

A Hybrid Ant Colony Differential Evolution and its Application to Water Resources Problems

Musrrat Ali and Millie Pant

Department of Paper Technology
Indian Institute of Technology Roorkee, Saharanpur
campus, Saharanpur, India.

E-mail: {musrrat.iitr, millidma}@gmail.com

Ajith Abraham

Machine Intelligence Research Labs (MIR Labs),
Scientific Network for Innovation and Research
Excellence, P.O. Box 2259, Auburn,
Washington-98071-2259, USA.
E-mail: ajith.abraham@ieee.org

Abstract—Differential Evolution (DE) is generally considered as a reliable, accurate and robust optimization technique. However, the algorithm suffers from slow convergence rate and takes large computational time for optimizing the computationally expensive objective functions. Therefore, an attempt to speed up DE is considered necessary. This research introduces a modified differential evolution, called Ant Colony Differential Evolution, ACDE. The ACDE algorithm initializes the population using opposition based learning, in mutation phase it applies random localization technique and it simulates the movement of ants to refine the best solution found in each generation. Also, it maintains a single set of population while updating the population for next generation. ACDE validated on a test bed of 7 benchmark problems and two real life problems and the numerical results are compared with original DE. It is found that ACDE requires less computational effort to locate global optimal solution without compromising with the quality of solution.

Keywords: Differential Evolution, Ant colony optimization, opposition based learning, hybridization.

I. INTRODUCTION

DIFFERENTIAL evolution was proposed by Storn and Price [1] in 1995. It soon became a popular tool for solving global optimization problems because of several attractive features like having fewer control parameters, ease in programming, efficiency etc. DE is similar to GAs in the sense that it uses same evolutionary operators like mutation, crossover and selection for guiding the population towards the optimum solution. Nevertheless, it's the application of these operators that makes DE different from GA. The main difference between GAs and DE is that; in GAs, mutation is the result of small perturbations to the genes of an individual while in DE mutation is the result of arithmetic combinations of individuals. DE has been successfully applied to solve a wide range of real life application problems [2 - 4] and has reportedly outperformed other optimization techniques [5]. Despite several positive features, it has been observed that DE suffers from slow convergence rate and takes large computational time for optimizing the computationally expensive objective functions. Besides this, DE also suffers from the problem of premature convergence. Like other EA, the performance of DE deteriorates with the increase in

dimensionality of the objective function. Several modifications have been made in the structure of DE to improve its performance [6, 7]. In the present study we propose four modifications in the basic scheme of DE. The first modification is initialization of population by opposition based learning, second is in mutation where mutant vector is chosen tournament best not random, third is in update process, if a better individual found update the population of current generation in contrast to DE where population is updated in next generation and four is generate a point in the neighborhood of global best individual using ant colony algorithm.

The remainder of the paper is structured as follows. Section II describes the basics Differential Evolution. Section III presents the proposed DE version. Benchmark problems and real life problems are given in Section IV. Experimental setting and numerical results are given in Section V. Finally the paper is concluded in section VI.

II. DIFFERENTIAL EVOLUTION

Throughout the present study we shall follow *DE/rand/1/bin* version of DE and shall refer to it as basic version. This particular scheme is briefly described as:

DE starts with a population of NP candidate solutions: $X_{i,G}$, $i = 1, \dots, NP$, where the index i denotes the population and G denotes the generation to which the population belongs. The three main operators of DE are mutation, crossover and selection.

Mutation: The mutation operation of DE applies the vector differentials between the existing population members for determining both the degree and direction of perturbation applied to the individual subject of the mutation operation. The mutation process at each generation begins by randomly selecting three individuals $\{X_{r_1}, X_{r_2}, X_{r_3}\}$ in the population set of (say) NP elements. The i^{th} perturbed individual, $V_{i,G+1}$, is generated based on the three chosen individuals as follows:

$$V_{i,G+1} = X_{r_3,G} + F * (X_{r_1,G} - X_{r_2,G}) \quad (1)$$

Where, $i = 1 \dots NP$, $r_1, r_2, r_3 \in \{1 \dots NP\}$ are randomly selected such that $r_1 \neq r_2 \neq r_3 \neq i$,

F is the control parameter such that $F \in [0, 1]$.

