Constrained self regulating particle swarm optimization

Tayyab Ahmed Shaikh[1], Syed Sajjad Hussain Rizvi[1,2], Muhammad Rizwan Tanweer[1,3]
[1]Faculty of Engineering Science and Technology, Hamdard University, Karachi, Pakistan
[2]Department of Computer Science, Shaheed Zulfikar Ali Bhutto Institute of Science and Technology, Karachi, Pakistan
[3]Department of Computer Science, DHA Suffa University, Karachi, Pakistan

Article Info

Article history:

Received Jul 3, 2021
Revised Dec 17, 2021
Accepted Feb 1, 2022

Keywords:

Constrained optimization
Convergence
Diversity
Self regulation
Swarm intelligence

ABSTRACT

Self regulating particle swarm optimization (SRPSO) is a variant of particle swarm optimization (PSO) which has proved to be a very efficient algorithm for unconstrained optimization compared with other evolutionary algorithms (EAs) and utilized recently by the researchers for solving real-world problems. However, SRPSO has not been evaluated and analyzed for constrained optimization. In this work, SRPSO has been evaluated exhaustively for constrained optimization using the 24 constrained benchmark problems by coupling it with four efficient constraint handling techniques (CHTs). The results of constrained SRPSO algorithm have been compared with two other algorithms i.e. Differential evolution (DE) and PSO. DE and PSO have also been coupled with same four CHTs and evaluated on the 24 constrained benchmark problems. Statistical analysis on performance evaluation of three algorithms on the benchmark problems shows that constrained SRPSO algorithm performance is better than constrained PSO but it is found to be deficient when compared with constrained DE with 95% confidence level. Therefore, the objective of this work is to evaluate the SRPSO algorithm comprehensively for constrained optimization with different views to come up with suitability of constrained SRPSO algorithm when coupled with particular CHT for solving specific type of problems.

Corresponding Author:

Tayyab Ahmed Shaikh
Faculty of Engineering Science and Technology, Hamdard University
Madinat al-Hikmah, Hakim Mohammad Said Road, Karachi-74600, Pakistan
Email: tayyab.ahmed@hamdard.edu.pk

1. INTRODUCTION

Complex real-world engineering problems are difficult to be solved by conventional techniques like linear programming, quadratic programming, non-linear programming and dynamic programming because they involve non-linearity, discontinuous function, and discrete search space. Therefore, evolutionary algorithms (EAs) have been developed for these purposes, which are derived from some natural processes. Constrained optimization is inevitable in various real-world engineering problems. This includes, but not limited to, energy optimization [1], network traffic optimization [2], antenna design optimization [3], image and signal processing optimization [4]. EAs are basically used to solve unconstrained problems, but in case of real-world problems, constraints are encountered. Therefore, to solve constrained problems, constraint handling techniques (CHTs) are required to be coupled with EAs. A constrained optimization problem (COP) can be formulated generally in the form of a nonlinear programming problem [5] as:

Minimize:

$$f(X), X = (x_1, x_1, \dots, x_n) \ and \ X \in S$$

Subject to:

$$g_i(X) \leq 0, i = 1, \ldots, p \quad (1)$$
$$h_j(X) = 0, j = p + 1, \ldots, m$$

where *f(X)* in problem description (1) must not be a continuous function, but bounded. *S* is the complete search space. *p* is the number of inequality constraint and (*m*-*p*) is the number of equality constraints. At the global optimum solution, if inequality constraints satisfy the condition $g_i(X) = 0$, then the constraints are called active constraints. Hence, all the equality constraints are active constraints. The equality constraints are changed into inequality constraints and bundled as described in (2).

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

here δ in (2) is subtracted as a tolerance value to convert equality constraints into inequality constraints. A careful setting of the tolerance value has been proposed and used in various researches [6]–[8]. The objective of constrained optimization is to find the best feasible solution i.e. all inequality constraints should be satisfied. If the solution is infeasible, its overall constraint violation is computed, which is given by (3).

$$v(X) = \frac{\sum_{i=1}^{m} wi(G_i(X))}{\sum_{i=1}^{m} w_i} \quad (3)$$

where $G_i(X)$ in (3) are bundled inequality constraints as given by (2). $w_i = 1/G_{max_i}$ is weight parameter and G_{max_i} is the maximum violation of the constraint obtained so far.

There are several variants of particle swarm optimization (PSO) [9] algorithms based on different strategies such as fully informed particle swarm (FIPS) [10], dynamic multi-swarm particle swarm optimizer (DMSPSO) [11], bare bones particle swarm optimization (BBPSO) [12], unified particle swarm optimization (UPSO) [13], comprehensive learning particle swarm optimization (CLPSO) [14] and self regulating particle swarm optimization (SRPSO) [15]. All these PSO based algorithms are improved version of standard PSO algorithm, however SRPSO has been proved to be the most efficient among them [15]. Differential evolution (DE) [16] algorithm has been reported to be one of the most efficient algorithm for constrained optimization [17] and applied in constrained optimization such as in [18], therefore it has been included in the comparison. SRPSO incorporates human cognition into PSO i.e. human learning principles have been incorporated into the PSO algorithm so that it becomes self-regulating. SRPSO has been proved to be a very efficient algorithm for unconstrained optimization [15] compared to other selected algorithms available in the literature. Therefore it has been utilized by the research community recently [19]–[22] for solving real-world problems, but it has not been extensively evaluated for constrained optimization problems. This work is about evaluating the SRPSO algorithm exhaustively for constrained optimization on 24 benchmark functions [23] using not only a single CHT, but four versatile and efficiently proven CHTs. The analysis of results has been performed to identify the best suited CHT for SRPSO algorithm. Furthermore, shortcomings associated with SRPSO algorithm when solving constrained optimization problems have also been identified together with suitability of SRPSO algorithm when solving particular type of problems.

The rest of the paper is organized is organized as; section 2 presents the research method which contains review of CHTs, EAs, and implementation of CHTs incorporation into EAs. The comparsison among three algorithms i.e. DE, PSO, and SRPSO is presented and analysis has been performed in section 3. Section 4 contains the concluding remarks.

