Repair Algorithms with Tolerance Relaxation for Chemical Engineering Optimization Problems

Nutchawat Ploywattanawong[1] and Soorathep Kheawhom[2] and Prabhas Chongstitvatana[1]

[1]Department of Computer Engineering, Chulalongkorn University
Nutchawat.P@Student.chula.ac.th, Prabhas@chula.ac.th

[2]Department of Chemical Engineering, Chulalongkorn University
Soorathep.K@chula.ac.th

Abstract

Global optimization is required when the objective function and constraints are nonlinear. One of global optimizers, Differential Evolution, has been applied to constrained optimization problems successfully. This paper introduces a new constraint handling scheme using adaptive relax tolerance and familiar selection scheme with differential evolution (T-DRDE). This method is an improvement of a previous method, a dominance-based selection scheme with a repair algorithm based on the gradient information derived from the equality constraint (DRDE). To compare T-DRDE with DRDE, several test problems and chemical engineering optimization problems are used. The results show that the performance of T-DRDE is competitive with DRDE. It can effectively handle constraints encountered in chemical engineering optimization problems.

Keywords*:* Constraint handling, Differential evolution, Constrained optimization

1. Introduction

There are two methods for solving optimization problems [1]: 1) Deterministic method uses explicit estimation and branch and bound search [2], 2) Stochastic methods contain many methods such as Simulated Annealing, Tabu search and Evolutionary Algorithm [3]. Deterministic method guarantees global optimum with respect to objective functions and constraints such as continuity and convexity. Stochastic methods obtain convergence to optima but do not guarantee global optimum [4]. Frequently in practice, stochastic methods obtain the value near global optimum. Stochastic methods must be able to manage constraints effectively.

Presently, Evolutionary Algorithm (EA) has been receiving a lot of attention because it can search for solutions of the complex problems effectively. There are large numbers of works in applying EA for optimization problems in the past ten years [5]. Differential Evolution (DE) is one of EA. It can handle both non-differentiable and nonlinear functions [6]. DE is proved to be stable and efficient for many optimization problems. However, it is difficult for DE to handle nonlinear constraints. Often, it gets stuck at local minima. To solve this problem it is important to have a method to handle constraints effectively.

Many algorithms have been proposed to solve optimization problems that focused on handling constraints [7, 8]. For EA, there are many techniques to handle constraints: 1) penalty functions [9], 2) special representations and operators, 3) repair algorithms, 4) separation of objective and constraints, and 5) hybrid methods.

One of the most popular methods is penalty functions. It is based on transforming constrained problems to unconstrained problems by penalising infeasible points. Another method is repair algorithm [10]. This method attempts to move infeasible points into feasible region. Repair algorithm uses heuristics to guide the repair process [11, 12]. Chootinan and Chen [13] proposed a heuristic that used gradient computed from a set of constraints to repair infeasible points. Repair algorithm is effective if infeasible solutions are easy to adjust.

There are many examples of the work that used separation of objective functions. Coello and Montes [14] proposed a non-dominance based selection which constraints are considered as objectives. Takahama and Sakai [15] proposed εDE. It is DE with ε mutation based on gradient. Kheawhom [16] proposed a combined method of dominance-based and repair based on gradient. Dominance-based method considers variable vectors that are better than constraints. Repair-base method focuses on equational constraints. Haibo Zhang [17] proposed C-IDE method. This method combined DE with Tabu search and handled constraints similar to εDE method. The parameter μ is computed from the sum of excess constraints to determine suitable search

direction. A heuristic function is used to improve μ in each iteration.

This work proposes a method to handle constraints effectively. The proposed method is based on improving the repair algorithm by extending Kheawhom's method and using the parameter μ from C-IDE to increase flexibility. Moreover, the DE framework has been extended to improve the selection of population. The proposed method reduces the computation resource while maintaining the quality of solutions.