Crossover: once the mutant vector is generated, the perturbed individual, $V_{i,G+1} = (v_{1,i,G+1}, \dots, v_{n,i,G+1})$, and the current population member, $X_{i,G} = (x_{1,i,G}, \dots, x_{n,i,G})$, are then subject to the crossover operation, that finally generates the population of candidates, or “trial” vectors, $U_{i,G+1} = (u_{1,i,G+1}, \dots, u_{n,i,G+1})$, as follows:

$$u_{j,i,G+1} = \begin{cases} v_{j,i,G+1} & \text{if } rand_j \leq C_r \vee j = k \\ x_{j,i,G} & \text{otherwise} \end{cases} \quad (2)$$

Where, $j = 1 \dots n$, $k \in \{1, \dots, n\}$ is a random parameter’s index, chosen once for each i . The crossover rate, $C_r \in [0, 1]$, is set by the user.

Selection: The selection scheme of DE also differs from that of other EAs. The population for the next generation is selected from the individual in current population and its corresponding trial vector according to the following rule:

$$X_{i,G+1} = \begin{cases} U_{i,G+1} & \text{if } f(U_{i,G+1}) \leq f(X_{i,G}) \\ X_{i,G} & \text{otherwise} \end{cases} \quad (3)$$

Thus, each individual of the temporary (trial) population is compared with its counterpart in the current population. The one with the lower objective function value will survive from the tournament selection to the population of the next generation. As a result, all the individuals of the next generation are as good as or better than their counterparts in the current generation. In DE trial vector is not compared against all the individuals in the current generation, but only against one individual, its counterpart, in the current generation.

III. PROPOSED DE ALGORITHM

Differential evolution algorithm described above typically converges relatively rapidly in the initial stage of the search and then slow down as it approach to global minimum. So keeping it in mind, in this section we proposed modified version, ACDE, which uses the concepts of opposition based learning to initialize the population, random localization for selecting the base vector. Besides this the concept of the movement of ants (following Gaussian distribution) [8] is applied as the algorithm approaches the global optimum. Further, one population set is used in contrast to the double population set as in basic DE. The basic structure of ACDE is same as DE and a step by step procedure of the algorithm is given here:

Step 1: Initialization: Generate randomly a set P of NP individuals, generate another set OP of size NP using opposition based method and take NP fittest individuals from the union of these two sets as initial population S. Input scaling factor F, crossover rate C_r and $\sigma=1$.

Step 2: Mutation: Select randomly three distinct individuals X_{r1} , X_{r2} and X_{r3} from population S and perform mutation using formula:

$$V_i = X_{tb} + F \times (X_{r2} - X_{r3})$$

Where individual X_{tb} is best of these three individuals and X_{r2} , X_{r3} are the remaining two.

Step 3: Crossover: Recombine the each target vector X_i with perturbed individual generated in step 4 to generate a trial vector U_i using equation (2).

Step 4: Selection: Calculate the objective function value at new generated individual. If it is better than target individual then replace target individual by this new individual in current population.

Step 5: Refining the Global Best Solution Using Ant Movement: After each iteration generate an individual in the neighborhood of global best individual X_{best} using ant colony algorithm by the expression $X_{new} = N(X_{best}, \sigma)$. If this individual is better than the worst individual in the population then replace the worst by this new individual. Update σ after each iteration as $\sigma = \sigma * 0.30$ if $\sigma < 10^{-3}$ then $\sigma = 10^{-3}$.

Step 6: Check whether the termination criterion met if yes then stop otherwise go to step 2.

These modifications enable the algorithm to get a better tradeoff between the convergence rate and robustness. Thus it is possible to increase the convergence rate of the differential evolution algorithm and thereby obtain an acceptable solution with a lower number of objective function evaluations. Such an improvement can be advantageous in many real-world problems where the evaluation of a candidate solution is a computationally expensive operation and consequently finding the global optimum or a good suboptimal solution with the original differential evolution algorithm is too time-consuming or even impossible within the time available.

IV. BENCHMARK AND REAL LIFE PROBLEMS

A. Benchmark problems

In the present study we have considered seven benchmark problems having box constraints. Out of these the first three problems are of fixed dimensions 2, 3 and 4, while the remaining problems are scalable in nature. Mathematical model of the problems along with their exact global optimum value are given the Appendix.

B. Real life problems

We have considered two real life problems that are quite common in the field of water resource management. These problems are described in the following subsections.

