36, 1293-1314 (2020)
DOI: 10.6688/JISE.202011_36(6).0011

Broadening Selection Competitive Constraint Handling Algorithm for Faster Convergence

TAYYAB AHMED SHAIKH[1], SYED SAJJAD HUSSAIN[1], MUHAMMAD RIZWAN TANWEER[1] AND MANZOOR AHMED HASHMANI[2]
[1]*Faculty of Engineering Sciences and Technology*
Hamdard University
Karachi, 74600 Pakistan
E-mail: {tayyab.ahmed; dr.sajjad; rizwan.tanweer}@hamdard.edu.pk
[2]*Faculty of Science and Information Technology*
University Technology Petronas
32610 Seri Iskandar, Perak, Malaysia
E-mail: manzoor.hashmani@utp.edu.my

In this paper, a new algorithm incorporating broadening selection strategy in competitive constraint handling paradigm for finding the optimum solution in constrained problems has been proposed, referred as Broadening Selection Competitive Constraint Handling (BSCCH). Although, competitive constraint handling approaches have proved to be very efficient, but they lack faster convergence due to offspring generation from random individuals. By incorporating selection strategy such as broadening selection in the competitive approach, better results are obtained and convergence rate is improved significantly. Incorporating said strategy, the BSCCH algorithm has been proposed which is generic in nature and can be coupled with various evolutionary algorithms. In this study, the BSCCH algorithm has been coupled with Differential Evolution algorithm as a proof of concept because it is found to be an efficient algorithm in the literature for constrained optimization problems. The proposed algorithm has been evaluated using 24 benchmark functions. The mean closure performance of the BSCCH algorithm is compared against seven selected state-of-the-art algorithms, namely Differential Evolution with Adaptive Trial Vector Generation Strategy and Cluster-replacement-based Feasibility Rule (CACDE), Improved Teaching Learning Based Optimization (ITLBO), Modified Global Best Artificial Bee Colony (MGABC), Stochastic Ranking Differential Evolution (SRDE), Novel Differential Evolution (NDE), Partical Swarm Optimization for solving engineering problems – a new constraint handling mechanism (CVI-PSO) and Ensemble of Constraint Handling Techniques (ECHT). The median convergence traces have been compared with two different algorithms based on differential evolution, *i.e.* Ensemble of Constraint Handling Techniques (ECHT) and Stochastic Ranking Differential Evolution (SRDE). ECHT is considered to be a flagship ensemble technique till date for constrained optimization problems, whereas SRDE employs a parent selection mechanism for constrained optimization. The proposed algorithm is found to provide better solutions and achieve significantly faster convergence in most of the problems.

Keywords: constraint handling techniques, competitive approach, selection strategy, differential evolution, ranking methodology

Received December 4, 2019; revised March 12 & April 27, 2020; accepted May 27, 2020.
Communicated by Chia-Feng Juang.

1. INTRODUCTION

Constrained optimization is an essential requirement in various real-world problems such as energy consumption , antenna design [1] *etc*. Conventional techniques for solving optimization problems such as linear programming, quadratic programming, non-linear programming and dynamic programming find difficulty in solving complex problems involving non-linearities, discontinuous functions and discrete search spaces [2]. Therefore, researchers have devised evolutionary algorithms (EAs) which are usually derived from some natural phenomena [3] and are also called Nature-Inspired (NI) algorithms. There are various popular EAs presented in the literature like Differential Evolution (DE) [4], Artificial Colony Bee (ABC) [5], Particle Swarm Optimization (PSO) [6] *etc*. EAs are inherently developed for unconstrained optimization problems and Constraint Handling Techniques (CHTs) need to be separately incorporated into them.

Several CHTs have been given in the literature of which some of the efficient CHTs are Penalty Functions (PF)[7], Decoders[8], Feasibility Rules (FR)[9], Stochastic Ranking (SR)[10], ε-Constrained (EC) method [11] and various ensemble methods [13-15]. Besides, there are some of the newly developed techniques present in the literature which have not been thoroughly tested such as Multiple Constraint Ranking (MCR) [12] *etc*.

According to the No Free Lunch (NFL) theorem [13], no single EA and CHT is able to solve different types of problems efficiently. This is the reason for emergence of competitive techniques which combine advantages of multiple EAs and CHTs. In competitive paradigm, multiple CHTs are used for a constrained optimization problem. The best CHT of the moment is exploited during evolutionary process in order to get the most out of the situation. Depending on various parameters such as the fraction of feasible search space to whole search space, multi-modality, the type of EA and exploration/exploitation, different CHTs are effective during different phases of evolutionary process. Primarily, the selection criterion is a function of performance during different stages of the optimization process in a specific problem. In this connection, few researchers have reported competitive techniques which are also referred to as ensemble or integrated techniques. However, in the competitive algorithms present in the literature, offspring has been generated from random individuals resulting in the low convergence rate. In this work, broadening selection strategy has been applied in the competitive approach which gives better solution and improve the convergence rate significantly as evident from the results. The mean closure value and the median convergence trace of BSCCH algorithm are compared against selected state-of-the-art algorithms and BSCCH is found to be superior among them.

The rest of paper is organized as follows. In Section 2, Background of the constrained optimization problem has been described along with Selection and Competitive techniques. The proposed BSCCH algorithm details are given along with flow chart and pseudo code in Section 3. Experimental setup and results are presented as mean closure values and median convergence traces in Section 4 along with statistical validation. Section 5 highlights the conclusion and future work that can be done to improve this work.

2. BACKGROUND

A constrained optimization problem is formulated generally in the form of a non-linear programming problem as:

$$\begin{aligned} \text{Minimize} \quad &: \quad f(X), X = (x_1, x_1, ..., x_n) \text{ and } X \in S \\ \text{Subject to} \quad &: \quad g_i(X) \leq 0, i = 1, ..., p \\ & \quad\;\; h_j(X) = 0, j = p+1, ..., m \end{aligned} \tag{1}$$

Where f must not be a continuous function, but bounded. S is the entire search space. p is the number of inequality constraint and the number of equality constraints are $(m-p)$. At the global optimum solution, if inequality constraints satisfy the condition $g_i(X) = 0$, then these constraints are called active constraints. So, all the equality constraints are active constraints. The equality constraints are converted into inequality constraints and bundled as:

$$G_i(X) = \begin{cases} max(g_i(X), 0), i = 1, ..., p \\ max(|h_j(X)| - \delta, 0), j = p+1, ..., m \end{cases} \tag{2}$$

Here δ is subtracted as a tolerance value. An adaptive setting of δ proposed in [14] is also being used in this work. Since the objective is to find a feasible solution, however if a feasible solution is not found, then the solution with minimal overall constraint violation ν given by Eq. (3) is considered as an optimal solution.

$$\nu(X) = \frac{\sum_{i=1}^{m} w_i(G_i(X))}{\sum_{i=1}^{m} w_i} \tag{3}$$

Here $G_i(X)$ are the bundled inequality constraints and $w_i = 1/G_{max_i}$ is the weight parameter. G_{max_i} is the maximum violation of the constraint obtained so far.

To validate BSCCH algorithm, a set of 24 benchmark functions [15] have been used. A summary of constrained benchmark functions used in this work is given in Table 1. Details of benchmark functions specification are given in [15]. Furthermore, DE has been selected as the candidate EA for constrained optimization as it has been found to be a very popular nature-inspired meta-heuristic tool for constrained optimization problems. It is evident from literature that different variants of DE has been efficiently utilized numerous times by researchers for solving constrained optimization problems [16]. DE is a simple and efficient heuristic for global optimization. It is a stochastic search method. The idea behind DE algorithm is to introduce as much randomness as possible and then retain the best possible solution. In DE, there are basically three stages namely mutation, crossover and selection.

In the **mutation** stage, new vectors are generated through the addition of weighted difference between two population vectors with a third vector.

In the **crossover** stage, mutant vectors are mixed with parameters of target vector to yield trial vector. The crossover operation results in increasing diversity of parameter vectors.

In the **selection** stage, decision is taken that trial would be a member of the new generation or not. The trial vector is compared with target vector using greedy criterion. In this work, selection of parents has been done in the mutation stage whereas target vector decision is taken in the selection stage. The crossover is performed as usual.

Table 1. Summary of the 24 benchmark functions.

The columns show name of the function (Function No.), optimal values ($f(x*)$), number of variables (N), Function Type, LI is number of linear inequality constraints, NLI is number of nonlinear inequality constraints, LE is number of linear equality constraints and NLE is number of nonlinear equality constraints.