2. RESEARCH METHOD

The research method adopted in this work is based on multiple comparson tests. The four efficient and versatile CHTs from the literature i.e. superiority of feasibility (SF) [24], self adaptive penalty (SP) [25], stochastic ranking (SR) [26] and ε-constraint (EC) [27] have been selected and coupled with three EAs i.e. DE [16], PSO [9] and SRPSO [15] for evaluation of constrained performance evaluation of the three EAs. The reason for the selection of these four CHTs from the literature is their efficiency and diverse nature when compared with each other. SRPSO has been evaluated for constrained optimization since it has not been tested in the literature for constrained optimization. DE is selected for comparison because it is found to be very efficient algorithm for constrained optimization in the literature [17], whereas PSO has been selected being the basic architect of SRPSO algorithm. The ranking of the results of algorithms mean value run is performed among the three EAs for each of the CHT. Statistical validation is peformed for each of the four ranking tables to signify the difference in performance of the three algorithms. Multiple comparison tests have also been performed to identify the corresponding similar performing combinations. A detailed probe into the results of SRPSO for the particular CHT is perfomed to identify the best suited problems for SRPSO algorithm under a particular CHT. In the following subsections, a basic understading of CHTs and EAs used in this work is presented along with the method for incorporating a CHT under particular EA. All the equations of CHTs and EAs can be referred in the corresponding reference mentioned in that subsection.

2.1. Constraint handling techniques

There is variety of CHTs presented in the literature. However, four efficient CHTs have been selected which are versatile in nature when compared with each other in order to evaluate SRPSO algorithm exhaustively with diversified CHTs. The selected CHTs are described in the following subsections.

2.1.1. Superiority of feasibility solutions

SF [24] is based on three simple rules is being as: i) a feasible solution is preferred over an infeasible solution; ii) among two feasible solutions, particle with better fitness is preferred; iii) among two infeasible solutions, particle with a smaller overall constraint violation $\upsilon(X)$ as computed by (3) is preferred.

2.1.2. Self adaptive penalty

Tessema and Yen in SP [25], if there are less feasible individuals, a greater penalty value is added with infeasible particles which have greater amount of constraint violation. However, if there are more feasible individuals, then lesser penalty value is added to infeasible individuals having greater fitness values. The final fitness value for choosing optimum solution is given as $F(X)=d(X)+p(X)$. Here, $p(X)$ is penalty value and $d(X)$ is distance value which is calculated as:

$$d(X) = \begin{cases} v(X), & r_f = 0 \\ \sqrt{f''(X)^2 + v(X)^2}, & otherwise \end{cases} \tag{4}$$

where r_f =(number of feasible individuals)/(population size), $\upsilon(X)$ is the overall constrain violation as defined in (3), $f''(X) = (f\ (X)\ - f_{min})\ /\ (f_{max}\ - f_{min})$. $f_{\max}$ and $f_{\min}$ are maximum and minimum values of objective function $f(X)$ in current combined population. The penalty value is given by $p(X)=(1-r_f)M(X)+r_f N(X)$, where:

$$M(X) = \begin{cases} 0, & r_f = 0 \\ v(X), & otherwise \end{cases}$$

$$N(X) = \begin{cases} 0, & \text{if X is feasible individual} \\ f''(X), & if\ X\ is\ infeasible\ individual \end{cases} \tag{5}$$

2.1.3. Stochastic ranking

In SR method [26], a trade-off between objective and the overall constraint violation is exploited stochastically through a probability factor p_f to evaluate the rank of individuals based on objective function value or overall constraint violation value. The SR technique eliminated the problem with SP technique in which over and under penalization problem occurs due to inappropriate penalty factors. The ranking is based on the following criteria.

if ($\upsilon(X) == 0$ or *rand (0,1)* $< p_f$)
sort based on $f(X)$
else
sort based on $\upsilon(X)$

where $f(X)$ and $\upsilon(X)$ are as given in (1) and (3) respectively.

2.1.4. ε-constraint

In EC technique, constraints are relaxed through ε parameter [27]. EC technique transforms a constrained numerical optimization problem into unconstrained numerical optimization problem. The value of ε is updated for a particular number of generations T_c. If the number of generations exceed T_c, the value of ε is set to zero in order to steer the solution towards zero constraint violation.

$$\varepsilon(0) = v(X_\theta)$$

$$\varepsilon(k) = \begin{cases} \varepsilon(0)\left(1 - \frac{k}{T_c}\right)^{cp}, & 0 < k < T_c \\ 0, & k > T_c \end{cases} \tag{6}$$

here X_θ in (6) is top θ_{th} individual and $\theta=(0.05*N)$. N is the population size. Parameter ranges for T_c and cp are given in [27]. k is the generation counter.

2.2. Evolutionary algorithms

There are variety of EAs in the literature, however DE [16] has been reported to be one the most efficient EA for constrained optimization and used by researchers in most of the cases [17]. Since, DE is strong candidate for constrained optimization, therefore, it has been selected for comparison with SRPSO, whereas, PSO [9] has been added in comparison being the architect of SRPSO [15]. Furthermore, the comparison results in section 3 clearly highlight the superiority of constrained SRPSO over constrained PSO.

2.2.1. Differential evolution

DE is a very efficient heuristic algorithm based on three simple operations [16]. The idea behind development of DE is to produce offspring with as much randomness as possible by using mutation and crossover operations. The best solutions are then retained through selection operation. In mutation, a step towards generation of new individuals is taken by adding weighted difference between two randomly selected individuals to a third random individual as given by (7).

$$V_i(t) = X_{r1}(t) + F(X_{r2}(t) - X_{r3}(t)) \tag{7}$$

The mutant's individuals are then mixed with predetermined target individuals, to produce the trial vector. This mixing is called "crossover" as given by (8). This results in an increase in diversity of the perturbed individuals.

$$Uji(t+1) = \begin{cases} Vji(t+1) \;\; if(randb(j) \leq CR \; or \; j = rnbr(j) \\ Xji(t+1) \;\; if(randb(j) > CR \; and \; j \neq rnbr(j) \end{cases} \tag{8}$$

The trial individuals are compared to the target individuals using greedy criterion to determine whether trial individuals would be selected and replaced by target vector or not. This operation is called selection.

