SRDE: An Improved Differential Evolution Based on Stochastic Ranking

Jinchao Liu
Technical University of Denmark
Nils Koppels Alle
Kgs. Lyngby 2800, Denmark
+45 4525 5602
jliu@man.dtu.dk

Zhun Fan
Technical University of Denmark
Nils Koppels Alle
Kgs. Lyngby 2800, Denmark
+45 4525 6271
zf@mek.dtu.dk

Erik Goodman
2120 Engineering Building, MSU
East Lansing, MI, 48824, USA
+01 517 355 6453
goodman@egr.msu.edu

ABSTRACT
In this paper, we propose a methodology to improve the performance of the standard Differential Evolution (DE) in constraint optimization applications, in terms of accelerating its search speed, and improving the success rate. One critical mechanism embedded in the approach is applying Stochastic Ranking (SR) to rank the whole population of individuals with both objective value and constraint violation to be compared. The ranked population is then in a better shape to provide useful information e.g. direction to guide the search process. The performance of the proposed approach, which we call SRDE (Stochastic Ranking based Differential Evolution) is investigated and compared with standard DE with two variants of mutation strategies. The experimental results show that SRDE outperforms, or at least is comparable with standard DE in both variants in all the tested benchmark functions.

Categories and Subject Descriptors
I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search – *Heuristic methods*.

General Terms
Algorithms.

Keywords
Differential evolution, stochastic ranking, constrained optimization

1. INTRODUCTION
Since it advent [1], DE has been considered a powerful constrained optimization tool. It has been applied in a large variety of engineering optimization applications [2]. Many researchers have also proposed different ways to improve the performance of standard DE, in terms of its search speed and successful rate.

Noman and Iba[7] proposed a hybrid DE which adopted an adaptive local search algorithm to accelerating DE. Combining several strategies, Qin and Suganthan[13] proposed a self-adaptive DE(SaDE), SaDE chooses different strategies and adjusts the control parameters of DE according to previous search information. Sun et al.[14] proposed EDA that combines the standard DE with the estimation of distribution algorithm. EDA has been used to locate the most promising area to guide the search. Rahnamayan et al [6] proposed OBL which employs opposition-based learning to initialize the population and for the population to generate jumping in each generation to accelerate the convergence speed of DE.

```
Initialize
    Generate the initial generation P_0 , t= 1 ;
While termination criteria not satisfied do:
    Evaluate population P_t .
    For each individual x^t in P_t
        Generate a trial vector v^t by mutation and crossover
        If v^t is better than x^t then
            x^t ← v^t ;
        End if
    End Foreach
    t ← t + 1;
End while
```

Fig. 1. Pseudo-code of standard DE

This paper proposes a simple, but very useful mechanism that can accelerate the search speed of standard DE. From empirical study, it is also shown that it can increase the successful rate. In this approach, stochastic Ranking [3] is first applied to rank the whole population, with both objective value and level of constraint violation as comparison criteria. The ranked population is then able to provide valuable directional information to guide the search of DE. The resulting algorithm, which we call SRDE (Stochastic ranking based Differential Evolution) can consistently outperforms, or at least is comparable to standard DE in a comprehensive test with 24 benchmark problems [15].The remainder of the paper is organized as follows. Section 2 briefly reviews standard DE. Section 3 explains the idea and algorithm flow of SRDE. Section 4 presents experimental verification. Finally, the work is concluded in section 5.

2. DIFFERENTIAL EVOLUTION

2.1 Standard DE

As a population -based evolutionary algorithm, DE has been used to solve a large number of real-parameter optimization problems. Unlike other Evolutionary Algorithms (EAs), DE employs difference of individuals in generation of new individuals. In each iteration, a trial vector is generated by combining a parental individual and the difference vector(s) of several other individuals in the same population. If this trial vector is better than the parental individual, it will become a new offspring and go into the next generation.

Fig. 1 shows the pseudo code of the standard DE. As can be seen, DE first creates an initial population distributed randomly over the whole search space. Then, in each generation, DE creates a trial vector for each individual according to a certain mutation strategy. A crossover operation is then executed, followed by a selection operation, in which DE applies a *Knock-out* competition to select the survivor from the parental individual and its corresponding trial vector. The winner, then, goes into the new generation. This scheme implicitly implements 'elite reserve' model, which typically is very important for optimization.