2. Background Theory

2.1 Constrained optimization problem: is generally defined as:

$$\begin{aligned} &\min && F(x), \\ &\text{s.t.} && g_l(x) \le 0;\ l = 1, 2, \ldots, L, \\ &&& h_m(x) = 0;\ m = 1, 2, \ldots, M, \\ &&& x_n^{(L)} \le x_n \le x_n^{(U)};\ n = 1, 2, \ldots, N \end{aligned} \tag{1}$$

where x_n is an optimization variable varying in the range $[x_n^{(L)}, x_n^{(U)}]$, The function F(x) is the objective function with $g_l(x)$ inequality constraint and $h_m(x)$ equality constraint.

2.2 Total absolute violation (TAV) [17]**:** A total violation value of the variables in both equality and inequality constraints. An equality constraint violation is an absolute differentiate result from left and right side. An inequality constraint violation is similar but it has zero value when the differentiate result is negative. The TAV of constraint optimization problem is defined as

$$\mathrm{TAV}_k = \sum_{i=1}^{m1} |h_i(x)| + \sum_{j=1}^{m2} \max(0, g_j(x)) \quad k = 1, 2, \ldots, \mathrm{NP} \tag{2}$$

2.3 Differential Evolution [6]**:** is a typical of EAs which is very powerful and robust stochastic optimization algorithm. It is capable of handling non-differentiable, nonlinear and multi-modal objective functions. The DE process works as follows**:**

2.3.1 Parent vector: Assign values of the first parent vector randomly. The values are in the range of lower and upper boundary.

Iteration process: After that, DE iteration will be started. The iterative process consists of two main functions as follows:

2.3.2 Mutation [18]: Create the perturb vectors (V_i) for k individuals which originate the parent vectors. In this work, the amount of V_i is equal to a number of population (NP). Various mutation schemes have been proposed. We show two such schemes [19]:

a) DE/rand/1/bin scheme: Randomly select three vectors in the populations and generated V_i as follows

$$Vi = X_{r3} + F(X_{r2} - X_{r1}) \tag{3}$$

where X_{r1}, X_{r2} and X_{r3} are randomly selected vectors, and $r1 \neq r2 \neq r3 \neq i$ are satisfied, $F \in [0,1+]$ is a control parameter of the algorithm.

b) Trigonometric mutation scheme: Randomly select vector as DE/rand/1/bin scheme. The perturb vector V_i is then generated by the center point, a sum of three weighted vector differentials, as follows

$$\begin{aligned} V_i &= \frac{X_{r1} + X_{r2} + X_{r3}}{3} + (p_2 - p_1)(X_{r1} - X_{r2}) \\ &\quad + (p_3 - p_2)(X_{r2} - X_{r3}) + (p_1 - p_3)(X_{r3} - X_{r1}), \\ p_1 &= \frac{|F(X_{r1})|}{|F(X_{r1})| + |F(X_{r2})| + |F(X_{r3})|}, \\ p_2 &= \frac{|F(X_{r2})|}{|F(X_{r1})| + |F(X_{r2})| + |F(X_{r3})|}, \\ p_3 &= \frac{|F(X_{r3})|}{|F(X_{r1})| + |F(X_{r2})| + |F(X_{r3})|} \end{aligned} \tag{4}$$

where X_{r1}, X_{r2} and X_{r3} are randomly selected vectors, and $r1 \neq r2 \neq r3 \neq i$ and $F(X_{r1})$, $F(X_{r2})$ and $F(X_{r3})$ are the objective functions evaluated at X_{r1}, X_{r2} and X_{r3} respectively.

2.3.3 Crossover operator: Alternate all variables in perturb vector $V_i(v_{i,1}, v_{i,2}, \ldots, v_{i,n})$ and its parent vector $X_i(x_{i,1}, x_{i,2}, \ldots, x_{i,n})$ that generates the trial vector $U_i(u_{i,1}, u_{i,2}, \ldots, u_{i,n})$ as follows

$$u_{i,j} = \begin{cases} v_{i,j} & \text{if random}[0, 1) \le \mathrm{CR} \\ x_{i,j} & \text{otherwise.} \end{cases} \tag{5}$$

where $CR \in [0.1]$ is a crossover factor.