Function No.	Optimal Value($f(x*)$)	N	Function Type	LI	NLI	LE	NLE
F_1	-15.000	13	Quadratic	9	0	0	0
F_2	-0.80361910412559	20	Nonlinear	0	2	0	0
F_3	-1.00050010001000	10	Polynomial	0	0	0	1
F_4	-30,665.5386717834	5	Quadratic	0	6	0	0
F_5	5126.4967140071	4	Cubic	2	0	0	3
F_6	-6961.81387558015	2	Cubic	0	2	0	0
F_7	24.30620906818	10	Quadratic	3	5	0	0
F_8	-0.0958250414180359	2	Nonlinear	0	2	0	0
F_9	680.630057374402	7	Polynomial	0	4	0	0
F_{10}	7049.24802052867	8	Linear	3	3	0	0
F_{11}	0.7499	2	Quadratic	0	0	0	1
F_{12}	-1.000	3	Quadratic	0	1	0	0
F_{13}	0.053941514041898	5	Nonlinear	0	0	0	3
F_{14}	-47.7648884594915	10	Nonlinear	0	0	3	0
F_{15}	961.715022289961	3	Quadratic	0	0	1	1
F_{16}	-1.90515525853479	5	Nonlinear	4	34	0	0
F_{17}	8853.53967480648	6	Nonlinear	0	0	0	4
F_{18}	-0.866025403784439	9	Quadratic	0	12	0	0
F_{19}	32.6555929502463	15	Nonlinear	0	5	0	0
F_{20}	-	24	Linear	0	6	2	12
F_{21}	193.724510070035	7	Linear	0	1	0	5
F_{22}	236.430975504001	22	Linear	0	1	8	11
F_{23}	-400.055099999999584	9	Linear	0	2	3	1
F_{24}	-5.50801327159536	2	Linear	0	2	0	0

2.1 Selection Strategies

Individual selection is a very vital process in EAs as it directs evolutionary process efficiently towards goal. There are two stages in which selection of individuals is involved—Parent Selection and Survivor Selection. **Parent selection** is usually a random process which select individuals to produce offspring. **Survival selection** involves selection of population from current set of parent and offspring based on fitness and constraint violation values according to rules of CHT in constrained optimization algorithm. Survivor selection is an integral part of an EA. However, there are only few algorithms emphasizing selection of good parents for reproduction [17]. The importance of parent selection in EAs has recently been highlighted through design and analysis in [17] by exploring use of different parent selection mechanisms for evolutionary multi-objective optimization. Through selection of individuals which have greater chance to produce effective offspring generation, much accelerated performance towards optimal solution has been achieved for an optimization problem. In [18], a multi-objective evolutionary algorithm with parent selection using prospect indicator is used to effectively provide solutions in multi-objective optimization problem. The prospect indicator looks for the potential of an individual to generate offspring that dominates itself. Furthermore, a probabilistic parent selection mechanism in mutation stage of DE algorithm is presented in [19] for directing search effectively towards the goal.

2.2 Competitive Techniques

Competitive techniques have proved to be very efficient in solving constrained optimization problems [20]. Some of the ensemble approaches presented in literature can also be classified as competitive techniques as different methods compete with each other in these ensemble paradigms during the course of optimization process. In [20], a comprehensive survey of ensemble approaches for evolutionary algorithms is presented, in which different ensemble approaches present in literature have been classified based upon the technique applied to perform the ensemble. One of the flag-ship work for solving constrained problems regarding the ensemble paradigm has been presented in [21] named as Ensemble of Constraint Handling Techniques (ECHT). In ECHT, a competition of four CHTs had been performed during the course of EA. ECHT had been developed in such a way that each of the four CHT was used to evolve a specific sub-population closely communicating with each other. Based on experiments, it was concluded that ECHT performs better than each of the four CHTs used by ECHT algorithm. In [22], an ensemble using DE algorithm is presented with three CHTs *i.e.* feasibility rules, adaptive penalty functions and ε-constrained method used during different stages of the algorithm. In [23], DE based algorithm has been applied with four DE-mutations, two DE-recombination and two CHTs *i.e.* feasibility rules and ε-constrained method so that it generates sixteen variants. Each variant has been assigned to an individual in a single population algorithm.

Out of various competitive approaches present in literature [20], all of them have a lack of selection mechanism regarding the constrained optimization problem, instead they generate offspring from random individuals resulting in the low convergence rate.

3. THE PROPOSED BSCCH ALGORITHM

As concluded in the previous section, although, competitive approaches have proved to be very efficient in solving constrained optimization problem, there is no parent selection mechanism present in them. This work is about developing an algorithm which broadens the selection of individuals based on fitness and penalty values to generate offspring in the competitive paradigm, resulting in better solutions together with significantly improved convergence rate.

3.1 Broadening Selection

The broadening selection strategy means that selection is two-pronged *i.e.* in both parent and survivor selection stages. For parent selection, individuals are sorted based on fitness or penalty value of each individual. To broaden the selection, BSCCH algorithm employs probabilistic crossover (P_c) value for ranking based on fitness and penalty value. Sorting of individuals is performed based on these two factors and best performing individuals are selected to generate offspring. For survivor selection, the best penalty value is also selected and kept along with fitness through competition, according to rules of each CHT. The selected penalty value later is used in the next generation by sorting mechanism. Broadening Selection strategy has been summarized pictorially in Fig. 1 where as its details are given in Section 3.3.1.

The penalty value $p(X)$ given by Eq. (4) is adopted from [24]. $p(X)$ is computed such that all constraints are normalized so that each constraint has the same contribution.

$$p(X) = \sum_{i=1}^{n} \frac{C_i(X)}{C_{max_i}} \tag{4}$$

Here $C_i(X)$ is the i_{th} constraint violation, *i.e.* $C_i(X) = max(g_i(X), 0)$ for inequality constraints and $C_i(X) = max(|h_i(X)| - \delta, 0)$ for equality constraints. C_{max_i} is maximum violation of i_{th} constraint found yet during the evolutionary process. The tolerance value δ for equality constraints has been set to 10^{-4} as given in benchmark specifications [15].

3.2 Competitive Constraint Handling

In this work, four modern CHTs, namely Superiority of Feasibility Solutions (SF) [9], Self-Adaptive Penalty (SP) [25], Stochastic Ranking (SR) [10], ε-Constrained (EC)

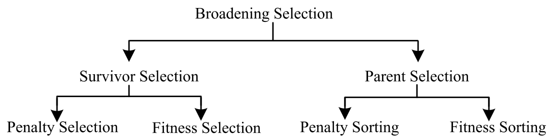


Fig. 1. Pictorial representation of broadening selection strategy.

method [11] have been used. In SF, simple rules are used to guide search towards the feasible region. This technique is very popular for constrained optimization due to its simplicity and flexibility, which makes it very suitable to be coupled with any sort of algorithm relatively easy without introducing new parameters, however, the main drawback is its affinity towards premature convergence. In SP, penalty value is tuned and added to infeasible individuals by using information from search process. SP is easy to implement and do not need to define parameters by the user. The SR technique is designed to deal with over and under penalization that occurs in penalty functions. SR has been applied to various EAs, but it is found to be deficient in generality, this is primarily because it requires tailoring according to the particular EA. EC technique transforms a constrained numerical optimization problem into an unconstrained numerical optimization problem. In EC, constraints are relaxed through the use of ε parameter. This technique is specifically effective for problems having equality constraints, but need careful tuning of ε parameter. These four selected CHTs have proved to be very efficient as evident from literature and their ensemble has marked breakthrough in constrained optimization [21]. As stated, each of the four CHTs has its own advantages and disadvantages, so they have been blended in the competitive paradigm in order to exploit the most out of the situation during the course of evolutionary process. The implementation and significance details of competitive constraint handling is given in Section 3.3.2.

3.3 Algorithm Details

Fig. 2 shows the proposed algorithm in the form of flowchart highlighting main contributions that have been performed in parent and survivor selection stages whereas Algorithm 1 list steps for BSCCH algorithm in the form of pseudo-code. The algorithm starts with random initialization of four populations corresponding to each CHT within the search space using bound constraints. Fitness and Penalty values of each population are then evaluated. In conventional DE algorithm, there is a random selection of individuals to generate offspring in the mutation stage. The broadened selection strategy employs the selection of individuals in both parent and survivor selection stages. In **parent selection** stage, selection of parents for offspring generation in DE algorithm is based on two factors, namely fitness and penalty. Individuals with the best fitness and penalty value are selected to generate offspring. To broaden parent selection, the algorithm employs probabilistic crossover (P_c) value for sorting individuals as given in Step 4 of Algorithm 1. Fitness and penalty value of each population along with its offspring are then evaluated. In **survivor selection** stage, competitive constraint handling has been performed in two phases as given in Step 6 of Algorithm 1. In Phase-I , offspring in one population is compared with the nearest neighbor in all other offspring populations to determine if it is better according to the rules of corresponding CHT. The nearest neighbor is determined by calculating euclidean distance between an individual in one offspring and all individuals of other offspring. The reason for choice of minimum distance is based on findings that it indicates both solutions are nearly similar. In other words, it actually does the operation of a crossover in DE, hence imparting more diversity [4]. In Phase-II, a population is compared with its offspring and updated accordingly. The updated penalty value is also selected and kept under competitive paradigm in survivor selection step, since it would be used in the next generation for broadening selection. The optimal value is selected from four populations based on minimum fitness if constraint violation is zero, otherwise minimal constraint violation is used to select optimal value.