Many variants of standard DE have been proposed, including the following five different mutation strategies:

"rand/1": $\vec{v}_i^T = \vec{x}_{r1}^T + F(\vec{x}_{r2}^T - \vec{x}_{r3}^T)$

"best/1": $\vec{v}_i^T = \vec{x}_{best}^T + F(\vec{x}_{r1}^T - \vec{x}_{r2}^T)$

"current to best/1":

$$\vec{v}_i^T = \vec{x}_i^T + F(\vec{x}_{best}^T - \vec{x}_i^T) + F(\vec{x}_{r1}^T - \vec{x}_{r2}^T)$$

"rand/2":

$$\vec{v}_i^T = \vec{x}_{r1}^T + F(\vec{x}_{r2}^T - \vec{x}_{r3}^T) + F(\vec{x}_{r4}^T - \vec{x}_{r5}^T)$$

"best/2":

$$\vec{v}_i^T = \vec{x}_{best}^T + F(\vec{x}_{r1}^T - \vec{x}_{r2}^T) + F(\vec{x}_{r3}^T - \vec{x}_{r4}^T)$$

where the indexes r_1, r_2, r_3, r_4 and r_5 represent five different integers generated within range $[1, Np]$ at random, which should also not be equal to i. $\vec{x}_{best}^T$ denotes the 'best' individual at generation T.

2.2 Different Rules of Handling Constraints

Very few constraint handling techniques have been reported in differential evolution for constrained optimization applications. Two very important and similar techniques are proposed by Lampinen [16] and Becerra and Coello [17]. Both techniques use three rules for the replacement during the selection procedure, and first two are the same. They are:

- A feasible individual is always better than an infeasible individual.
- If both individuals are feasible, the one with better value of the objective function is selected for the next generation.

The third rule, regarding the situation when both individuals are infeasible, is different. In Lampinen's approach, the comparison is made in the Pareto sense in the constraint violation space. It can be expressed as:

- If both individuals are infeasible, the parent is replaced if the new individual has lower or equal violation for all the constraints.

In Becerra and Coello's approach, a sum of normalized constraint violation is used for comparison, and can be written as:

- If both individuals are infeasible, the individual with less level of constraint violations is better. The level of constraint violation is measured with normalized constraints with the expression of $viol(x_j) = \sum_{c=1}^{constr} \frac{g_c(x)}{g_{\max c}}$, where $g_c(x)$ are the violated constraints of the problem, and $g_{\max c}$ the largest violation of the constraint $g_c(x)$ found so far.

It is worthwhile to point out that both approaches bear some resemblance with an approach proposed by Deb [18] previously, even though Deb's approach is not based in differential evolution. The key difference also lies in the comparison for the case of two infeasible individuals: Lampinen's method makes the comparison in the Pareto sense, Deb' method sums all the constraint violations and compares a single value, Becerra and Coello's method makes normalization for the constraints violations before summing them together.

3. SRDE: STOCHASTIC RANKING BASED DIFFERENTIAL EVOLUTION

3.1SRDE/rand/1

The 'rand/1' mutation strategy used in standard DE provides no information of direction towards the global optimum. If the information of direction can be obtained and utilized in the search process, the performance of the algorithm has a potential to be improved. To avoid the search to be stuck in local minimum, however, the direction information should not be local, but global. To define a 'global direction' information for the whole population is not an easy task, especially when each individual has actually two features to compare with others in a constraint optimization problem – one feature is objective value, the other is level of constraint violation. How to optimally balance them in the comparison procedure presents a challenge.

Stochastic Ranking (SR) [3] provides a convenient and powerful mechanism to balance the dominance in ranking the whole population with both objective value and constraint violation as comparison criteria. The pseudo code of SR is provided in Fig. 3.

The improved DE algorithm, SRDE is designed with a focus on a modified mutation strategy, which can be described in more details as the following: for generation of trial vectors, the whole population is first made to undergo a stochastic ranking procedure. Then the ranked population is divided into two parts –