In this part of work, the repair algorithm has been introduced to improve the trial vector U_i. At each generation i, the trial vector U_i is compared with its parent vector X_i for the better fitness function then uses that to create the new parent vector X_{i+1}. In this work, the dominance-based selection scheme is used to improve effective selection.

The evolutionary process is repeated until the stopping criteria are satisfied. That is, the differentiate minimum objective function of generation i^{th} and $i+1^{th}$ are less than SC parameter continuously for T times.

2.4 Constraint handling method: It is a heuristic which adjusts variables according to constraints to get the best objective function. The constraint handling methods can be adapted from many stochastic algorithms. DE is used in this work.

2.4.1 Gradient-based repair algorithm [13]**:** It is a repairing algorithm for adjusting variables to reduce its violation from infeasible to feasible, when the violation less than a tolerance (Ɛ). Generally, the dimension number of the equality constraints can be reduced without distorting the results.

In case of N-dimensional optimization problem with M equality constraints, the degree of freedom for this case is N – M. Any infeasible vector X containing N variables can be repaired by solving the system of M equations. In this work, a gradient-based repair algorithm is applied to solve the equality constraint violation by Newton method.

Newton method [20]**:** It is the method for solving a root of function approximated by tangent line. In this work, Newton method is improved by adjusting violation of equality constraint, h(X), via approximate vector X. Assume the differential is available at h(X), then h'(X) or Jacobian, J(X), also can be found at $h(X_i)$. The Jacobian is a slope of function y = h(X) at (X_i, $h(X_i)$), then equation is represented as $h(X_{i+1}) - h(X_i) = J(X_i)(X_{i+1} - X_i)$. Reduce the lowest violation, $h(X_{i+1})$ by assigning it to zero. Therefore, the vector X of the next generation can be estimated as follows

$$X_{i+1} = X_i - J^{-1}(X_i)h(X_i) \quad (6)$$

where $J(X_i)$ is Jacobian matrix, and $H(X_i)$ is the vector of equality constraints violation. Iteration stops if either the sum of the degree of constraints violation is less than a tolerance specified, Ɛ, or the maximum iteration number has been reached.

In each repair iteration, Newton method estimate moves to zero of equality constraints violation, $h(X_{i+1})$ of vector X_{i+1} at k^{th} individuals of all population, NP; k = 1, 2,…, NP. The result of $h(X_{i+1})$ is used to obtain the accuracy of global optimum in range $10^{-3} - 10^{-4}$ [16]. Therefore, we assign 10^{-4} as a minimum tolerance value (Ɛ) to be a stop criterion in estimation the vector X in the repair process. Another stopping criteria is the repair process vector X at individual k cannot be found within 100 times. Then the next individual vector X will be used. The repair process is ended when all individuals are repaired.

Fig. 1a presents the distribution of population before repair process is started. Fig. 1b shows the distribution of population after repairing [16].

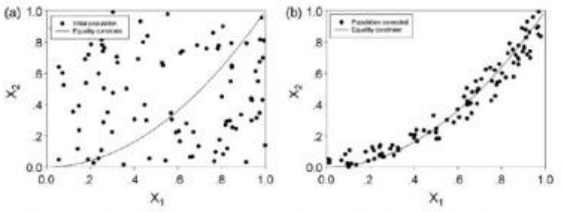

Fig. 1. (a) the distribution of population before repair process and (b) the distribution of population repaired by Kheawhom repair algorithm.

2.4.2 Dominance-based selection scheme [14]**:** is a scheme to compare and select the vector that is better. In this work, the candidate vector pair are the parent vector, X_i and the trial vector, U_i for generating the next generation parent vector, X_{i+1}. We consider the total absolute violation, TAV and objective function. Three possible cases are expected. In the first case, both U_i and X_i are feasible, TAV is less than or equal to Tolerance (Ɛ). The vector with a better objective value is picked to X_{i+1}. In the second case, one is feasible, but the other one is infeasible, it is infeasible when TAV is more than Ɛ. The feasible vector is picked to X_{i+1}. In the last case, both vectors are infeasible. The vector with lower TAV is picked to X_{i+1}. The scheme for the selection is defined as follows:

$$X_{i+1} = \begin{cases} X_i & \text{if } X_i \prec U_i \\ U_i & \text{otherwise} \end{cases} \quad (7)$$

where $X_i \prec U_i$ denotes that X_i dominates U_i. That is X_i has better objective value than U_i and/or lower degree of TAV.