Algorithm 1: Pseudo-code of BSCCH Algorithm

Input: Fitness($f(X)$), Constraints($g_i(X), h_i(X)$), Bound Constraints(X_{min}, X_{max})
Output: Optimal Solution $O(X)$

1: Randomly initialize each population X_i *i.e.* $X_{min} \leq X_i \leq X_{max}$, where $i = 1,2,3,4$
2: Evaluate $f(X), v(X), p(X)$ for each X_i using Eqs. (1), (3) and (4) respectively
3: Set *gen* to 1 and repeat Steps 4 to 8 until $gen < Max\_Fes$
4: Sort population and generate offspring
  if $p(X_i) == 0$ *OR* $rand[0,1] < P_c$ **then**
    sort X_i using $f(X_i)$
  else
    sort X_i using $p(X_i)$
  $r_i1 = sort(1), r_i2 = sort(2), r_i3 = sort(3)$ and $OS_i = r_i1 + F(r_i2 - r_i3)$
  Apply crossover for conventional DE and use bound constraints as in Step 1
5: Evaluate $f(X), v(X), p(X)$ for each X_i and OS_i as in Step 2
6: Competitive constraint handling is performed
  $d_{min} = min(d_{jk})$ where $d_{jk} = ||d_j - d_k||, j \neq k$ and $j,k = 1,2,3,4$
  if $OS_{d_{min}}$ is better than OS_i according to CHT$_i$ **then**
    $OS_i = OS_{d_{min}}$
  if OS_i is better than X_i according to CHT$_i$ **then**
    $X_i = OS_i$
7: Optimal value selection
  $f(X_{min}) = min(f(X_i)), v_{min}(X) = min(v_i)$
  if $v_{min}(X) == 0$ **then**
    $O(X) = f(X_{min})$
  else
    $O(X) = f(X_{v_{min}})$
8: $gen = gen + 1$

3.3.1 Implementation and significance of broadening selection

In Fig. 3, sorting mechanism adopted in broadening selection has been depicted. The mechanism has been illustrated by taking a low dimensional population of five individuals. Fig. 3 shows sorting of five individuals with respect to fitness, penalty and broadening selection. Individuals sorted according to fitness and penalty are as *1f,2f,3f,4f,5f* and *1p,2p,3p,4p,5p* respectively. However the broadening selection strategy sort the individuals according to the following criteria:

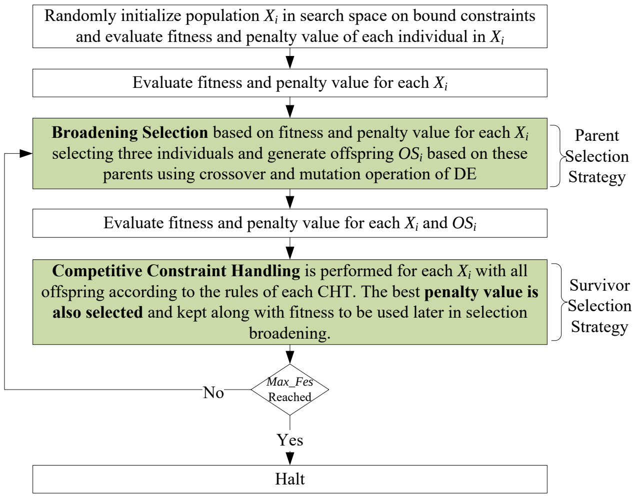


Fig. 2. Flowchart of the BSCCH algorithm.

if $(p(X_i) == 0$ or $rand[0,1] < P_c)$
  sort X_i using $f(X_i)$
else
  sort X_i using $p(X_i)$

Therefore, broadening selection sorting becomes *1f,1p,2f,3f,2p*. The broadening selection sorting in Fig. 3 clearly illustrates that there are more individuals (three yellow) from fitness ranking than individuals (two red) from penalty sorting. The reason for selection pressure being slighlty towards fitness sorting is the value of P_c, which has been explained in detail in Section 4.1.1.

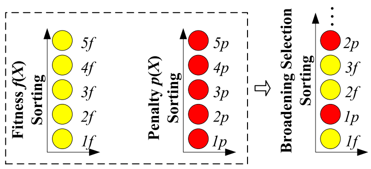


Fig. 3. Broadening selection sorting.

A pictorial representation of offspring generation through broadening selection is illustrated in Fig. 4. The mutation operation in DE algorithm is given by the following equation:

$$OS_i = r_i 1 + F(r_i 2 - r_i 3)$$

Here, $r_i 1, r_i 2, r_i 3$ are selected to be *1f,1p,2f* respectively as per broadening selection sorting criteria. The selection of $r_i 1, r_i 2, r_i 3$ is critical to exploit the trade-off between global exploration and local exploitation, since better selection leads faster towards the global optimum solution. In Fig. 4, it has been shown that the best individual *i.e.* $r_i 1$ which is closest to the global optimum has been selected to generate offspring OS_i.

Among the other two best particles, r_i2 has been selected from the infeasible region. The reason for selection of this infeasible individual is the broadening selection strategy which allows probabilistic best particle selection as illustrated in Fig. 3. The selection of r_i2 also ensures global exploration *i.e.* infeasible particles close to the optimum remain in the population. Thus, population diversity is maintained which may help in achieving the solution faster while avoiding premature convergence as shown in Fig. 4. The third particle r_i3 has also been selected from the feasible region. Thus r_i1, r_i3 helps in local exploitation resulting in faster convergence *i.e.* the generated offspring move very fast towards the global optimum. Therefore, BSCCH algorithm contain a balance between diversity and convergence when selecting individuals to generate offspring avoiding premature convergence as evident from the results in Section 4.

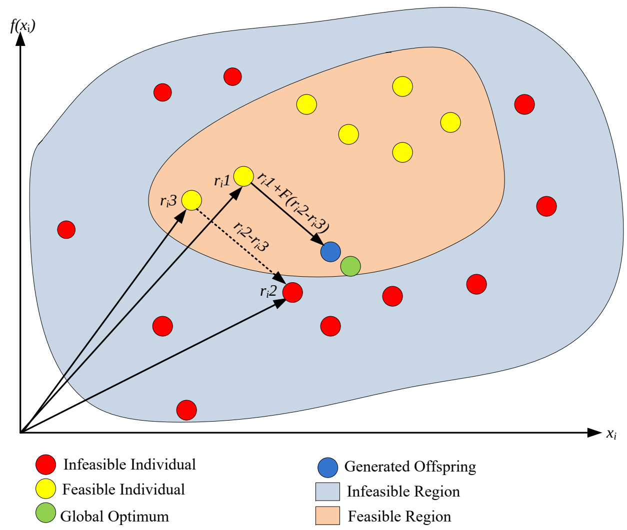


Fig. 4. Importance of broadening selection.

3.3.2 Implementation and significance of competitive constraint handling

The implementation and significance of competitive constraint handling in BSCCH algorithm is illustrated in Fig. 5. The importance of survivor selection through competitive constraint handling has been depicted in Fig. 5 by taking example for population X_1 corresponding to Constraint Handling Technique one (CHT_1). The same thread is running independently for other three populations *i.e.* X_2, X_3 and X_4. The survivor selection is performed in two phases, In Phase-I, an offspring is compared with other offsprings, where as in Phase-II, the corresponding offspring is compared with its parent. In Fig. 5, horizontal axis represent phases along which the algorithm proceeds whereas vertical axis represents selection criteria of an individual according to the rules of corresponding Constraint Handling Technique (CHT_1 Criteria).