1. **Initialize**:
2. Parameters: N_P , F , p_f , p_{CR} , γ;where N_P denotes the size of population; F denotes scaling factor; p_f is a parameter used in stochastic ranking. p_{CR} denotes the probability of crossover, γ represents the number of individuals in the upper part of the population Q_1 .
3. Generate the initial generation P_0 , $t = 0$;
4. **While** termination criteria not satisfied **do:**
5. Evaluate population $P_t : (f, \varphi) = eval(P_t)$; where f, φ denote objective and violation of constraints, respectively.
6. Rank population using stochastic ranking: $I = stochastic\_rank(f, \varphi, p_f)$;
7. Divide population into two sets:
8. $Q_1^t = \{x_{I(1)}^t \quad x_{I(2)}^t \quad \dots \quad x_{I(\gamma)}^t\}$;
9. $Q_2^t = \{x_{I(\gamma+1)}^t \quad x_{I(\gamma+2)}^t \quad \dots \quad x_{I(\lambda)}^t\}$;
10. **For** $k = 1$ to N_P **do**
11. Select $x_{r_1}^t \in P_t$, $x_{r_2}^t \in Q_1^t$, $x_{r3}^t \in Q_2^t$ at random;
12. $u_k^t \leftarrow x_{r_1}^t + F \times \left(x_{r_2}^t - x_{r_3}^t \right)$;
13. $v_k^t \leftarrow crossover\left(u_k^t, x_k^t, P_{CR}\right)$;
14. **If** v_k^t is better than x_k **then** $x_k^t \leftarrow v_k^t$;
15. **End if**
16. **End for**
17. $t \leftarrow t + 1$;
18. **End while**

---

1. **Initialize**:
2. Parameters: N_P , F , p_f , p_{CR} ;where N_P denotes the size of population; F denotes the scaling factor; p_f is a parameter used in stochastic ranking. p_{CR} denotes the probability of crossover.
3. Generate the initial generation P_0 , $t = 0$;
4. **While** termination criteria not satisfied **do:**
5. Evaluate population $P_t : (f, \varphi) = eval(P_t)$; where f, φ denote objective and violation of constraints, respectively.
6. Rank population using stochastic ranking: $I = stochastic\_rank(f, \varphi, p_f)$;
7. **For** $k = 1$ to N_P **do**
8. Select $x_{r_1}^t = x_{I(1)}^t$ /*the 'best' individual in current population in Stochastic Ranking sense.*/
9. Select $x_{r_2}^t$, $x_{r_3}^t \in P_t$ at random;
10. $u_k^t \leftarrow x_{r_1}^t + F \times \left(x_{r_2}^t - x_{r_3}^t \right)$;
11. $v_k^t \leftarrow crossover\left(u_k^t, x_k^t, P_{CR}\right)$;
12. **If** v_k^t is better than x_k **then** $x_k^t \leftarrow v_k^t$;
13. **End if**
14. **End for**
15. $t \leftarrow t + 1$;
16. **End while**

Fig. 2. Pseudo-code of iterative search procedure of SRDE/best/1/bin

upper part and lower part. The upper part comprises of the 'better' individuals who have been ranked high after stochastic ranking procedure. For each individual trial vector, the base individual is selected randomly from the whole population, and the second, third individuals are selected from the upper part and lower part, respectively. The three individuals then make a mutation operation according to 'rand/1' strategy, with the difference vector obtained through extracting one 'good' individual with the 'less-good' individual. It is notable that in this way the difference vector will always be directed towards the upper part of the population, thus leading the population to search upwards (Fig. 4). This procedure is repeated until the whole population of trail vectors is obtained. The rest of the algorithm is the same as standard DE/rand/1. The overall procedure of the SRDE/rand/1 algorithm can be illustrated using the pseudo-code listed in Fig. 2.

$I_j = j \;, \forall j \in \{1,..,\lambda\}$
for $i = 1$ to N **do**
  for $j = 1$ to $\lambda - 1$ **do**
    sample $\mu \in U(0,1)$
    if $\varphi(I_j) = \varphi(I_{j+1}) = 0$ or $(\mu < P_f)$ **then**
      if $f(I_j) > f(I_{j+1})$ **then**
        swap(I_j, I_{j+1});
      end if
    else
      if $\varphi(I_j) > \varphi(I_{j+1})$ then
        swap(I_j, I_{j+1});
      end if
    end if
  end if
  end for
  if no *swap* done **break**
end for

Fig. 3. Pseudo-code of stochastic ranking [3]

3.2 SRDE/best/1

The mechanism of SR can be embedded not only to rand/1 mutation strategy, but also to other mutation strategies. The resultant variation of SRDE can also achieve performance improvement. In this section, we investigate the embedment of SR to best/1 mutation strategy, thus the SRDE/best/1.

The pseudo code of SRDE/best/1 is show in Fig. 2. In this variation of SRDE, the base individual is always selected as the uppermost individual of the population after SR, therefore the 'best' individual of the population in the SR sense.

4. EXPERIMENT

In this section, two sets of experiments were conducted to compare the performances of DE and SRDE with two different mutation strategies, in the well-known 24 benchmark test problems [15]. The codes were implemented in MATLAB and run on an Intel Core2 laptop with 4G RAM under WINDOWS-XP platform. For each test problem, 50 independent runs were conducted.