2.4.3 Relaxation of constraint (μ) [17]**:** The method adapts the violation of equality and inequality constraint to decide boundary relaxation for constraint violation of vector X. The vector X indicates feasibility. When the violation is less than or equal to μ, it is feasible and it is infeasible when the violation is more than μ. When applying this method with differential evolution (DE), μ is reduced in each generation. The adjusted value of μ is calculated from the median total absolute violation of all individuals in the first generation. The other generation, μ is defined as follows:

$$\mu_{i+1} = \mu_i\left(1 - \frac{F_F}{NP}\right) \quad (8)$$

where F_F is the fraction of feasible individuals with respect to the relaxed constraints, in the latest population at generation, i. The number of population is NP. The main idea is to accelerate or decelerate the adjustment of μ dependent to F_F. The greater value of F_F will affect μ_{i+1}.

2.4.4 Handling boundary constraints: It is important that the optimize variables must lie inside their allowed ranges. We replace each variable that violates boundary constraints by the upper or lower limits, according to the following rule:

$$x_{i,j} = \begin{cases} x_{i,j}^{(L)} & \text{if } x_{i,j} < x_{i,j}^{(L)} \\ x_{i,j} & \text{if } x_{i,j}^{(L)} \le x_{i,j} \le x_{i,j}^{(U)} \\ x_{i,j}^{(U)} & \text{if } x_{i,j} > x_{i,j}^{(U)} \end{cases} \quad (9)$$

where $x_{i,j}^{(L)}$ and $x_{I,j}^{(U)}$ are the upper and lower bounds of each variable, respectively. This is the method applied in this work. However, various boundary constraint handling methods can be found in the literature [21].

2.4.5 Handling integer and discrete variables: The original DE is incapable of handling discrete variables. However, it is very easy to modify the algorithm to deal with integers and/or discrete variables. First, continuous variables are converted to integer variables by truncation. Then, the truncated variables are used to evaluate the objective function.

Discrete variables can also be easily handled. Instead of directly using discrete variables as the optimized variables, the indexes of all discrete variables are assigned first. The index of each discrete variable is then used as the optimization variables. The original discrete variables are still used to evaluate the objective function.

3. Propose method

3.1 Relaxation of Tolerance: This research proposes the method of relaxation of constraint applied to Tolerance (Ɛ) within the repair algorithm. This method selects vector X with values less than the value Ɛ. This is more flexible than previous research of Kheawhom. By assigning Ɛ constant in the range 10^{-3} – 10^{-4} the relaxation of tolerance is intended to reduce the number of function evaluation (NFE). The value is determined from the assessment of the equality constraint violation during the iteration of repair process of population X.

The Ɛ in the first round will be given as a median of total absolute violation of all individuals. The other round, it is based on the equations 10. Another alternative is to assign a value to Ɛ in regarded to Familiar selection scheme:

$$\varepsilon_{i+1} = \varepsilon_i\left(1 - \frac{\log(|F(X_i) - F(X_{i-1})| + 1)}{\log(F_{max} - F_{min} + 1)}\right) \quad (10)$$

where $F(X_i)$, $F(X_{i-1})$ are the objective functions of the current and previous generation. F_{max} and F_{min} are the objective function of all past generations.

The key concept of this relation is either increase or decrease the reduction speed of the value of Ɛ by the trend of convergence. If $F(X_i)$ and $F(X_{i-1})$ are significantly different, F does not converge well enough, therefore, the Ɛ should be significantly reduce in the next generation. This will initiate convergence in early stage and then slow down to the normalized objective function value.