2.2.2. Particle swarm optimization

PSO is a swarm based procedure to obtain the optimum solution. It was introduced in 1995 by Eberhart and Kennedy [9]. It is inspired from the behaviour of bird flock and fish schooling. The birds together search for one piece of food and follow bird nearest to food. In PSO, swarms are randomly initialized in the search space. The particles fly within the search space. Flight of a swarm is influenced by own experience (exploration) and other swarms (exploitation). The velocity and position update equations for PSO are given by (9) and (10) respectively.

$$V(t+1) = \omega * V(t) + c_1 r_1 \big(P_{best} - X(t)\big) + c_2 r_2 \big(G_{best} - X(t)\big) \tag{9}$$

$$X(t+1) = X(t) + V(t+1) \tag{10}$$

where V and X are velocity and position of particles, ω is fixed inertia weight, r_1 and r_2 are random numbers in the range [0, 1], c_1 and c_2 are acceleration coefficients, P_{best} is the best position of each particle and G_{best} is the best position among all the particles.

2.2.3. Self regulating particle swarm optimization

SRPSO algorithm has been developed by incorporating human learning principles into standard PSO algorithm [15]. Research in human learning psychology ascertains that humans are better planners as they continuously regulate their strategies. Self regulation results in better planning. Best planner self regulates according to his own state and global knowledge. The velocity and position update equations for SRPSO are given by (11) and (12) respectively.

$$V(t+1) = \omega * V(t) + c_1 r_1 P_{se} \big(P_{best} - X(t)\big) + c_2 r_2 P_{so} \big(G_{best} - X(t)\big) \tag{11}$$

$$X(t+1) = X(t) + V(t+1) \tag{12}$$

where ω is linearly increasing for best particle and decreasing for other particles, P_{se} is self cognition which is 0 for best particle and 1 for other particles, P_{so} is social cognition and is randomly chosen with 50% confidence level to 0 or 1, rest of the parameters in SRPSO are same as in PSO.

2.3. Implementation of constraint handling techniques incorporation into evolutionary algorithms

The constrained PSO and SRPSO algorithm is concerned with selection of P_{best} and G_{best}. P_{best} is selected by comparing the current P_{best} with generated off-spring according to the rules of particular CHT. If the offspring is better, then P_{best} is updated. G_{best} is the best particle among P_{best} which is also selected according to rules of the particular CHT. G_{best} also becomes the global optimum solution at the final generation. Regarding the DE algorithm, the implementation of CHT comes into play during the selection operation of the DE algorithm. The generated offspring using mutation and crossover operations is kept for the next generation and becomes the parent if it is better than current parent population according to the rules of particular CHT, otherwise the previously generated parent remain in the current population for generating the offspring in next iteration. The best particle in the current population becomes the optimum solution at the final generation. The implementation is simple and it has been verified on constrained benchmark problems.

3. RESULTS AND DISCUSSION

The experiments have been setup based on specifications given in the benchmark functions [23]. A summary of benchmark functions is presented in Table 1. The algorithms have been exhausted 25 times for the 24 benchmark functions on Matlab software. The results of constrained DE algorithm are in-line with the results given in [18]. The population sizes for all algorithms have been set to 50 where as maximum number of generations have been set to 4800. Each of the CHT i.e. SF, SP, SR, EC have been coupled with DE, PSO, and SRPSO algorithms. The parameters for three competing algorithms are $F=0.7$, $CR=0.5$, $c_1=0.5$, $c_2=2$, where as $\omega=0.8$ for PSO and variable for SRPSO i.e. Initial value for $\omega= 1.05$, final value for $\omega= 0.5$. Ranking of the results is based on the following criteria as given in the benchmark specifications [23]. 1) feasible solutions are ranked better than infeasible solutions, 2) feasible solutions are ranked based on minimum fitness difference as compared to best known value, 3) infeasible solutions are ranked based on minimum mean overall constraint violation. The overall constraint violation (CV) value has been used to consider constraints in problem formulation as given by (3). In Table 1, the columns show function No. (Fn), 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 [28].

Table 1. Summary of the benchmark problems used in this work

Fn	f(x*)	N	Type	LI	NLI	LE	NLE
F1	-15.000	13	Quadratic	9	0	0	0
F2	-0.80361910412559	20	Nonlinear	0	2	0	0
F3	-1.00050010001000	10	Polynomial	0	0	0	1
F4	-30665.5386717834	5	Quadratic	0	6	0	0
F5	5126.4967140071	4	Cubic	2	0	0	3
F6	-6961.81387558015	2	Cubic	0	2	0	0
F7	24.30620906818	10	Quadratic	3	5	0	0
F8	-0.0958250414180359	2	Nonlinear	0	2	0	0
F9	680.630057374402	7	Polynomial	0	4	0	0
F10	7049.24802052867	8	Linear	3	3	0	0
F11	0.7499	2	Quadratic	0	0	0	1
F12	-1.000	3	Quadratic	0	1	0	0
F13	0.053941514041898	5	Nonlinear	0	0	0	3
F14	-47.7648884594915	10	Nonlinear	0	0	3	0
F15	961.715022289961	3	Quadratic	0	0	1	1
F16	-1.90515525853479	5	Nonlinear	4	34	0	0
F17	8853.53967480648	6	Nonlinear	0	0	0	4
F18	-0.866025403784439	9	Quadratic	0	12	0	0
F19	32.6555929502463	15	Nonlinear	0	5	0	0
F20	-	24	Linear	0	6	2	12
F21	193.724510070035	7	Linear	0	1	0	5
F22	236.430975504001	22	Linear	0	1	8	11
F23	-400.055099999999584	9	Linear	0	2	3	1
F24	-5.50801327159536	2	Linear	0	2	0	0