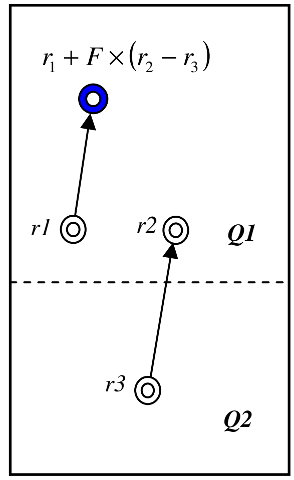

Fig. 4. Illustration of the modified mutation strategy in SRDE/rand/1. Note that the population ranked by SR is divided into upper part Q1 and lower part Q2. Difference of one randomly selected individual r2 from Q1 and one randomly selected individual r3 from Q2 form a differential vector pointing towards r2.

4.1Configuration and Parameters Setting

For all test problems, we set the same values for the parameters used both for standard DE and SRDE, and they are listed as following:

Population size: 100 [4][6]

Maximum number of generation: 10000

Differential factor: F = 0.7 [9]

Differential crossover probability: p_{CR} = 0.8

Value to reach, VTR = 10^{-4} [13]

The additional parameter for Stochastic Ranking: p_f = 0.45 [3]

The additional parameters for SRDE/rand/1 are: γ = 0.3

In this paper, the standard *binary* crossover operator was adopted for all algorithms.

4.2 Comparison of DE and SRDE with rand/1/bin mutation

The purpose of the conducted experiments in this subsection is to compare the performance of standard DE and SRDE with rand/1/bin strategy. In this experiment, median number of solution candidates (NSC) and successful rate (SuR) of 50 runs have been used as performance measures. For convenience of comparison, we further defined two metrics:

$$SuR_{diff} = SuR_{SRDE} - SuR_{DE}$$

$$NSC_{ratio} = {NSC_{DE}}\Big/{NSC_{SRDE}}$$

The second metric is the same as Acceleration Rate (AR) defined in [6], which provides a very convenient way to compare search speed. From the results listed in Table 1, it can be seen DE failed in problems g03,g13,g20,g22,g23, and had a very low successful rate in problems g17(20%) . For these problems, SRDE had significant improvements, i.e. for g13, SuR is improved from 0% to 44%,

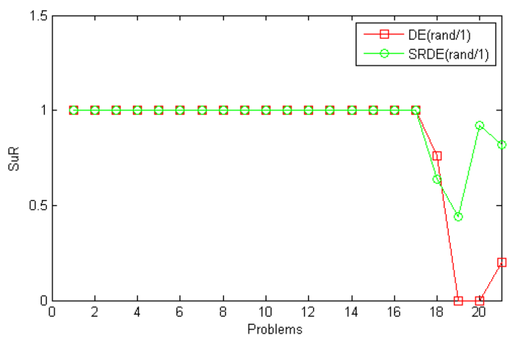

Fig. 5. SuR vs. Problems

.

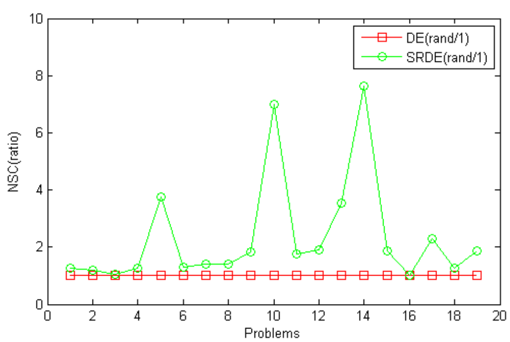

Fig. 6. NSC_{ratio} vs. Problems.

for g17, from 20% to 82%, for g23, from 0% to 92%. In all other test problems, the SuRs of DE and SRDE are comparable, except for g21 there is a slightly reduction from 76% to 64%. Fig. 5 and Fig. 6 show the SuRs and NSC_{ratio} of DE and SRDE in all benchmark functions respectively. The horizontal axis of Fig. 5 and Fig. 6 represents the new index of test problems according to the difficulty level of problems in an ascending order [19]. Table 1 also shows that SRDE outperforms DE in terms of search speed. SRDE had less NSCs than those of DE in all test functions in which they can both succeed to achieve the optimal. For those test functions that either SRDE or DE cannot find optimal, no information of NSCs will be used for comparison. Fig. 6 also demonstrates this in a graphical way, in which the vertical axis is the NSC_{ratio} As we explained before, NSC_{ratio} provides a very good metric to compare the search speed. From the average of NSC_{ratio} value of all test problems, it can be concluded that the average improvement of convergence speed was 134.4%. Fig. 9 show the comparison of convergence speed of SRDE/rand/1/bin and DE/rand/1/bin in test function 1, 2, 15, 18.