3.2 Familiar selection scheme: It considers a condition in a model to adjust Tolerance (Ɛ). It is used as a criterion of the equality constraint violation. The Familiar selection scheme is divided into two cases as follows. Case 1, the difference of $F(X_i)$ and $F(X_{i-1})$ value is greater than the parameter SC, the stopping criterion. The value of Ɛ will be assigned by the equation 10. Case 2, the difference of $F(X_i)$ and $F(X_{i-1})$ value does not exceed SC and/or X_i with X_{i-1} are the same value. The Ɛ is assigned the value of the minimum total absolute violation of all individuals.

The reason for the value adjustment of Tolerance (Ɛ) in the second case is that we cannot know whether the difference of $F(X_i)$ and $F(X_{i-1})$ are the same vectors as X_i and X_{i-1} or not. However, if both are the same vector then this could mean that the process of DE in the generation i^{th} cannot create a trial vector U_i with the objective function $F(U_i)$ which is better than $F(X_{i-1})$. Thus $F(X_i)$ and $F(X_{i-1})$ are the same. That is a sign that the vector X_i may be at local optimum under the equality constraint with an inappropriate Ɛ value. Therefore, the Ɛ value is set to the minimum total absolute violation of all individuals. However, the lower bound of Ɛ is located at 10^{-4}, it means that if Ɛ is assigned a value below the lower bound, the value will be rounded up to the lower bound.

4. Test problems

In this section, the performance and effectiveness of a combination of a dominance-based selection scheme and gradient-based repair algorithm (DRDE) is compared with the proposed constraint handling scheme, T-DRDE. Nine different constrained optimization problems that have been previously studied in the literature are used. They consist of the optimization problems and nonlinear programming (NLP) [22], P01 and P02; mixed integer nonlinear programming (MINLP) [23], P03; non-convex MINLP [24], P04. The chemical engineering optimization problem are: heat exchanger network synthesis [24], P05; MINLP small planning [25], P06; reactor network design [26], P07; heat exchanger network design [27], P08; and separation network synthesis [26], P09. Each problem was run 20 times to check the consistency of the results. The

parameters and mathematical characteristics of the problems are shown in Table 1 and 2 respectively.

Table 1: The parameters of the problems

Parameter	amount
Number of population, NP	1000
Maximum generation	1000
Maximum repairing	100
Minimum tolerance, Ɛ	0.0001
Factor of mutation, F	0.85
Crossover rate, CR	0.8
Trigonometric mutation probability	0.05
Different objective function, SC	0.0000001
Time to stop criteria, T	10
*P08 only set time to stop criteria to 40	

Table 2: Mathematical characteristics of the problems where LI, NI, LE, and NE are linear inequality, nonlinear inequality, linear equality, and nonlinear equality constraint respectively.

Problem	Variables	LI	NI	LE	NE	Global optimum, f*
P01	5	0	0	0	3	0.0539498
P02	4	2	0	0	3	5126.4981
P03	9	4	0	3	2	99.23963
P04	5	3	0	0	2	7.66718
P05	12	0	0	11	0	36162.9886
P06	11	5	0	3	2	-1.9230986
P07	6	0	1	0	4	-0.388812
P08	8	0	3	3	0	7049.25
P09	22	0	0	7	9	1.864

5. Results and analysis

The number of function evaluation, the number of objectives and equality constraints of DRDE are compared with T-DRDE at the population size 1000 in Table 3.

From Table 3 the number of function evaluation (NFE) of DRDE is higher than T-DRDE in 1st problem group (P03, P04, P06, P08, and P09). For 2nd problem group (P01, P02, P05, and P07) both methods are similar. Observing the mathematical characteristic of the problem in Table 2, it shows that the 2nd group has only equality constraint. Except the problem P07 which has one inequality constraint. DRDE is worse than T-DRDE in this problem (see also that the NFE ratio is 1.3). Consider the 1st group, except the P09, it appears that this group always has the inequality more than or equal to equality constraint; the result has a high NFE ratio value about 2 – 10. Consider to the inequality constraint, it appears that increasing of NFE ratio is dependent on the number of inequality constraint. It can be assumed that T-DRDE can reduce NFE if the problem has a large number of inequality constraints. However, P09 is different. It has equality constraint only but the NFE ratio is high about 4.65. P09 has 22 variables and they are almost independent. That fact affects the efficiency of repairing process of Newton method. Finding the inverse Jacobian with low dependency variable has a high chance to be Jacobian matrix with a singular. That means the algorithm which concentrates on repairing process (such as DRDE) will be worse off. All results can be summarized as "repairing process has a low performance for a problem with high inequality constraint and low variable dependency equality constraint." Not only the relaxation of tolerance reduces the inflexibility of repairing process but also increases cooperation with differential evolution in the mutation and crossover operations.