In Phase-I, the euclidean distance between offspring OS_1 and other three populations *i.e.* OS_2, OS_3 and OS_4 is calculated by using the following equation.

$$d_{jk} = ||d_j - d_k||, j \neq k \text{ and } j,k = 1,2,3,4$$

Here, $j = 1$ for offspring OS_1. Therefore, three distances calculated by the preceding equation from offspring OS_1 to OS_2, OS_3, OS_4 are d_{12}, d_{13}, d_{14} respectively as illustrated in Phase-I of Fig. 5. Among these distances the minimum distance d_{min} as evident from the Fig. 5 is d_{13}, so offspring OS_3 is selected for comparison with OS_1 according to CHT_1 Criteria to see if is better. As evident from Phase-I of Fig. 5, OS_3 is better than OS_1 according to CHT_1 Criteria, hence offspring OS_1 is replaced by offspring OS_3 *i.e.* OS_3 becomes OS_1, as shown in the Phase-II.

In Phase-II, the replaced offspring *i.e.* OS_1 is compared with parent population *i.e.* X_1 to see if it is better according to CHT_1 Criteria. Since it has been shown in Fig. 5 that OS_3 is better than X_1 , therefore, X_1 is replaced by OS_3 *i.e.* OS_3 becomes X_1, as depicted in Phase-II result.

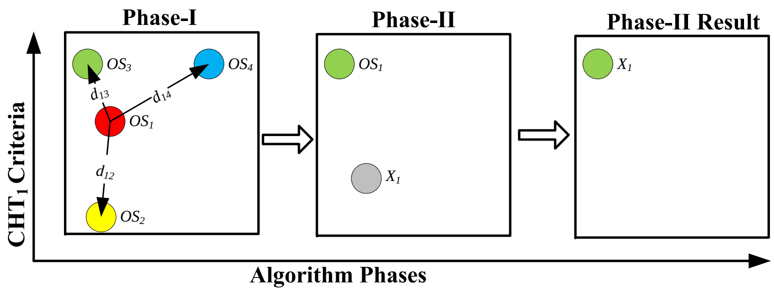


Fig. 5. Illustration of competitive constraint handling phases.

It is worth mentioning here that in BSCCH algorithm, apart from keeping the selected individuals and their fitness, penalty values are also selected and kept in survivor selection stage through competitive constraint handling. Other competitive techniques in the literature such as ECHT [21] do not keep the selected penalty values as opposed to BSCCH algorithm. The reason for keeping the selected penalty values is that, they are being used in next iteration for sorting individuals by broadening selection strategy.

Furthermore, as BSCCH algorithm involves sorting and competitive constraint handling stages. The complexity of BSCCH algorithm is driven by the competitive technique ($O(N^3)$) because it has higher complexity as compared to sorting ($O(N^2)$). Competitive approach involves comparison of each corresponding CHT population with its nearest neighbour, therefore this result in the computational complexity of $O(N^3)$.

3.3.3 Demonstration of BSCCH algorithm convergence compared with other strategies

A demonstration of selection strategies in BSCCH algorithm has been presented in Fig. 6 with the help of two problems from benchmark functions [15] *i.e.* Function#2 and Function#23 having high dimensionality *i.e.* 20 and 22 respectively. Function#2 is a non-linear function having non-linear inequality constraints whereas Function#23 is a linear function having non-linear inequality, linear equality and non-linear equality constraints. These two functions show remarkable improvement in the convergence towards solution when solved by BSCCH algorithm. In Fig. 6, five convergence traces have been plotted against no. of generations namely ECHT, SRDE, BSCCH, Competitive Constraint Handling with Parent Selection only (CCH-PS) and Competitive Constraint Handling with Parent Selection incorporating Fitness based ranking only (CCH-PS-FIT). These experiments have been setup based on parameters described in Section 4.3. A brief description of the four methods is given below:

1. In ECHT [21], a combination of the four CHTs has been done with no parent selection mechanism. ECHT has better convergence rate as compared with SRDE [19] but results in slow convergence rate when compared with BSCCH due to absence of selection strategy.
2. In SRDE [19], a single CHT named SF is used along with parent selection strategy to demonstrate the effectiveness of selection strategy but it lacks both closure and convergence.
3. In CCH-PS, competitive constraint handling incorporating parent selection mechanism has been performed. There is no penalty selection in survivor selection stage. Therefore, the convergence gets badly effected and it falls even below ECHT because best penalty individuals are ignored. This highlights the significance of employing Broadening Selection Strategy which incorporate selection into both parent and survivor stages.
4. In CCH-PS-FIT, only fitness is used for ranking in parent selection. Therefore, this technique show oscillatory behavior due to juggling of the population between feasible and infeasible individuals.
5. In BSCCH algorithm, both parent and survivor selection is employed in CCH resulting in better closure value together with significantly improved convergence rate.

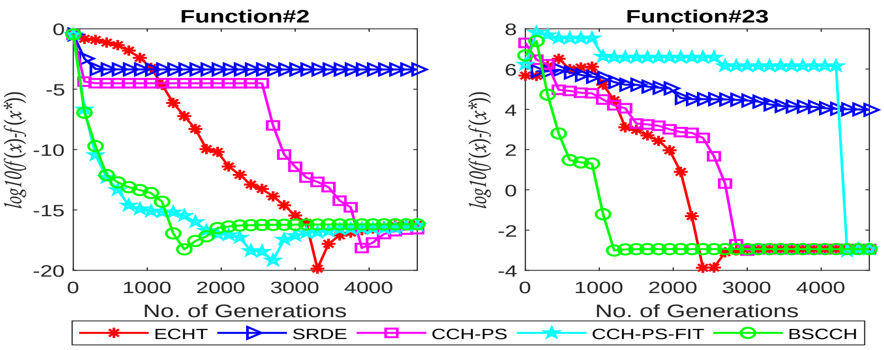


Fig. 6. Demonstration of BSCCH algorithm convergence.

4. EXPERIMENTAL SETUP AND RESULTS

In order to evaluate the effectiveness and performance of the proposed algorithm, two separate experiments have been performed, namely mean closure value comparison and median convergence trace comparison. The experiments have been setup based on benchmark specifications [15]. The maximum number of function evaluations (*Max_Fes*) have been set to 240000, since this value is being used by most of the researchers with which comparison has been performed.

4.1 Parameter Settings

4.1.1 Probabilistic crossover value (P_c)

The selection pressure can be adjusted by changing the value of P_c. If $P_c = 0$, the ranking is completely based on penalty value and results in over-penalization because infeasible individuals are given preference over feasible individuals. If $P_c = 1$, the ranking is under-penalized such that infeasible individuals get higher rank than feasible individuals. Since the objective is to found a feasible solution, a value of less than 0.5 should be selected to increase the selection pressure against infeasible individuals. The crossover value $P_c = 0.45$ (or probability crossover value) for BSCCH has been selected based on the results in [10]. This value is optimal to strike the balance between feasible and infeasible individuals.

4.1.2 Population size (N)

In case of population size for the base DE algorithm, most of the researchers have used a population size of 50 [21, 26]. Increasing the population size significantly increases the computational complexity of the algorithm. Therefore, we have also chosen 50 as the population size for BSCCH algorithm.

4.2 Mean Closure Value Comparison

Here, BSCCH algorithm has been run 25 times on selected benchmark functions. The algorithm has successfully found a feasible solution for all the problems except functions F_{20} and F_{22}. Other algorithms included in comparison have also been unable to find a feasible solution in functions F_{20} and F_{22}, therefore these two functions are not included in comparison.The mean closure value of BSCCH is compared against seven selected state-of-the-art algorithms *i.e.* CACDE [27], ITLBO [28], MGABC [29], SRDE [19], NDE [26], CVI-PSO [30] and ECHT [21]. The parameters *i.e.* population size (N), number of independent runs (*runs*) and *Max_Fes* for all algorithms have been given in Table 2 (n is dimension of the problem).

Table 2. Parameters setting of the competing algorithms.

Parameter	Algorithm							
	BSCCH	CACDE	ITLBO	MGABC	SRDE	NDE	CVI-PSO	ECHT
N	50	$N = \begin{cases} 50, 0 < n \leq 5 \\ 80, 5 < n \leq 10 \\ 100, 10 < n \leq 30 \end{cases}$	50	100	100	200	50	50
runs	25	25	25	30	31	30	20	25
Max_Fes	240,000	240,000	240,000	240,000	250,000	240,000	25,000	240,000

Table 3. Best, mean and standard deviation performances for F_1-F_{12} benchmark functions. Entries marked as * are not available in the literature.