4.3 Comparison of DE and SRDE with best/1/bin mutation strategy

In this subsection, comparison study of DE and SRDE with respect to best/1/bin strategy has been carried out. The results are showed in Table 2. It can be seen from Table 2 that SRDE improved the successful rate in problem g13, g17, g21 considerably while in the rest problems the results are comparable. In terms of the convergence speed, it can be seen that SRDE could obtain the optimal with much less NSCs than DE in problems g05, g11, g13, g14, g15, g17, g23, although in some problems SRDE needed slightly more NSCs. It is noted that to avoid one value to dominate others in calculating the average improvement of convergence speed, we discount the NSC_{ratio} of g13, which is 14.96 and significantly bigger than others. The average improvement of convergence speed is therefore 82.15%.

Fig. 7 and Fig. 8 also show the improvement of success rate and convergence speed of SRDE over DE with best/1/bin strategy. It is notable that the horizontal axis of Fig. 7 and Fig. 8 also represents the new index of test problems according to the difficulty level of problems in an ascending order as in Fig. 5 and Fig. 6. Fig. 10 show the comparison of convergence speed of SRDE/best/1/bin and DE/best/1/bin in test function 5, 11, 13, 15.

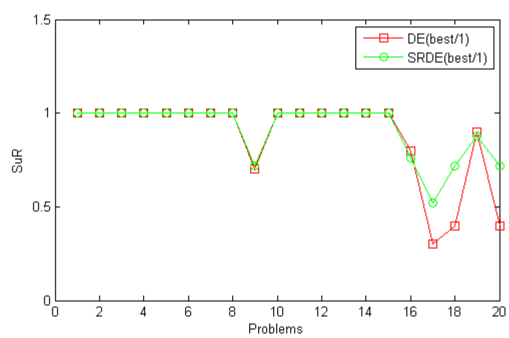

Fig. 7. SuR vs. Problems

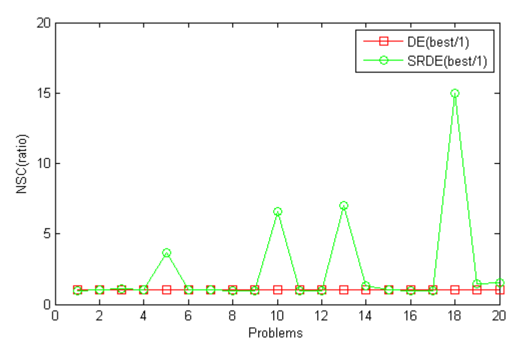

Fig. 8. NSC_{ratio} vs. Problems

5. CONCLUSION

In this paper, we propose a new scheme of DE, SRDE to improve the performance of the standard DE in constraint optimization applications. One critical mechanism embedded in SRDE is applying SR to rank the whole population of individuals with both objective value and constraint violation to be compared, before evolutionary operations are used. The ranked population is then able to provide useful information e.g. direction in the mutation operation to guide the search process. Li [5] also takes advantage of directional information within DE framework for multi-objective optimization. However, the way of extracting directional information is quite different from ours. The comprehensive experimental results show that SRDE outperforms, or at least is comparable with standard DE using both rand/1/bin and best/1/bin mutation strategies in all 24 tested benchmark functions, in terms of both convergence speed and success rate. The convergence speed, however, is improved much more significantly. For the rand/1/bin mutation strategy, the improvement ratio is 134.4%, for best/1/bin strategy, the ratio is 82.15%. It has also been shown in another application work [20] that a slight variation of SRDE can achieve much better optimization solution than standard DE and some other state-of-the-art EAs in a MEMS design optimization problem.

6. REFERENCES

[1] Storn, R. and Price, K. Differential evolution – A simple and efficient heuristic for the global optimization over continuous spaces. Journal of Global Optimization 11, 341 – 359.

[2] Silvio Priem Mendes, Juan A. Gomez Pulido, Miguel A. Vega Rodriguez, Maria D. Jaraiz Simon, Juan M. Sanchez Perez, "A Differential Evolution Based Algorithm to Optimize the Radio Network Design Problem," e-science,pp.119, Second IEEE International Conference on e-Science and Grid Computing (e-Science'06), 2006 .

[3] Runarsson T.P., Yao X., 2000. Stochastic Ranking for Constrained Evolutionary Computation. IEEE Trans. Evol. Comput. 4, 3 (Sept. 2000), 284-294

[4] X. Yao, Y. Liu, and G, Lin, Evolutionary programming made faster. IEEE trans. Evol, Compt. Vol 3, no 2,July 1999.