6. Conclusions

A new approach to relax the tolerance in repair algorithm was introduced in this paper. The proposed technique adopted a relax formulation and familiar selection scheme to adjust a tolerance in a repair algorithm based on the gradient information.

The developed scheme performed well with respect to the number of function evaluations required in inequality constraint and low dependency variable of equality constraint. To reduce the flexibility of the tolerance, it can be cooperated with differential evolutionary on mutation and crossover operators which it increased performance by reducing the computational cost. It is considerably lower than the cost required by a combination of a dominance-based selection scheme and a repair algorithm based on the gradient information.

7. References

[1] P.M. Pardalos, H.E. Romeijn, and H. Tuy, "Recent developments and trends in global optimization," Journal of Computational and Applied Mathematics, 124, 1-2, 1 December 2000, 209–228.

[2] C.A. Floudas, A. Aggarwal, and A.R. Ciric, "Global optimum search for nonconvex NLP and MINLP problems," Computers and Chemical Engineering, 13, 10, 10 October 1989, 1117–1132.

[3] H. Das, P.T. Cummings, and M.D. Levan, "Scheduling of serial multiproduct batch processes via simulated annealing," Computers and Chemical Engineering, 14, 12, 7 June 1990, 1351–1362.

[4] B. Lin, and D.C. Miller, "Tabu search algorithm for chemical process optimization," Computers and Chemical Engineering, 28, 11, 15 October 2004, 2287–2306.

[5] T. Back, D. Fogel, Z. Michalewicz, "Handbook of Evolutionary Computation," Institute of Physics Publishing and Oxford University Press, New York, 1997.

[6] R. Storn, and K. Price, "Differential Evolution - a Simple and Efficient Heuristic for Global Optimization