B1. Water pumping system formulation

A water pumping system [9] consists of two parallel pumps drawing water from a lower reservoir and delivering it to another that is 40 m higher, as shown in fig 1. In addition to overcoming the pressure difference due to the elevation, the friction in the pipe is $7.2w^2$ kPa, where w is the combined flow rate in kilograms per second. The pressure-flow-rate characteristics of the pumps are:

$$\text{Pump 1: } \Delta P \text{ (kPa)} = 810 - 25w_1 - 3.75w_1^2 \quad (4)$$

Pump 2: $\Delta P \text{ (kPa)} = 900 - 65w_2 - 30w_2^2$ (5)

Where w_1 and w_2 are the flow rates through pump 1 and pump 2, respectively. The system can be represented by four simultaneous equations. The pressure difference due to elevation and friction is:

$$\Delta P = 7.2w^2 + \frac{(40m)(1000kg/m)(9.807m/s)}{(1000 Pa/kPa)}$$

(6)

Equation (4), (5) and mass balance $w = w_1 + w_2$ (7)

The objective here is to minimize ΔP subject to the constraints (4), (5), (6), and (7).

Liebman et al. [10] modified the above problem as given below:

$$\min f = x_3$$

Subject to;

$$\begin{aligned} x_3 &= 250 + 30x_1 - 6x_1^2, & x_3 &= 300 + 20x_2 - 12x_2^2, \\ x_3 &= 150 + .5(x_1 + x_2)^2, \\ 0 &\leq X \leq (9.422, 5.903, 276.42) \end{aligned}$$

In general, the equality constraints are difficult to deal with. So there is a need to transform equality constraints into inequality constraints by some means or the other. Typically, they are handled by either of the following two methods, viz., (1) eliminating the parameter and hence reducing the dimensions of the problem (2) an equality constraint is formulated into two inequalities by introducing deviation variables on problem parameter. In the present study, one variable is eliminated while the other two equalities are transformed into inequalities using method 1. Hence, the reformulated problem is as follows:

$$\min f = x_3 = 150 + .5(x_1 + x_2)^2$$

Subject to;

$$\begin{aligned} g_1 &= 6x_1^2 - 30x_1 - 249.9999999 + 150 + .5(x_1 + x_2)^2 \\ &\geq 0 \\ g_2 &= 12x_2^2 - 20x_2 - 299.9999999 + 150 + .5(x_1 + x_2)^2 \\ &\geq 0 \\ 0 &\leq X \leq (9.422, 5.903) \end{aligned}$$

The global optimum obtained is: $(x; f) = (6.293429, 3.821839; 201.159334)$.

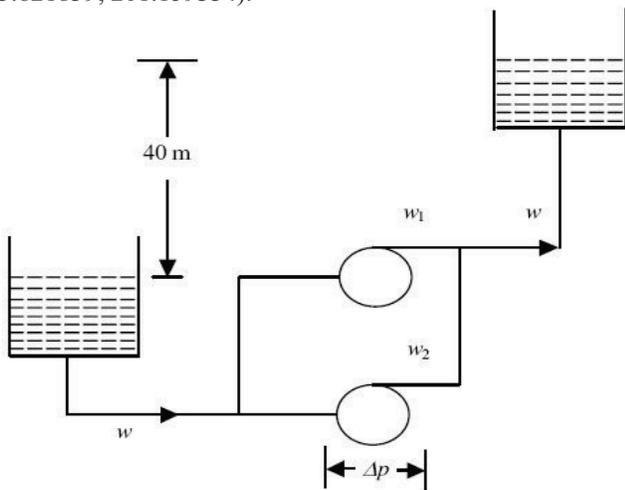


Fig 1: Water Pumping System

B2. Parameter estimation of water quality model [11]

This section represents the mathematical formulation of one-dimensional diffusion pollution problem. Consider a homogeneous stream region, where it is assumed that the flow field is steady. In this circumstance, the general advection-diffusion-reaction equation of the pollutant is defined by [12]

$$\frac{\partial C}{\partial t} + u \frac{\partial C}{\partial X} = D \frac{\partial^2 C}{\partial X^2} - KC$$

(8)

Where $C(X, t)$ (mass/volume) is the concentration of the pollutant at downstream distance X and time t ; X is distance from upstream boundary condition (length); u the cross-sectional mean stream velocity (length/time); D is longitudinal dispersion coefficient (length²/time); K denotes the first order reaction rate of the pollutant (time⁻¹).