[5] Iorio A.W., Li X., 2006. Incorporating Directional Information within a Differential Evolution Algorithm for Multi-objective Optimization. GECCO’06, Seattle, Washington, USA.

[6] Rahnamayan S., Tizhoosh, H.R., Salama M.M.A. Opposition – based Differential Evolution. Evolutionary Computation, IEEE Transactions on, Vol 12, Issue 1, 2008: 64 – 79

[7] Noman, N.; Iba, H. Accelerating Differential Evolution Using an Adaptive Local Search. Evolutionary Computation, IEEE Trans. on Vol 12, Issue 1, :107 - 125

[8] Brest J., Zumer V., Maucec M. S., 2006. Self-Adaptive Differential Evolution Algorithm in Constrained Real-Parameter Optimization. IEEE Congress on Evolutionary Computation Sheraton Vancouver Wall Center Hotel, Vancouver, BC, Canada, 215-222

[9] Takahama T., Sakai S., 2006. Constrained Optimization by the ϵ Constrained Differential Evolution with Gradient-Based Mutation and Feasible Elites, IEEE Congress on Evolutionary Computation Sheraton Vancouver Wall Center Hotel, Vancouver, BC, Canada, 1-8

[10] Kukkonen S., Lampinen J. 2006. Constrained Real-Parameter Optimization with Generalized Differential Evolution. IEEE Congress on Evolutionary Computation Sheraton Vancouver Wall Center Hotel, Vancouver, BC, Canada, 207-214

[11] Liang J.J., Suganthan P.N., 2006. Dynamic Multi-Swarm Particle Swarm Optimizer with a Novel Constraint-Handling Mechanism, IEEE Congress on Evolutionary Computation Sheraton Vancouver Wall Center Hotel, Vancouver, BC, Canada, 9-16

[12] Mezura-Montes E., Velazquez-Reyes J., Coello C.A.C., 2006. Modified Differential Evolution for Constrained Optimization, IEEE Congress on Evolutionary Computation Sheraton Vancouver Wall Center Hotel, Vancouver, BC, Canada, 25-32

[13] A. K. Qin, V. L. Huang, and P. N. Suganthan, Differential evolution algorithm with strategy adaptation for global numerical optimization", IEEE Trans. on Evolutionary Computations, DOI: 10.1109/TEVC.2008.927706, 2009

[14] J. Sun, Q. Zhang, and E. P. K. Tsang. DE/EDA: A New Evolutionary Algorithms for Global Optimization. Information Sciences, 169:249-262, 2004.

[15] J. J. Liang, Thomas Philip Runarsson, Efren Mezura-Montes, Maurice Clerc, P. N. Suganthan, Carlos A. Coello Coello & K. Deb,"Problem Definitions and Evaluation Criteria for the CEC 2006 Special Session on Constrained Real-Parameter Optimization", Technical Report, Nanyang Technological University, Singapore, March 2006

[16] J. Lampinen, A constraint handling approach for the differential evolution algorithm, in Proceedings of the Congress on Evolutionary Computation 2002 (CEC’02), May, vol. 2, IEEE Service Center Piscataway, NJ 2002, pp. 872-877.

[17] R. L. Becerra, C. A. C. Coello, Cultured differential evolution for constrained optimization , Computer Methods in Applied Mechanics and Engineering, Vol. 195, No. 33-36, 2006, pp. 4303-4322.

[18] K. Deb, and H. Gupta, A constraint handling strategy for robust multi-criterion optimization, KanGAL Report No. 2005001, 2005

[19] Kukkonen, S.; Lampinen, J. Constrained Real-Parameter Optimization with Generalized Differential Evolution. Evolutionary Computation, 2006. CEC 2006. IEEE Congress on Volume , Issue , 0-0 0 Page(s):207 – 214

[20] Z. Fan, J. Liu, T, Sørensen, P, Wang. Robust Layout Synthesis of MEMS Components Based on an Improved Differential Evolution Algorithm. IEEE trans. On Industrial Electronics.(accepted)