3.1. Comparison of DE, PSO, and SRPSO when coupled with SF, SP, SR, and EC for identifying the best EA

The mean fitness value (mean), standard deviation (SD), and mean constraint violation (CV) for SF, SP, SR, and EC have been listed in Tables 2 to 5 whereas their ranking have been performed in Tables 6 to 9 respectively (seen in Appendix). It is evident from the ranking tables that DE performs better than PSO and SRPSO algorithms for all CHTs i.e. DE with SF, SP, SR, and EC achieves best ranks as compared with PSO and

SRPSO except for SF in which DE and SRPSO performs similar. To statistically validate significance of constrained performance of DE algorithm, non-parametric Friedman test followed by pair-wise post-hoc Bonferroni test [29] has also been performed for each CHT as shown in Tables 10 to 13 respectively. For both tests, 95% confidence interval is used. The computed F statistic value (F_{stat}) is greater than critical value (F_{crit}) for all CHTs except SF, so null hypothesis is rejected for SP, SR, and EC. Furthermore, the difference in mean ranks for all CHTs of constrained DE with constrained PSO and constrained SRPSO algorithms are greater than the critical difference (CD). The exceptions for which mean difference rank is less than CD are when SRPSO algorithm is coupled with SF and SP i.e. 0.04 < 0.50 and 0.50 < 0.52 respectively. Therefore, it is concluded that in these cases, constrained SRPSO performance is similar to constrained DE. Moreover, there is very small margin (0.04) in case of SF as compared to SP (0.50). Therefore, it can be concluded that SF with SRPSO is a strong candidate for constrained problems because the four versatile CHTs are enough for evaluating the constrained SRPSO algorithm.

Table 2. Mean fitness, SD, and CV of DE, PSO, and SRPSO with SF for the 24 benchmark functions

Function (best		DE			PSO			SRPSO	
known)	Mean	SD	CV	Mean	SD	CV	Mean	SD	CV
F1(-15)	-12.4364	2.502466	0	-4.36	1.036018	0	-6.83228	1.305682	0
F2(-0.80362)	-0.79238	0.016276	0	-0.68648	0.062665	0	-0.68679	0.106712	0
F3(-1)	-0.01071	0.03005	0.149962	-0.18985	0.23417	0.148455	-0.48089	0.639691	0.210944
F4(-30665.5)	-31325.3	1122.595	0	-29486.8	405.6533	0	-30723.1	1198.687	0
F5(5126.498)	5219.651	946.6683	0.006027	4877.732	1354.467	0.008819	3974.255	940.1045	0.005773
F6(-6961.81)	-6933.85	398.9953	2.59E-08	-6693.97	173.6534	0	-6961.81	6.04E-12	0
F7(24.30621)	24.30621	3.53E-10	0	36.80601	6.730795	0	25.64887	0.841462	0
F8(-0.09583)	-0.09583	1.10E-17	0	-0.09583	5.67E-18	0	-0.09583	0	0
F9(680.6301)	680.6301	3.40E-13	0	680.6585	0.031011	0	680.6584	0.022747	0
F10(7049.331)	7049.248	9.01E-10	0	10234.9	4256.344	0.007786	9302.731	2852.326	0
F11(0.75)	0.992798	0.036008	0	0.953983	0.088969	0.00291	0.922428	0.088512	0.001842
F12(-1)	-1	0	0	-1	0	0	-1	0	0
F13(0.05395)	0.552437	0.419471	0.415903	0.835273	0.252608	0.283881	0.809473	0.569648	0.286918
F14(-47.7644)	65535	65535	10.78784	65535	65535	5.488816	65535	65535	13.95673
F15(961.7152)	963.0761	5.808714	3.497524	964.3291	5.594058	3.201315	963.8202	7.112324	4.884559
F16(-1.90516)	-1.90516	4.53E-16	0	-1.28053	0.403819	0.030798	-1.30832	0.249289	0.030733
F17(8876.981)	9003.642	155.3486	25.95694	8966.49	140.7033	17.49977	8919.62	69.80538	16.82341
F18(-0.86574)	-0.86603	1.17E-09	0	-0.85379	0.039339	0	-0.86011	0.007402	0
F19(32.65559)	32.65559	9.18E-08	0	47.25497	11.61631	0	36.7614	2.93434	0
F20(0.096737)	6.530594	3.335416	2.40E-101	14.97849	4.145621	0.009661	14.95971	2.553402	0.003393
F21(193.7783)	568.1837	297.6404	0.0301	479.2614	139.1603	0.050674	354.1652	146.2568	0.002646
F22(382.9022)	13102.8	6196.402	2.398096	8982.381	5909.016	4.167178	9252.614	3064.041	1.835317
F23(-400.003)	-557.235	613.7539	0.030127	-804.075	510.1871	0.015729	-720.201	507.5817	0.003075
F24(-5.50801)	-5.50801	3.40E-15	0	-5.50801	2.72E-15	0	-5.50801	2.72E-15	0

Table 3. Mean fitness, SD, and CV of DE, PSO, and SRPSO with SP for the 24 benchmark functions