over Continuous Spaces," Journal of Global Optimization, Kluwer Academic Publishers, 11, 1997, 341 - 359.
[7] C. Coello, "Theoretical and numerical constraint handling techniques used with evolutionary algorithms: a survey of the state of the art," Computer Method in Applied Mechanics and Engineering, 191, 11–12, 4 January 2002, 1245-1287.
[8] Z. Michalewicz, and M. Shoenauer, "Evolutionary Algorithm for Constrained Parameter Optimization Problems," Evolutionary Computation, 4, 1, 1996, 1-32.
[9] A.V. Fiacco, and G.P. McCormick, "Extensions of SUMT for Nonlinear Programming: Equality Constraints and Extrapolation," Management Science, 12, 1968, 816-828.
[10] G. Liepins, and M. Vose, "Representational Issues in Genetic Optimization," Journal of Experimental and Theoretical Artificial Intelligence, 2, 1990, 4-30.
[11] R.L. Riche, and R. Haftka, "Improved Genetic Algorithm for Minimum Thickness Composite Laminate Design," Composites Engineering, 3, 1994, 121-139.
[12] D.M. Tate, and A.E. Smith, "A genetic approach to the quadratic assignment problem," Computers and Operations Research, 22, September 1995, 73-85.
[13] P. Chootinan, and A. Chen, "Constraint handling in genetic algorithms using a gradient-based repair method," Computers and Operations Research, 33, 2006, 2263-2281.
[14] C. Coello, and E. Montes, "Constraint-handling in genetic algorithms through the use of dominance-based tournament selection" Advanced Engineering Informatics, 16, 3, July 2002, 193-203.
[15] T. Takahama, and S. Sakai, "Constrained optimization by the ε constrained differential evolution with gradient-based mutation and feasible elites," In Proceedings of the IEEE Congress on Evolutionary Computation, Vancouver, BC, Canada.
[16] K. Soorathep, "Efficient constraint handling scheme for differential evolutionary algorithm in solving chemical engineering optimization problem," Journal of Industrial and Engineering Chemistry, 16, 4, 25 July 2010, 620-628.
[17] H. Zhang, and G.P. Rangaiah, "An efficient constraint handling method with integrated differential evolution for numerical and engineering optimization," Computer & Chemical Engineering, 37, 10 February 2012, 74-88.
[18] K.V. Price, "An Introduction to differential evolution," in New Ideas in Optimization, McGraw-Hill, London, UK, 1999, 79-108.
[19] H.Y. Fan, and J. Lampinen, "A trigonometric mutation operation to differential evolution," Journal of Global Optimization, 27, 1, 2003, 105-129.
[20] C.T. Kelley, "Solving Nonlinear Equations with Newton's Method," Fundamentals of Algorithms for Numberical Calculations, SIAM, Philadelphia, 2003
[21] T. Ba¨ ck, D. Fogel, Z. Michalewicz, Evolutionary Computation 2: Advanced Algorithms.
[22] K. Deb, "An efficient constraint handling method for genetic algorithms," Computer Method in Applied Mechanics and Engineering, 186 , 2000, 311-88.
[23] U.M. Diwekar, I.E. Grossmann, and E.S. Rubin, "An MINLP Process Synthesizer for a Sequential Modular Simulator," Journal of Industrial and Engineering Chemistry Research, 31, 1992, 313.
[24] V. Summanwar, V. Jayaraman, B. Kulkarni, H. Kusumakar, K. Gupta, and J. Rajesh, "Solution of constrained optimization problems by multiobjective genetic algorithm," Computers and Chemical Engineering, 26, 2002, 1481-1492.
[25] G.R. Kocis, and I.E. Grossmann, "Relaxation Strategy for the Structural Optimization of Process Flowsheets," Industrial and Engineering Chemistry Research, 26, 1987, 1869.
[26] C.S. Adjiman, I.P. Androulakis, and C.A. Floudas, "A Global Optimization Method, alphaBB, for General Twice-Differentiable Constrained NLPs: II – Implementation and Computational Results," Computers and Chemical Engineering, 22, 1998, 1159-1179.
[27] C. Floudas, and P. Pardalos, "A Collection of Test Problems for Constrained Global Optimization Algorithms," Springer, Berlin, Germany, 1990.

Table 3: The number function evaluate of DRDE compared with T-DRDE at 1000 population size

Problem	No. Generation	Result	DRDE			No. Generation	Result	T-DRDE			NFE ratio
			Amout of evaluate					Amout of evaluate			
			Equality Constraint	Objective function	Summary NFE			Equality Constraint	Objective function	Summary NFE	DRDE/ T-DRDE
P01	27	0.053957	240,698	29,150	**269,848**	30	0.053953	234,654	31,450	**266,104**	**1.01**
P02	49	5126.497653	126,600	49,059	**175,659**	46	5126.497652	122,178	44,980	**167,158**	**1.05**
P03	44	99.239016	576,884	28,132	**605,015**	39	99.239031	181,081	22,945	**204,027**	**2.97**
P04	26	7.667005	175,346	27,617	**202,963**	28	7.667005	77,923	27,908	**105,831**	**1.92**
P05	10	36162.978241	36,134	12,000	**48,134**	11	36162.978241	36,952	12,000	**48,951**	**0.98**
P06	25	-1.923439	1,293,324	13,064	**1,306,388**	26	-1.923423	122,012	12,265	**134,277**	**9.73**
P07	32	-0.388753	129,745	22,449	**152,194**	28	-0.388698	95,705	18,783	**114,488**	**1.33**
P08	220	7092.492903	1,536,858	72,307	**1,609,165**	319	7059.817520	596,832	96,162	**692,994**	**2.32**
P09	10	1.864159	1,084,567	1,322	**1,085,889**	13	1.864159	232,006	1,432	**233,437**	**4.65**