TABLE 1

COMPARISON OF DE AND SRDE WITH *RAND/1/BIN* STRATEGY

DE(rand/1/bin)				SRDE(rand/1/bin)				SuR_{diff}	NSC_{ratio}
Prob.	**Median of NSC(x100)**	**Feasible Rate(FR)**	**Success Rate(SuR)**	**Prob.**	**Median of NSC(x100)**	**Feasible Rate(FR)**	**Success Rate(SuR)**		
g01	607	100%	100%	g01	334	100%	100%	0	1.8175
g02	7208	100%	100%	g02	2036	100%	100%	0	3.5403
g03	-	100%	-	g03	-	100%	-	-	-
g04	412	100%	100%	g04	295	100%	100%	0	1.3966
g05	3377	100%	100%	g05	442	100%	100%	0	7.6403
g06	223	100%	100%	g06	179	100%	100%	0	1.2458
g07	6473	100%	100%	g07	3409	100%	100%	0	1.8988
g08	30	100%	100%	g08	24	100%	100%	0	1.2500
g09	1076	100%	100%	g09	769	100%	100%	0	1.3992
g10	7514	100%	100%	g10	4303	100%	100%	0	1.7462
g11	529	100%	100%	g11	141	100%	100%	0	3.7518
g12	67	100%	100%	g12	63	100%	100%	0	1.0635
g13	-	100%	-	g13	5737	100%	44%	+44%	INF
g14	5828	100%	100%	g14	3114	100%	100%	0	1.8715
g15	1337	100%	100%	g15	191	100%	100%	0	7.0000
g16	283	100%	100%	g16	217	100%	100%	0	1.3041
g17	4821	100%	20%	g17	2574	100%	82%	+62%	1.8730
g18	5956	100%	100%	g18	2594	100%	100%	0	2.2961
g19	5415	100%	100%	g19	5318	100%	100%	0	1.0182
g20	-	-	-	g20	-	-	-	0	-
g21	1688	100%	76%	g21	1355	100%	64%	-12%	1.2458
g22	-	-	-	g22	-	-	-	0	-
g23	-	-	-	g23	8967	100%	92%	+92%	INF
g24	83	100%	100%	g24	70	100%	100%	0	1.1857
Ave									2.3443

TABLE 2

COMPARISON OF DE AND SRDE WITH *BEST/1/BIN* STRATEGY

DE(best/1/bin)				SRDE(best/1/bin)				SuR_{diff}	NSC_{ratio}
Prob.	**Median of NSC(x100)**	**Feasible Rate(FR)**	**Success Rate(SuR)**	**Prob.**	**Median of NSC(x100)**	**Feasible Rate(FR)**	**Success Rate(SuR)**		
g01	190	100%	70%	g01	199	100%	72%	+2%	0.9548
g02	-	100%	-	g02	-	100%	-	-	-
g03	-	100%	-	g03	-	100%	-	-	-
g04	213	100%	100%	g04	211	100%	100%	0	1.0095
g05	1362	100%	100%	g05	194	100%	100%	0	7.0206
g06	132	100%	100%	g06	124	100%	100%	0	1.0645
g07	1436	100%	100%	g07	1452	100%	100%	0	0.9890
g08	16	100%	100%	g08	17	100%	100%	0	0.9412
g09	369	100%	100%	g09	372	100%	100%	0	0.9919
g10	1976	100%	100%	g10	1988	100%	100%	0	0.9940
g11	121	100%	100%	g11	33	100%	100%	0	3.6667
g12	29	100%	100%	g12	27	100%	100%	0	1.0741
g13	3023	100%	40%	g13	202	100%	72%	+32%	14. 96
g14	1353	100%	100%	g14	1048	100%	100%	0	1.2910
g15	557	100%	100%	g15	85	100%	100%	0	6.5529
g16	134	100%	100%	g16	133	100%	100%	0	1.0075
g17	1897	100%	40%	g17	1222	100%	72%	+32%	1.5524
g18	845	100%	80%	g18	864	100%	76%	- 4%	0.9780
g19	2000	100%	100%	g19	1944	100%	100%	0	1.0288
g20	-	-	-	g20	-	-	-	-	-
g21	780	100%	30%	g21	796	100%	52%	+22%	0.9799
g22	-	-	-	g22	-	-	-	-	-
g23	4538	100%	90%	g23	3041	100%	88%	- 2%	1.4923
g24	54	100%	100%	g24	53	100%	100%	0	1.0189
Ave									1.8215