Function (best		DE			PSO			SRPSO	
known)	Mean	SD	CV	Mean	SD	CV	Mean	SD	CV
F1(-15)	-14.3525	1.069752	0	-4.48	0.87178	0	-6.66747	1.4649	0
F2(-0.80362)	-0.79437	0.01077	0	-0.69726	0.062699	0	-0.72699	0.080172	0
F3(-1)	-0.01294	0.031723	0.114725	-0.07786	0.101368	0.090108	-0.93726	2.105189	0.294649
F4(-30665.5)	-33547.8	617.7004	0	-29408.3	878.2032	0	-30452.3	905.1767	0
F5(5126.498)	3948.585	1153.36	0.009847	3279.686	764.054	0.016016	3718.906	1256	0.012615
F6(-6961.81)	-6956.9	2.539893	0	-7315.51	1379.523	0.00033	-4377.75	2963.329	0.025056
F7(24.30621)	24.30712	0.00086	0	35.45715	5.008133	0	26.09779	0.807172	0
F8(-0.09583)	-0.03183	0.038856	0.055451	2.369696	12.30772	0.045029	0.235898	1.481122	0.039053
F9(680.6301)	680.6301	3.16E-13	0	680.6648	0.020196	0	680.6715	0.036586	0
F10(7049.331)	7294.951	54.83747	0	11469.85	4403.133	0.005331	9526.829	2526.22	1.33E-07
F11(0.75)	0.989846	0.050772	0.000232	0.993523	0.022413	0.000754	0.916785	0.080556	0.00141
F12(-1)	-1	0	0	-1	0	0	-1	0	0
F13(0.05395)	0.560414	0.413835	0.390944	0.927177	0.459518	0.292293	0.882828	0.920231	0.307028
F14(-47.7644)	-600.508	78.6155	16.10625	-600.326	102.2095	16.23023	-579.912	81.80742	15.53552
F15(961.7152)	963.1289	6.305416	4.063261	961.3322	8.355632	3.769078	966.3887	7.154036	4.99641
F16(-1.90516)	-1.59044	0.256306	0	-1.04276	0.35944	0.201877	-1.19034	0.284281	0.450135
F17(8876.981)	9003.853	142.7094	30.96555	8990.757	193.768	31.91695	8929.208	116.5034	18.74239
F18(-0.86574)	-0.84201	0.027479	0	-0.82717	0.362107	0.001021	-0.84893	0.071195	0
F19(32.65559)	33.99111	0.264876	0	63.73579	28.78157	0	38.20427	5.304822	0
F20(0.096737)	10.08235	1.651068	4.1E-102	3.335363	3.852258	0.000374	13.78916	3.029195	0.00066
F21(193.7783)	455.7864	256.5569	0.023609	530.8236	221.341	0.053087	383.6921	105.0488	0.002268
F22(382.9022)	10174.84	6337.497	2.071688	7229.996	5026.967	4.038499	9366.282	3320.666	1.776864
F23(-400.003)	-2640.01	822.7776	0.03889	-331.988	591.8494	0.020182	-384.566	542.0023	0.032969
F24(-5.50801)	-5.50801	2.72E-15	0	-5.50801	2.72E-15	0	-5.50801	2.72E-15	0

Table 4. Mean fitness, SD and CV of DE, PSO, and SRPSO with SR for the 24 benchmark functions

Function (Best Known)	DE Mean	DE SD	DE CV	PSO Mean	PSO SD	PSO CV	SRPSO Mean	SRPSO SD	SRPSO CV
F1(-15)	-13.0103	1.6923	0	-4.44	1.386843	0	-7.94773	1.488485	0
F2(-0.80362)	-0.19712	0.015934	0	-0.62442	0.079308	0	-0.41673	0.046221	0
F3(-1)	-0.00454	0.022667	0.0037	-0.0793	0.133033	0.013335	-0.89853	1.526189	0.263529
F4(-30665.5)	-33849.5	12.42595	0	-29485.3	794.2444	0	-30392.2	150.1825	0
F5(5126.498)	134.1894	670.9471	0.00023	5123.157	1184.92	0.002453	4142.449	941.5794	0.00429
F6(-6961.81)	-5475.51	2167.583	8.53E-15	-7030.35	1896.147	0.000159	-5921.14	2419.561	0.010841
F7(24.30621)	24.48388	0.076643	0	89.078	185.4835	0.00122	73.33224	232.6751	0.010564
F8(-0.09583)	-0.09583	1.2E-17	0	-0.09583	7.49E-18	0	-0.09583	2.83E-18	0
F9(680.6301)	680.6301	3.47E-06	0	680.8526	0.156221	0	680.736	0.091668	0
F10(7049.331)	7156.786	40.42441	0	8716.707	4335.267	0.005914	8286.578	1796.72	1.86E-05
F11(0.75)	0.860305	0.116336	0	0.978189	0.05159	7.09E-06	0.89946	0.074143	0.000771
F12(-1)	-1	0	0	-1	0	0	-1	0	0
F13(0.05395)	0.824238	0.767846	0.09026	0.830601	0.618224	0.075564	0.984808	1.130536	0.037303
F14(-47.7644)	65535	65535	6.35359	65535	65535	3.316417	65535	65535	14.4995
F15(961.7152)	965.7359	4.300763	0.719579	961.5763	4.045743	0.958437	965.5757	3.38638	0.767159
F16(-1.90516)	-1.90516	6.66E-16	0	-0.9586	0.45069	0.009398	-1.44565	0.242351	0.04682
F17(8876.981)	8954.244	66.9859	17.26602	9030.771	111.3835	25.6206	8904.431	45.76783	10.95915
F18(-0.86574)	-0.7504	0.071398	0	-1.7533	1.96835	0.013182	-0.90486	0.320363	0.000113
F19(32.65559)	33.26251	0.271567	0	56.03146	17.43594	0	41.52282	5.324627	0
F20(0.096737)	6.2342	2.379899	1.2E-101	2.463069	2.713516	0.000821	12.78582	2.528382	0.000468
F21(193.7783)	0	0	1.9E-10	545.5506	190.508	0.097395	433.9268	181.5699	0.016669
F22(382.9022)	0	0	0.021383	5345.159	3704.681	3.48975	5040.805	2682.456	0.339444
F23(-400.003)	-3041.85	1427.239	0.000632	-247.172	492.6405	0.010224	-588.604	358.556	0.004287
F24(-5.50801)	-5.50801	2.72E-15	0	-5.50801	2.72E-15	0	-5.50801	2.72E-15	0

.