Algorithm	**F_1**			**F_2**		
	Best	Mean	SD.	Best	Mean	SD.
BSCCH	-15.0000	**-15.0000**	0.00E+00	-0.80362	-0.80238	3.44E-02
CACDE	-15.0000	**-15.0000**	0.00E+00	-0.8036	**-0.8036**	7.99E-08
ITLBO	*	**-15.0000**	0.00E+00	*	-0.80226	3.26E-03
MGABC	-15.0000	-13.55354	16.36E-01	-0.8036108	-0.7890629	1.19E-02
SRDE	-15.0000	**-15.0000**	0.00E+00	-0.80362	-0.796772	9.50E-03
NDE	-15.0000	**-15.0000**	0.00E+00	-0.80348	-0.801809	5.10E-04
CVI-PSO	-15.0000	**-15.0000**	0.00E+00	-0.800097	-0.790875	1.09E-02
ECHT	-15.0000	**-15.0000**	4.00E-14	-0.80362	-0.80114	4.52E-03
	F_3			**F_4**		
	Best	Mean	SD.	Best	Mean	SD.
BSCCH	-1.0005	**-1.0005**	9.39E-16	-30665.5387	-30665.5387	7.43E-12
CACDE	-1.0005	**-1.0005**	2.32E-09	-30665.5387	-30665.5387	3.71E-12
ITLBO	*	**-1.0005**	2.58E-09	*	-30665.539	3.71E-12
MGABC	-1.0004	-1.0003	4.10E-05	-30665.54	-30665.54	1.04E-11
SRDE	-1.0005	**-1.0005**	1.70E-07	-30665.5387	-30665.5387	0.00E+00
NDE	-1.0005	**-1.0005**	2.32E-09	-30665.539	-30665.539	0.00E+00
CVI-PSO	-1.0000	-1.0000	3.70E-16	-30665.8217	**-30665.8210**	3.39E-03
ECHT	-1.0005	**-1.0005**	2.48E-16	-30665.5387	-30665.5387	5.04E-12
	F_5			**F_6**		
	Best	Mean	SD.	Best	Mean	SD.
BSCCH	5126.4967	**5126.4967**	2.78E-12	-6961.8139	**-6961.8139**	3.71E-12
CACDE	5126.4967	**5126.4967**	2.66E-12	-6961.8139	**-6961.8139**	0.00E+00
ITLBO	*	**5126.4967**	2.78E-12	*	**-6961.8139**	0.00E+00
MGABC	5126.497	5467.7560	3.30E+02	-6961.803	-6959.4890	1.17E+00
SRDE	5126.4967	5143.9610	4.13E+01	-6961.8139	**-6961.8139**	0.00E+00
NDE	5126.4967	**5126.4967**	0.00E+00	-6961.8139	**-6961.8139**	0.00E+00
CVI-PSO	5127.2776	5127.2776	0.00E+00	-6961.8139	**-6961.8139**	0.00E+00
ECHT	5126.4967	**5126.4967**	2.78E-12	-6961.8139	**-6961.8139**	3.71E-12
	F_7			**F_8**		
	Best	Mean	SD.	Best	Mean	SD.
BSCCH	24.3062	**24.3062**	3.58E-05	-0.09582504	-0.09582504	8.95E-18
CACDE	24.3062	**24.3062**	7.64E-15	-0.09582504	-0.09582483	3.59E-07
ITLBO	*	**24.3062**	1.51E-05	*	-0.095825	1.42E-17
MGABC	24.4056	24.6639	1.25E-01	-0.095825	-0.095825	0.00E+00
SRDE	24.3062	**24.3062**	3.10E-06	-0.095825	-0.095825	0.00E+00
NDE	24.3062	**24.3062**	1.35E-14	-0.095825	-0.095825	0.00E+00
CVI-PSO	24.4738	26.5612	1.64E+00	-0.10545951	**-0.105459505**	0.00E+00
ECHT	24.3062	**24.3062**	7.06E-15	-0.09582504	-0.09582504	1.32E-17
	F_9			**F_{10}**		
	Best	Mean	SD.	Best	Mean	SD.
BSCCH	680.630057	680.630057	2.30E-13	7049.2480	**7049.2480**	4.60E-04
CACDE	680.630057	680.630057	3.60E-13	7049.2480	**7049.2480**	2.72E-12
ITLBO	*	**680.63**	3.36E-13	*	7049.249	4.29E-05
MGABC	680.6302	680.6309	5.12E-04	7104.006	7357.461	1.21E+02
SRDE	680.630057	680.630057	0.00E+00	7049.2480	**7049.2480**	5.60E-04
NDE	680.630057	680.630057	0.00E+00	7049.2480	**7049.2480**	3.41E-09
CVI-PSO	680.635400	680.7557052	7.92E-02	7049.2765	7053.2143	1.06E+01
ECHT	680.630057	680.630057	2.53E-13	7049.2480	**7049.2480**	2.91E-12
	F_{11}			**F_{12}**		
	Best	Mean	SD.	Best	Mean	SD.
BSCCH	0.7499	**0.7499**	1.13E-16	-1.000	**-1.000**	0.00E+00
CACDE	0.7499	0.749914912	7.46E-05	-1.000	**-1.000**	0.00E+00
ITLBO	*	**0.7499**	1.13E-16	*	**-1.000**	0.00E+00
MGABC	0.749995	0.750025	3.40E-04	-1.000	**-1.000**	0.00E+00
SRDE	0.7499	**0.7499**	0.00E+00	-1.000	**-1.000**	0.00E+00
NDE	0.749999	0.749999	0.00E+00	-1.000	**-1.000**	0.00E+00
CVI-PSO	0.750000	0.7500000	0.00E+00	-1.000	**-1.000**	0.00E+00
ECHT	0.7499	**0.7499**	1.13E-16	-1.000	**-1.000**	0.00E+00

Table 4. Best, mean and standard deviation performances for F_{13}-F_{24} benchmark functions. Entries marked as * are not available in the literature.

Algorithm	F_{13}			F_{14}		
	Best	Mean	SD.	Best	Mean	SD.
BSCCH	0.053941514	**0.053941514**	2.61E-17	-47.7649	**-47.7649**	2.01E-04
CACDE	0.053941514	**0.053941514**	3.98E-12	-47.7649	**-47.7649**	2.24E-14
ITLBO	*	0.054008	0.00E+00	*	**-47.7649**	3.80E-05
MGABC	0.05394861	0.171074	1.74E-01	-47.675860	-47.246220	2.85E-01
SRDE	0.053942	0.260159	2.10E-01	-47.764888	-47.764671	5.90E-04
NDE	0.05394151514	**0.053941514**	0.00E+00	-47.7649	**-47.7649**	0.00E+00
CVI-PSO	0.0555558210	0.065590744	1.02E-02	-47.4530	-44.4246	1.41E+00
ECHT	0.053941514	0.146308176	1.67E-01	-47.7649	**-47.7649**	2.21E-14
	F_{15}			F_{16}		
	Best	Mean	SD.	Best	Mean	SD.
BSCCH	961.7150222	961.7150222	5.80E-13	-1.905155	**-1.905155**	8.88E-16
CACDE	961.7150223	961.7150223	3.42E-11	-1.905155	**-1.905155**	4.53E-16
ITLBO	*	961.72	5.80E-13	*	-1.9052	4.53E-16
MGABC	961.715100	962.173700	7.77E-01	-1.905155	**-1.905155**	0.00E+00
SRDE	961.715022	**961.715022**	0.00E+00	-1.905155	**-1.905155**	0.00E+00
NDE	961.7150223	961.7150223	0.00E+00	-1.905155	**-1.905155**	0.00E+00
CVI-PSO	961.71570715	961.7185955	6.87E-04	-1.905155	**-1.905155**	8.52E-15
ECHT	961.7150222	961.7150222	5.80E-13	-1.905155	**-1.905155**	4.53E-16
	F_{17}			F_{18}		
	Best	Mean	SD.	Best	Mean	SD.
BSCCH	8853.53967	8853.53967	5.56E-12	-0.866025404	**-0.866025404**	2.26E-17
CACDE	8853.533875	8853.533965	3.14E-04	-0.866025404	**-0.866025404**	4.53E-17
ITLBO	*	8959.8	3.77E+01	*	-0.866025	1.68E-05
MGABC	8853.53	8915.998	7.08E+01	-0.8660253	-0.8657735	2.99E-04
SRDE	8853.820571	8924.71736	2.88E+01	-0.866025	-0.85369	4.70E-02
NDE	8853.533874	**8853.533874**	0.00E+00	-0.8660254	-0.8660254	0.00E+00
CVI-PSO	8853.539891	8853.539891	3.70E-12	-0.8646313	-0.809109259	6.27E-02
ECHT	8853.53967	8853.53967	5.56E-12	-0.866025404	**-0.866025404**	5.49E-16
	F_{19}			F_{21}		
	Best	Mean	SD.	Best	Mean	SD.
BSCCH	32.655703	32.73189	1.08E-01	193.7245100	**193.7245100**	3.40E-11
CACDE	32.65559	**32.65559**	5.79E-10	193.7245101	193.7245101	3.31E-11
ITLBO	*	32.662	1.06E-02	*	222.22	4.84E+01
MGABC	*	*	*	*	*	*
SRDE	32.655594	32.728826	1.70E-01	193.72451	210.62308	2.75E+01
NDE	632.65559	32.65562	3.73E-05	193.7245101	193.7245101	6.26E-11
CVI-PSO	32.82702	35.06733	2.28E+00	193.7869252	193.7869352	3.38E-05
ECHT	32.65559	**32.65559**	2.08E-07	193.7245100	**193.7245100**	1.95E-11
	F_{23}			F_{24}		
	Best	Mean	SD.	Best	Mean	SD.
BSCCH	-400.0551	**-400.0551**	1.85E-07	-5.508013272	**-5.508013272**	2.71E-15
CACDE	-400.0551	-399.992164	3.15E-01	-5.508013272	**-5.508013272**	9.06E-16
ITLBO	*	-256.4	1.42E+02	*	**-5.508013**	9.06E-16
MGABC	*	*	*	-5.508013	-5.508013	1.77E-15
SRDE	-400.0551	-383.525923	2.26E+01	-5.508013	-5.508013	0.00E+00
NDE	-400.0551	**-400.0551**	0.00E+00	-5.50801327	-5.50801327	0.00E+00
CVI-PSO	-400.0000	-400.000000	0.00E+00	-5.508013272	**-5.508013272**	9.46E-15
ECHT	-400.0551	**-400.0551**	3.26E-06	-5.508013272	**-5.508013272**	2.71E-15