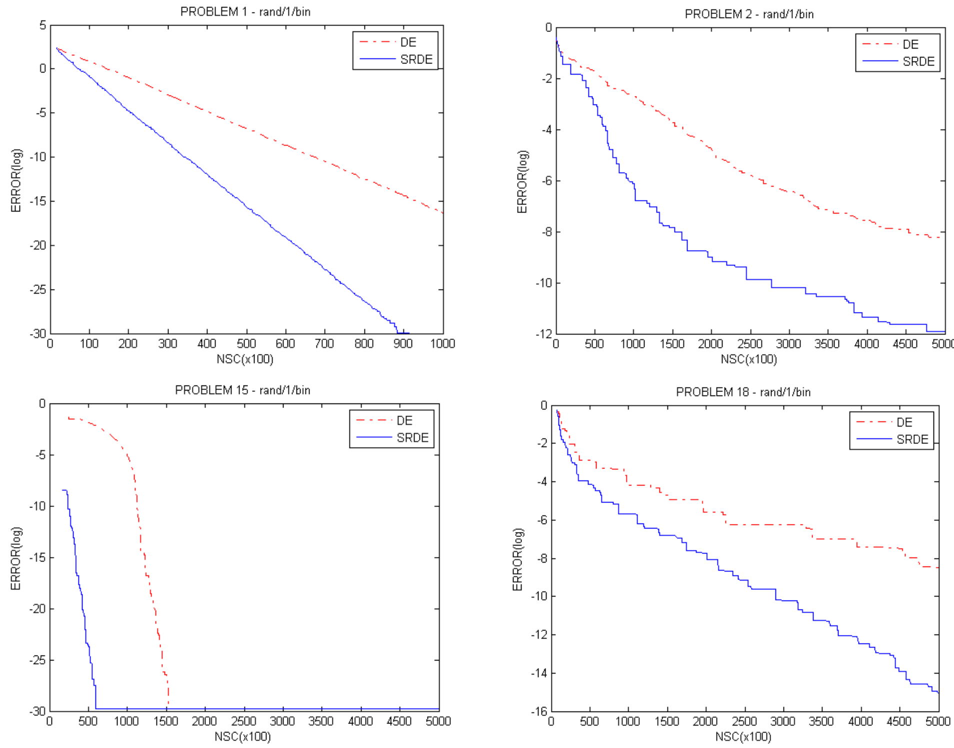

Fig. 9. Convergence graphs of DE and SRDE with rand/1 mutation strategy in problems 1, 2, 15, 18

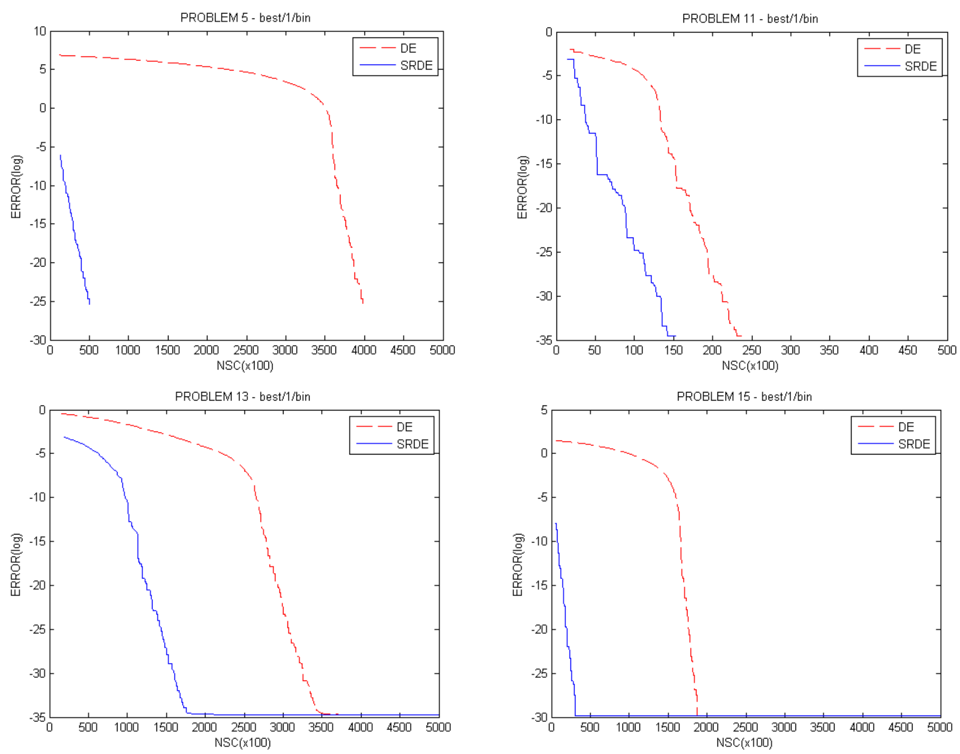

Fig. 10. Convergence graphs of DE and SRDE with best/1 mutation strategy in problems 5, 11, 13, 15