Table 5. Mean fitness, SD, and CV of DE, PSO, and SRPSO with EC for the 24 benchmark functions

Function (Best Known)	DE Mean	DE SD	DE CV	PSO Mean	PSO SD	PSO CV	SRPSO Mean	SRPSO SD	SRPSO CV
F1(-15)	-14.3563	1.183441	0	-4.76	0.925563	0	-6.11632	1.071534	0
F2(-0.80362)	-0.79238	0.016276	0	-0.6932	0.061398	0	-0.67331	0.121804	0
F3(-1)	-0.01071	0.03005	0.149962	-0.14721	0.187942	0.113368	-0.78411	1.188818	0.201064
F4(-30665.5)	-31325.3	1122.595	0	-29712	592.019	0	-30705.9	1125.215	0
F5(5126.498)	5133.714	981.7052	0.005989	4512.796	1152.576	4.83798	3876.284	659.3796	0.009583
F6(-6961.81)	-6933.85	398.9953	2.59E-08	-7293.36	1362.639	0.000929	-4171.39	3212.246	0.042969
F7(24.30621)	24.30621	5.31E-09	0	264.3745	533.246	0.001091	25.4833	0.778905	0
F8(-0.09583)	-0.09583	1.06E-17	0	-0.09583	4.91E-18	0	-0.09583	0	0
F9(680.6301)	680.6301	3.32E-13	0	680.6969	0.043265	0	680.6644	0.0305	0
F10(7049.331)	7049.248	2.99E-08	0	9668.624	3849.549	0.008153	10007.8	3351.308	2.05E-06
F11(0.75)	0.992798	0.036008	0	0.970284	0.068685	0.001345	0.90767	0.080384	0.001471
F12(-1)	-1	0	0	-1	0	0	-1	0	0
F13(0.05395)	0.533317	0.409617	0.415205	0.908666	0.201244	0.377034	0.789345	0.230082	0.372937
F14(-47.7644)	65535	65535	10.78784	65535	65535	6.283597	65535	65535	15.64636
F15(961.7152)	963.0761	5.808714	3.497524	963.4342	4.949284	3.120307	966.2169	6.753518	3.558619
F16(-1.90516)	-1.90516	4.53E-16	0	-1.26721	0.387681	0.119457	-1.36218	0.249364	0.010487
F17(8876.981)	9038.18	165.8207	27.75915	9059.186	153.1178	28.82786	8930.053	99.96802	17.15935
F18(-0.86574)	-0.86603	4.47E-08	0	-1.0607	0.859783	0.000103	-0.86076	0.007094	0
F19(32.65559)	32.65559	9.18E-08	0	54.48409	17.55147	0	36.5076	2.597452	0
F20(0.096737)	6.530594	3.335416	2.4E-101	5.048785	4.137055	0.00082	11.89985	4.615015	0.00114
F21(193.7783)	400.8797	359.9267	0.027802	511.2833	243.6481	0.063354	363.8738	131.4688	0.004926
F22(382.9022)	2352.393	4957.482	0.69517	7051.872	6477.757	4.105713	9101.561	3352.511	1.866953
F23(-400.003)	-3175.53	848.1321	0.000532	-169.829	353.1054	0.011828	-846.372	460.6264	0.004514
F24(-5.50801)	-5.50801	3.4E-15	0	-5.50801	2.72E-15	0	-5.50801	2.72E-15	0

Table 10. Statistical validation test for SF with DE, PSO, and SRPSO

F_{stat} (3.16) < F_{crit} (3.20) with CD=0.52	Algorithm PSO	Algorithm SRPSO
Mean Rank Diff w.r.t. DE	0.58	0.04

Table 11. Statistical validation test for SP with DE, PSO, and SRPSO

F_{stat} (4.35) > F_{crit} (3.20) with CD=0.52	Algorithm PSO	Algorithm SRPSO
Mean Rank Diff w.r.t. DE	0.75	0.50

Table 12. Statistical validation test for SR with DE, PSO, and SRPSO

F_{stat} (12.23) > F_{crit} (3.20) with CD=0.45	Algorithm PSO	Algorithm SRPSO
Mean Rank Diff w.r.t. DE	1.04	0.83

Table 13. Statistical validation test for EC with DE, PSO, and SRPSO

F_{stat} (10.31) > F_{crit} (3.20) with CD=0.46	Algorithm PSO	Algorithm SRPSO
Mean Rank Diff w.r.t. DE	1.00	0.75

3.2. Analysis from the test results

Since SF has been identified to be the best CHT for SRPSO algorithm, detailed analysis by probing into each benchmark function for SF case in Table has been performed. It has been found that constrained SRPSO behaves well for functions having less number of inequality constraints (equality constraints may be higher) like F5, F8, F9, F17, F21, F22, and F23. For some non-linear functions such as: F6, F11, and F16 constrained SRPSO algorithm violate constraints as compared to DE. For other most of the functions, constrained SRPSO algorithm converges prematurely. This is due to lack of diversity in SRPSO algorithm because of following the leader (P_{best} and G_{best}) as mentioned in (11) and (12), since if the leaders get trapped in local minima or infeasible region, they cannot leave. This is not the case with DE as a lot of diversity is present due to crossover and mutation operations as expressed in (7) and (8) where there is no leader. Therefore, DE has found to be the best candidate for constrained optimization in most of the problems. However, when adapting the SRPSO algorithm for constrained optimization, one must use SF technique. SF is the simplest technique to implement and based on three simple rules which direct the search towards the feasible region.

4. CONCLUSION

In this work, SRPSO algorithm has been evaluated for constrained problems. To comprehensively evaluate the performance of constrained SRPSO algorithm, not a single, but four versatile CHTs have been selected from the literature namely SF, SP, SR, and EC. SRPSO algorithm has been coupled with these four CHTs and its performance has been evaluated on 24 benchmark problems by comparing it with PSO and DE algorithms which have also been coupled with the same four CHTs. The following can be concluded from this work: It has been found through statistical analysis with 95% confedence level that DE has the best performance compared to PSO and SRPSO except when SRPSO is coupled with SP and SF for which the performance is similar, specifically the difference in performance is negligible between DE and SRPSO under SF. Whenever it comes to SRPSO algorithm, SF technique has been emerged as the best CHT for constrained problems. Furthermore, it has been found by analyzing results of SRPSO with SF technique combination that, it is the most suitable choice for constrained problems having linear objective function with higher number of equality constraints and lower number of inequality constraints. Some tuning of parameters and cognition strategies can be applied into SRPSO algorithm to further improve its performance specifically for CHTs other than SF.