Table 5. Statistical validation test for mean closure ranks.

$F_{stat}(9.39) > F_{crit}(2.07)$ with $CD = 1.06$	**Algorithm**						
	CACDE	ITLBO	MGABC	SRDE	NDE	CVI-PSO	ECHT
Mean Rank Diff. w.r.t. BSCCH	0.16	1.68	3.02	1.87	0.82	2.37	0.09

Table 6. Statistical validation test for median convergence trace ranks.

$F_{stat}(10.21) > F_{crit}(3.21)$ with $CD = 0.45$	**Algorithm**	
	SRDE	ECHT
Mean Rank Diff. w.r.t. BSCCH	1.00	0.70

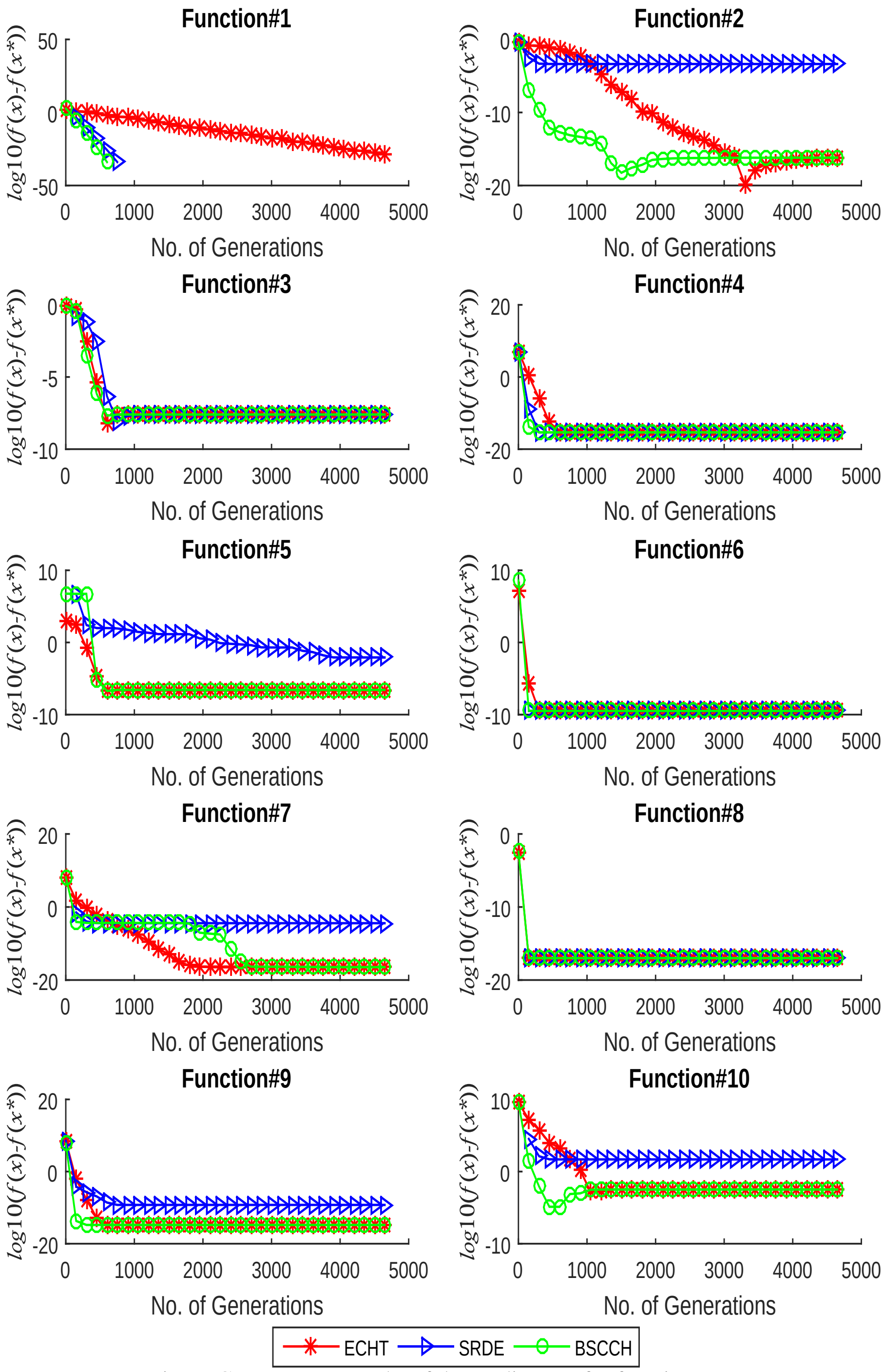


Fig. 7. Convergence graphs of the median run for functions F_1-F_{10}.