APPENDIX

Table 6. Ranking of DE, PSO, and SRPSO with SF for the 24 benchmark functions

F_n	DE	PSO	SRPSO
F_1	1	3	2
F_2	1	3	2
F_3	2	1	3
F_4	1	3	2
F_5	2	3	1
F_6	3	2	1
F_7	1	3	2
F_8	1	1	1
F_9	3	2	1
F_{10}	1	3	2
F_{11}	1	3	2
F_{12}	1	1	1
F_{13}	3	1	2
F_{14}	2	1	3
F_{15}	2	1	3
F_{16}	1	3	2
F_{17}	3	2	1
F_{18}	1	3	2
F_{19}	1	3	2
F_{20}	1	3	2
F_{21}	2	3	1
F_{22}	2	3	1
F_{23}	3	2	1
F_{24}	1	1	1
Sum	40	54	41
Avg	1.67	2.25	1.71

Table 7. Ranking of DE, PSO, and SRPSO with SP for the 24 benchmark functions

F_n	DE	PSO	SRPSO
F_1	1	3	2
F_2	1	3	2
F_3	2	1	3
F_4	1	3	2
F_5	1	3	2
F_6	1	2	3
F_7	1	3	2
F_8	3	2	1
F_9	1	2	3
F_{10}	1	3	2
F_{11}	1	2	3
F_{12}	1	1	1
F_{13}	3	1	2
F_{14}	2	3	1
F_{15}	2	1	3
F_{16}	1	2	3
F_{17}	2	3	1
F_{18}	1	3	2
F_{19}	1	3	2
F_{20}	1	2	3
F_{21}	2	3	1
F_{22}	2	3	1
F_{23}	3	1	2
F_{24}	1	1	1
Sum	36	54	48
Avg	1.5	2.25	2

Table 8. Ranking of DE, PSO, and SRPSO with SR for the 24 benchmark functions

F_n	DE	PSO	SRPSO
F_1	1	3	2
F_2	3	1	2
F_3	1	2	3
F_4	1	3	2
F_5	1	2	3
F_6	1	2	3
F_7	1	2	3
F_8	1	1	1
F_9	1	3	2
F_{10}	1	3	2
F_{11}	1	2	3
F_{12}	1	1	1
F_{13}	3	2	1
F_{14}	2	1	3
F_{15}	1	3	2
F_{16}	1	2	3
F_{17}	2	3	1
F_{18}	1	3	2
F_{19}	1	3	2
F_{20}	1	3	2
F_{21}	1	3	2
F_{22}	1	3	2
F_{23}	1	3	2
F_{24}	1	1	1
Sum	30	55	50
Avg	1.25	2.29	2.08

Table 9. Ranking of DE, PSO, and SRPSO with EC for the 24 benchmark functions

F_n	DE	PSO	SRPSO
F_1	1	3	2
F_2	1	2	3
F_3	2	1	3
F_4	1	3	2
F_5	1	3	2
F_6	1	2	3
F_7	1	3	2
F_8	1	1	1
F_9	1	3	2
F_{10}	1	3	2
F_{11}	1	2	3
F_{12}	1	1	1
F_{13}	3	2	1
F_{14}	2	1	3
F_{15}	2	1	3
F_{16}	1	3	2
F_{17}	2	3	1
F_{18}	1	3	2
F_{19}	1	3	2
F_{20}	1	2	3
F_{21}	2	3	1
F_{22}	1	3	2
F_{23}	1	3	2
F_{24}	1	1	1
Sum	31	55	49
Avg	1.29	2.29	2.04

REFERENCES

[1] R. Banos, F. Manzano-Agugliaro, F. G. Montoya, C. Gil, A. Alcayde, and J. Gómez, "Optimization methods applied to renewable and sustainable energy: A review," *Renewable and sustainable energy reviews*, vol. 15, no. 4, pp. 1753–1766, May 2011, doi: 10.1016/j.rser.2010.12.008.

[2] M. Josefsson and M. Patriksson, "Sensitivity analysis of separable traffic equilibrium equilibria with application to bilevel optimization in network design," *Transportation Research Part B: Methodological*, vol. 41, no. 1, pp. 4–31, January 2007, doi: 10.1016/j.trb.2005.12.004.

[3] N. Jin and Y. Rahmat-Samii, "Advances in Particle Swarm Optimization for Antenna Designs: Real-Number, Binary, Single-Objective and Multiobjective Implementations," in *IEEE Transactions on Antennas and Propagation*, vol. 55, no. 3, pp. 556–567, March 2007, doi: 10.1109/TAP.2007.891552.

[4] M. Zibulevsky and M. Elad, "L1-L2 optimization in signal and image processing," in *IEEE Signal Process. Mag.*, vol. 27, no. 3, pp. 76–88, May 2010, doi: 10.1109/MSP.2010.936023.

[5] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential Evolution Algorithm With Strategy Adaptation for Global Numerical Optimization," in *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, April 2009, doi: 10.1109/TEVC.2008.927706.

[6] E. Mezura-Montes and C. A. C. Coello, "Adding a diversity mechanism to a simple evolution strategy to solve constrained optimization problems," *The 2003 Congress on Evolutionary Computation, 2003. CEC '03.*, vol. 1, pp. 6–13, 2003, doi: 10.1109/CEC.2003.1299550.

[7] E. Mezura-Montes and C. A. C. Coello, "A simple multimembered evolution strategy to solve constrained optimization problems," in *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 1, pp. 1–17, Feb. 2005, doi: 10.1109/TEVC.2004.836819.