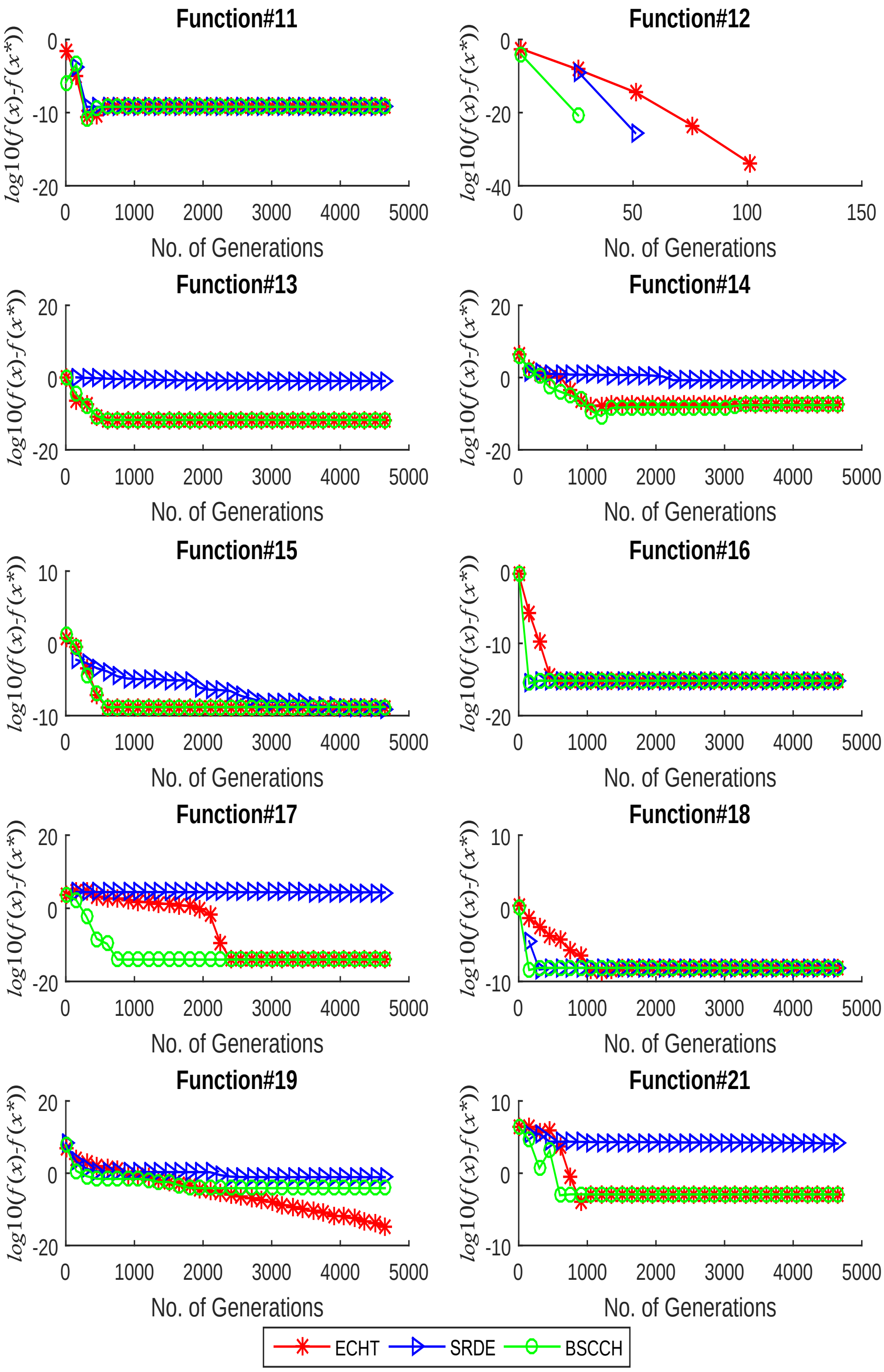


Fig. 8. Convergence graphs of the median run for functions F_{11}-F_{21}.

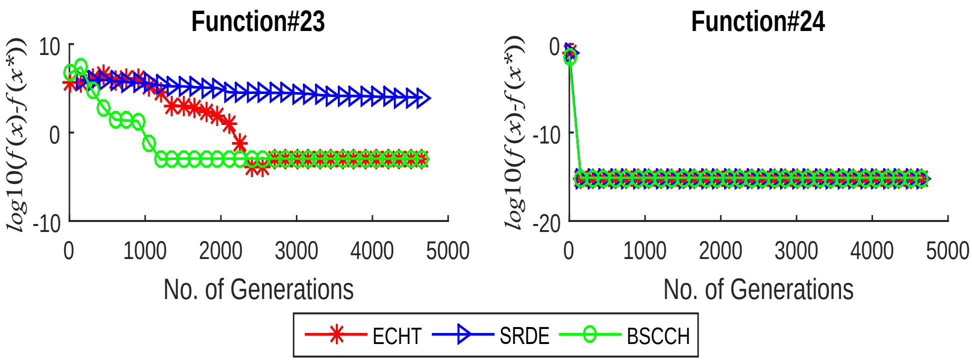


Fig. 9. Convergence graphs of the median run for functions F_{23}-F_{24}.

Table 7. Rank based analysis of mean closure performances.

Algorithm	Mean Ranking of Benchmark Functions																						Rank Sum	Rank Avg.	Rank
	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}	F_{16}	F_{17}	F_{18}	F_{19}	F_{21}	F_{23}	F_{24}			
BSCCH	1	2	1	2	1	1	1	2	2	1	1	1	1	1	2	1	3	1	7	1	1	1	35	1.59	1
CACDE	1	1	1	2	1	1	1	4	2	1	5	1	1	1	4	1	2	1	1	4	6	1	43	1.95	3
ITLBO	1	3	1	7	1	1	1	5	1	6	1	1	4	1	7	8	6	5	5	8	8	6	87	3.95	6
MGABC	8	8	7	6	8	8	7	5	7	8	8	1	7	7	8	1	8	6	1	1	1	6	12	5.77	8
SRDE	1	6	1	2	7	1	1	5	2	1	1	1	8	6	1	1	7	7	6	7	7	6	85	3.86	5
NDE	1	4	1	7	1	1	1	5	2	1	6	1	1	1	5	1	1	4	4	5	1	5	59	2.86	4
CVI-PSO	1	7	8	1	6	1	8	1	8	7	7	1	5	8	6	1	5	8	8	6	5	1	10	4.95	7
ECHT	1	5	1	2	1	1	1	2	2	1	1	1	6	1	2	1	3	1	1	1	1	1	37	1.68	2

Table 8. Rank based analysis of median convergence graphs.

Algorithm	Convergence Rate Ranking of Benchmark Functions																						Rank Sum	Rank Avg.	Rank
	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}	F_{16}	F_{17}	F_{18}	F_{19}	F_{21}	F_{23}	F_{24}			
BSCCH	1	1	1	1	1	1	1	1	1	1	2	1	1	2	1	1	1	1	2	1	1	1	25	1.14	1
SRDE	2	3	3	1	3	1	3	1	3	3	1	2	3	3	3	1	2	2	3	3	3	1	50	2.27	3
ECHT	3	1	1	3	1	3	1	1	2	2	3	3	1	1	1	3	3	3	1	2	2	1	42	1.91	2

The final mean value of all runs for each algorithm have been listed in Tables 3 and 4. The best values of benchmark functions for ITLBO algorithm are not available in literature. However, these results do not contribute significantly as ranking of algorithms have been performance based on mean values. Similarly, values for MGABC algorithm are not available for functions F_{19}, F_{21} and F_{23} (they are assigned best rank, *i.e.* 1 for statistical comparison). The ranking has been performed based on mean value for all algorithms and average rank of each algorithm is calculated to give a clear picture of top performing algorithms. The ranking in Table 7 shows that BSCCH algorithm achieves the best rank *i.e.* 1.55 among eight algorithms. To statistically validate significance of BSCCH algorithm, non-parametric Friedman test has been performed for null hypothesis. Furthermore, pairwise post-hoc Bonferroni test has also been performed. For both tests, 95% confidence interval is used. The computed F-statistic value (F_{stat}) is greater than critical value (F_{crit}), so null hypothesis is rejected. The difference in mean ranks of BSCCH algorithm with ITLBO, MGABC, SRDE and CVI-PSO algorithms are greater than the critical difference (CD) which proves that BSCCH is significantly better than these algorithms as given in Table 5. For other three algorithms *i.e.* CACDE, NDE, ECHT, although, mean difference is less than CD but positive, so BSCCH performance is comparable to them also.

4.3 Median Convergence Trace Comparison

Here, the second top performing algorithm, *i.e.* ECHT with rank 1.68 and another algorithm *i.e.* SRDE have been reproduced. Both these algorithms are DE based. SRDE has been selected because it employs a selection strategy in parent selection stage, therefore convergence rate comparison is performed with an algorithm employing parent selection strategy. The algorithms have been exhausted 25 times for each function. The population size has been set to 50 for all algorithms to have a fair comparison. The maximum number of function evaluations (*Max_Fes*) have been set to 240,000 for each algorithm. Figs. 7-9 show convergence graphs for 22 benchmark functions (functions F_{20} and F_{22} have been omitted from comparison and ranking due to infeasible solution findings in all three algorithms). Convergence graphs have been plotted for the median run of each function, *i.e.* single run of algorithm trace has been selected based on median closure value and its $log10(f(x)-f(x*))$ is plotted against number of generations. The convergence graph ranking has been performed in Table 8, which shows that BSCCH algorithm has significantly improved performance (rank is 1.09) over other two algorithms. The convergence ranking has been done based upon the fact that how quickly an algorithm settles at minimum error ($log10(f(x)-f(x*))$) value, *i.e.* an algorithm stabilized at minimum error value first is ranked 1. Only Function F_{12} converges so fast that it has been displayed for a very small number of generations. The BSCCH algorithm reaches top convergence rank in all problems except functions F_{11},F_{14} and F_{19}, although, BSCCH algorithm is not even degraded to worst in these functions and is ranked 2. This shows the effectiveness in the convergence rate of BSCCH algorithm. To statistically validate significance of BSCCH algorithm, non-parametric Friedman test has been performed for null hypothesis. Furthermore, pair-wise post-hoc Bonferroni test has also been performed. For both tests, 95% confidence interval is used. The computed F-statistic value (F_{stat}) is greater than critical value (F_{crit}), so null hypothesis is rejected. The difference in mean ranks of BSCCH algorithm with selected two algorithms are greater than critical difference (*CD*) which proves that BSCCH is significantly better than these algorithms as given in Table 6.

5. CONCLUSION AND FUTURE WORK

In this paper, a new algorithm has been introduced for solving constrained optimization problems, named as Broadening Selection Competitive Constraint Handling (BSCCH). In BSCCH algorithm, selection of individuals have been performed by broadening selection strategy in parent and survivor selection stages of the competitive paradigm. Although, competitive techniques are very efficient for solving constrained optimization problems, they lack faster convergence due to inadequate selection mechanism. By incorporating efficient selection mechanism in competitive approach, better feasible solution is obtained and convergence rate is improved significantly compared to the selected state-of-the-art algorithms. The proposed algorithm has been tested through a set of 24 benchmark functions by comparing mean closure value and median convergence trace. The statistical comparative analysis clearly validates that the proposed algorithm has comparable mean closure performance. Furthermore, in median convergence rate, the proposed algorithm outperforms the selected algorithms significantly. One of the main shortcomings of BSCCH algorithm is added complexity due to existence of both broadening selection and competitive techniques. In this regard, a merger of broadening selection and competitive techniques efficiently is required to reduce complexity of the algorithm.

REFERENCES

1. N. Jin and Y. Rahmat-Samii, "Advances in particle swarm optimization for antenna designs: Real-number, binary, single-objective and multiobjective implementations," *IEEE Transactions on Antennas and Propagation*, Vol. 55, 2007, pp. 556-567.
2. M. M. Mäkelä, "Survey of bundle methods for nonsmooth optimization," *Optimization Methods and Software*, Vol. 17, 2002, pp. 1-29.
3. I. Fister, X. S. Yang, J. Brest, and D. Fister, "A brief review of nature-inspired algorithms for optimization," *Elektrotehniski Vestnik/Electrotechnical Review*, Vol. 80, 2013, pp. 116-122.
4. R. Storn and K. Price, "Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, Vol. 11, 1997, pp. 341-359.
5. D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm," *Journal of Global Optimization*, Vol. 39, 2007, pp. 459-471.
6. J. Kennedy and R. Eberhart, "Particle swarm optimization," *Natural Computing Series*, Vol. 4, 2018, pp. 97-102.
7. A. M. Rubinov, X. Q. Yang, and A. M. Bagirov, "Penalty functions with a small penalty parameter," *Optimization Methods and Software*, Vol. 17, 2002, pp. 931-964.
8. S. Koziel and Z. Michalewicz, "A decoder-based evolutionary algorithm for constrained parameter optimization problems," in *Proceedings of International Conference on Parallel Problem Solving from Nature*, 1998, pp. 1-10.
9. A. I. Oyman, K. Deb, and H. G. Beyer, "An alternative constraint handling method for evolution strategies," in *Proceedings of Congress on Evolutionary Computation*, Vol. 1, 1999, pp. 612-619.
10. T. P. Runarsson and X. Yao, "Stochastic ranking for constrained evolutionary optimization," *IEEE Transactions on Evolutionary Computation*, Vol. 4, 2000, pp. 284-294.
11. T. Takahama, S. Sakai, and N. Iwane, "Constrained optimization by the epsilon constrained hybrid algorithm of particle swarm optimization and genetic algorithm," in *Advances in Artificial Intelligence*, Vol. 3809, 2005, pp. 389-400.
12. R. d. P. Garcia, B. S. L. P. de Lima, A. C. d. C. Lemonge, and B. P. Jacob, "A rank-based constraint handling technique for engineering design optimization problems solved by genetic algorithms," *Computers & Structures*, Vol. 187, 2017, pp. 77-87.
13. D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, Vol. 1, 1997, pp. 67-82.
14. S. B. Hamida and M. Schoenauer, "Aschea: new results using adaptive segregational constraint handling," in *Proceedings of IEEE Congress on Evolutionary Computation*, Vol. 1, 2002, pp. 884-889.

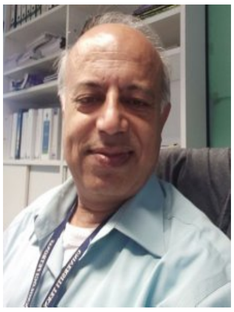
15. P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y. P. Chen, A. Auger, and S. Tiwari, "Problem definitions and evaluation criteria for the CEC 2006 special session on constrained real-parameter optimization," Technical Report, KanGAL, Indian Institute of Technology, 2005, pp. 251-256.
16. P. Verma, K. Sanyal, D. Srinivsan, and K. S. Swarup, "Information exchange based clustered differential evolution for constrained generation-transmission expansion planning," *Swarm and Evolutionary Computation*, Vol. 44, 2019, pp. 863-875.
17. E. C. Osuna, W. Gao, F. Neumann, and D. Sudholt, "Design and analysis of diversity-based parent selection schemes for speeding up evolutionary multi-objective optimisation," *Theoretical Computer Science*, Vol. 1, 2018, pp. 1-20.
18. D. H. Phan, J. Suzuki, and P. Boonma, "SMSP-EMOA: Augmenting SMS-EMOA with the prospect indicator for multiobjective optimization," in *Proceedings of IEEE 23rd International Conference on Tools with Artificial Intelligence*, 2011, pp. 261-268.
19. G. Toscano, R. Landa, G. Lárraga, and G. Leguizamón, "On the use of stochastic ranking for parent selection in differential evolution for constrained optimization," *Soft Computing*, Vol. 21, 2017, pp. 4617-4633.
20. G. Wu, R. Mallipeddi, and P. N. Suganthan, "Ensemble strategies for population-based optimization algorithms – A survey," *Swarm and Evolutionary Computation*, Vol. 44, 2019, pp. 695-711.
21. R. Mallipeddi and P. N. Suganthan, "Ensemble of constraint handling techniques," *IEEE Transactions on Evolutionary Computation*, Vol. 14, 2010, pp. 561-579.
22. M. F. Tasgetiren, P. N. Suganthan, Q. K. Pan, R. Mallipeddi, and S. Sarman, "An ensemble of differential evolution algorithms for constrained function optimization," in *Proceedings of IEEE World Congress on Computational Intelligence*, 2010.
23. S. M. Elsayed, R. A. Sarker, and D. L. Essam, "Integrated strategies differential evolution algorithm with a local search for constrained optimization," in *Proceedings of IEEE Congress on Evolutionary Computation*, 2011, pp. 2618-2625.
24. R. L. Becerra and C. A. C. Coello, "Cultured differential evolution for constrained optimization," *Computer Methods in Applied Mechanics and Engineering*, Vol. 195, 2006, pp. 4303-4322.
25. B. Tessema and G. G. Yen, "A self adaptive penalty function based algorithm for constrained optimization," in *Proceedings of IEEE Congress on Evolutionary Computation*, 2006, pp. 246-253.
26. A. W. Mohamed, "A novel differential evolution algorithm for solving constrained engineering optimization problems," *Journal of Intelligent Manufacturing*, Vol. 29, 2018, pp. 659-692.
27. B. Xu, X. Chen, and L. Tao, "Differential evolution with adaptive trial vector generation strategy and cluster-replacement-based feasibility rule for constrained optimization," *Information Sciences*, Vol. 435, 2018, pp. 240-262.
28. B. C. Wang, H. X. Li, and Y. Feng, "An improved teaching-learning-based optimization for constrained evolutionary optimization," *Information Sciences*, Vol. 456, 2018, pp. 131-144.
29. J. C. Bansal, S. K. Joshi, and H. Sharma, "Modified global best artificial bee colony for constrained optimization problems," *Computers and Electrical Engineering*, Vol. 67, 2018, pp. 365-382.

30. I. Mazhoud, K. Hadj-Hamou, J. Bigeon, and P. Joyeux, "Particle swarm optimization for solving engineering problems: a new constraint-handling mechanism," *Engineering Applications of Artificial Intelligence*, Vol. 26, 2013, pp. 1263-1273.

Tayyab Ahmed Shaikh is currently working as an Assistant Professor at Faculty of Engineering Sciences and Technology, Hamdard University, Karachi, Pakistan. He is currently enrolled in Ph.D. at the Hamdard University and working in the area of evolutionary computation. He is specifically working towards devising an efficient approach for constraint handling in evolutionary algorithms.

Syed Sajjad Hussain is currently working as an Associate Professor at Faculty of Engineering Sciences and Technology, Hamdard University, Karachi, Pakistan. He is also Chairman of Computer Systems Engineering Department at the Hamdard University. He has done Ph.D. in Telecommunication Engineering from Iqra University, Karachi, Pakistan in 2014. His research interest include machine learning, data sciences and evolutionary computation.

Muhammad Rizwan Tanweer is currently working as an Associate Professor and Chairman post-graduate studies at Faculty of Engineering Sciences and Technology, Hamdard University, Karachi, Pakistan. He has done Ph.D. from School of Computer Science and Eingineering, Nanyang Technological University, Singapore in 2017. His research interest include evolutionary computation, specifically efficient incorporation of human learning principles into evolutionary algorithms.



Manzoor Ahmed Hashmani is currently working as an Associate Professor at Faculty of Science and Information Technology, University Technology Petronas, Malaysia. He earned Ph.D. degree from Nara Institute of Science and Technology, Japan in 1999. His research interest include high speed communication networks, artificial intelligence and data sciences.