[8] Y. Wang, Z. Cai, Y. Zhou, and W. Zeng, "An Adaptive Tradeoff Model for Constrained Evolutionary Optimization," in *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 80–92, Feb. 2008, doi: 10.1109/TEVC.2007.902851.

[9] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Proceedings of ICNN'95-International Conference on Neural Networks*, vol. 4, pp. 1942–1948, 1995, doi: 10.1109/ICNN.1995.488968.

[10] R. Mendes, J. Kennedy, and J. Neves, "The fully informed particle swarm: simpler, maybe better," in *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 204–210, June 2004, doi: 10.1109/TEVC.2004.826074.

[11] J. J. Liang and P. N. Suganthan, "Dynamic multi-swarm particle swarm optimizer," *Proceedings 2005 IEEE Swarm Intelligence Symposium, 2005. SIS 2005.*, 2005, pp. 124–129, doi: 10.1109/SIS.2005.1501611.

[12] J. Kennedy, "Bare bones particle swarms," *Proceedings of the 2003 IEEE Swarm Intelligence Symposium. SIS'03 (Cat. No.03EX706)*, 2003, pp. 80–87, doi: 10.1109/SIS.2003.1202251.

[13] K.E. Parsopoulos and M. N. Vrahatis, "UPSO: A unified particle swarm optimization scheme," *International Conference of Computational Methods in Sciences and Engineering 2004 (ICCMSE 2004)*, vol. 1, pp. 868–873, 2004, doi: 10.1201/9780429081385.

[14] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," in *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 281–295, June 2006, doi: 10.1109/TEVC.2005.857610.

[15] M. R. Tanweer, S. Suresh, and N. Sundararajan, "Self regulating particle swarm optimization algorithm," *Information Sciences*, vol. 294, pp. 182–202, February 2015, doi: 10.1016/j.ins.2014.09.053.

[16] R. Storn and K. Price, "Differential Evolution–A Simple and Efficient Heuristic for global Optimization over Continuous Spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, December 1997, doi: 10.1023/A:1008202821328.

[17] 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, pp. 863–875, February 2019, doi: 10.1016/j.swevo.2018.09.009.
[18] R. Mallipeddi and P. N. Suganthan, "Ensemble of Constraint Handling Techniques," in *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 4, pp. 561-579, Aug. 2010, doi: 10.1109/TEVC.2009.2033582.
[19] A. L. Ara, N. M. Shahi, and M. Nasir, "CHP Economic Dispatch Considering Prohibited Zones to Sustainable Energy Using Self-Regulating Particle Swarm Optimization Algorithm," *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, vol. 44, no. 3, pp. 1147–1164, 2020, doi: 10.1007/s40998-019-00293-5.
[20] Y. Fan, C. Zhang, Y. Xue, J. Wang, and F. Gu, "A Bearing Fault Diagnosis Using a Support Vector Machine Optimised by the Self-Regulating Particle Swarm," *Shock and Vibration*, vol. 2020, 2020, doi: 10.1155/2020/9096852.
[21] R. Ganotra, S. Dora, and S. Gupta, "Identifying brain regions contributing to Alzheimer's disease using self regulating particle swarm optimization," *International Journal of Imaging Systems and Technology*, vol. 31, no. 1, pp. 106–117, June 2020, doi: 10.1002/ima.22458.
[22] R. George and P. Samuel, "Fixing state change inconsistency with self regulating particle swarm optimization," Soft Computing, vol. 24, no. 24, pp. 18937–18952, 2020, doi: 10.1007/s00500-020-05124-y.
[23] J. J. Liang *et al.*, "Problem definitions and evaluation criteria for the CEC 2006 special session on constrained real-parameter optimization," *Journal of Applied Mechanics*, vol. 41, no. 8, pp. 8–31, 2006.
[24] K. Deb, "An efficient constraint handling method for genetic algorithms," *Computer methods in applied mechanics and engineering*, vol. 186, no. 2–4, pp. 311–338, June 2000, doi: 10.1016/S0045-7825(99)00389-8.
[25] B. Tessema and G. G. Yen, "A Self Adaptive Penalty Function Based Algorithm for Constrained Optimization," *2006 IEEE International Conference on Evolutionary Computation*, 2006, pp. 246–253, doi: 10.1109/CEC.2006.1688315.
[26] T. P. Runarsson and X. Yao, "Stochastic ranking for constrained evolutionary optimization," in *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 3, pp. 284–294, Sept. 2000, doi: 10.1109/4235.873238.
[27] T. Takahama and S. Sakai, "Constrained Optimization by the ε Constrained Differential Evolution with Gradient-Based Mutation and Feasible Elites," *2006 IEEE International Conference on Evolutionary Computation*, 2006, pp. 1–8, doi: 10.1109/CEC.2006.1688283.
[28] T. A. Shaikh, S. S. Hussain, M. R. Tanweer, and M. A. Hashmani, "Broadening Selection Competitive Constraint Handling Algorithm for Faster Convergence," *Journal of Information Science & Engineering*, vol. 36, no. 6, pp. 1293–1314, 2020.
[29] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *The Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

BIOGRAPHIES OF AUTHORS

Tayyab Ahmed Shaikh is currently working as 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. He can be contacted at email: tayyab.ahmed@hamdard.edu.pk.

Syed Sajjad Hussain is currently working as Associate Professor in Department of Computer Science, Shaheed Zulfikar Ali Bhutto Institute of Science and Technology, Karachi, Pakistan. He did Ph.D in Telecommunication Engineering from Iqra University, Karachi, Pakistan in 2014. His research interests include machine learning, data sciences and evolutionary computation. He can be contacted at email: sshussainr@gmail.com

Muhammad Rizwan Tanweer is currently working as Assistant Professor and Chairman Department of Computer Science in DHA Suffa University, Karachi, Pakistan. He did Ph.D in Computer Engineering from Nanyang Technological University, Singapore in 2017. His research interests include evolutionary computation, specifically efficient incorporation of human learning principles into evolutionary algorithms. He can be contacted at email: rizwantanweer85@gmail.com