Assuming that the pollutant is injected instantaneously into the stream, we can obtain the equation of the pollutant for the process of diffusion, convection, and absorption, as follows:

$$\begin{cases} \frac{\partial C}{\partial t} + u \frac{\partial C}{\partial X} = D \frac{\partial^2 C}{\partial X^2} - KC & (0 < X, 0 < t) \\ C(X, 0) = 0 & (0 < X) \\ C(0, t) = C_0 \delta(t) \quad C(\infty, t) = 0, & (0 \leq t) \end{cases}$$

(9)

Where $C_0 = m/Au$, m the mass of the pollutant, A the mean cross sectional area of the stream. By using Laplace transform, we can easily obtain the analytic solution of (9):

$$C(X, t) = \frac{m}{A\sqrt{4\pi Dt}} \exp\left(-\frac{(X-ut)^2}{4Dt} - Kt\right)$$

(10)

According to the measured histories of the pollutant concentration, estimation of parameter vector $\mathbf{Q} = (u, D, K)$ can be formulated as an optimization problem, whose objective function is the total summation of square error (SSE). The total summation of square error, for the predicted data compared to measured data, can be cast in the discrete normalized form as

$$SSE = \sum_{i=1}^n (C(X_0, t, \mathbf{Q}) - C'(X_0, t))^2$$

Where $C'(X_0, t)$ is the measured data of the pollutant concentration at X_0 , $C(\cdot)$ the predicted data at X_0 , and n the number of data

V. EXPERIMENTAL SETUP AND NUMERICAL RESULTS

A. Experimental Setup

As a general rule, an effective NP (population size) is between $3 * n$ and $10 * n$, but can often be reduced to minimize the NFE. For the present study we performed various experiments with the population size as well as with the crossover rate and mutation probability rate and observed that for problems up to dimension 30 a population size of $10 * n$ is sufficient. Values of scaling factor F , $F=0.5$

is usually a good initial choice. Here we have taken crossover rate $C_r = 0.5$. All the algorithms are executed on a PIV PC, using DEV C++, thirty times for each problem. In every case, a run was terminated when an accuracy 10^{-4} achieved or when the maximum number of function evaluation (NFE= 10^6) was reached.

B. Numerical results

Table I gives the average best fitness function value, standard deviation and number of function evaluation (NFE) after 30 runs for the benchmark functions [6], dimension is taken 30 for scalable functions. From this Table it can be clearly observed that in terms of average fitness function value the proposed scheme performed at par with the original DE and in some cases even gave better results. The finer performance of the proposed versions is more evident from NFE column which gives the average number of function evaluations. From the % improvement column, we can see that the improvement in terms of number of functions evaluations is more than 35% in all the test cases or in other words the proposed ACDE algorithm is 35% faster than the basic DE algorithm.

Numerical result of water pumping system is given in Table II. The experimental setting for this comparison is same as for benchmark problems except $F=0.8$. Method

proposed by Deb [13], is used in the present study to handle constraints. From this table it is clear that both the algorithm gives same solution but ACDE takes less number of functions evaluation to find out the solution in comparison to DE.

Numerical results for parameter estimation of water quality model are summarized in Table III. For the solution of this problem parameters are taken as $X_0 = 8\text{km}$, the injected pollutant $m=378.52\text{g}$, cross sectional area $A=14\text{m}^2$. And the exact values of parameters (u, D, K) were assumed to be fixed at 0.53m/s, 22.0m²/s and 0.065/h respectively. We have taken a sample of 20 points with same interval 0.1158h between 3.2h and 5.4h. From Table III it is clear that ACDE outperforms all the algorithms.

VI. CONCLUSIONS

In this paper we proposed a modified version of basic DE called ACDE. The simulation of results showed that the proposed algorithm is quite competent for solving benchmark problems and real life problems in less time and less number of function evaluations without compromising with the quality of solution. For the test problems, the improvement in NFE is on an average around 45% for benchmark problems. Similarly for real life problems also ACDE outperformed the other algorithms mentioned in literature.

TABLE I. MEAN FITNESS, STANDARD DEVIATION, NUMBER OF FUNCTION EVALUATION AND %IMPROVMENT.

Fun	Fitness and Std.		NFE		% Impro.
	DE	ACDE	DE	ACDE	
f_{GP}	3.00000 1.07046e-06	3.00000 9.37862e-07	970	518	46.59
f_{HS}	-3.86230 1.46942e-06	-3.8623 1.70044e-06	1170	714	38.97
f_{CV}	1.65825e-06 1.86572e-06	2.5184e-006 1.10352e-06	12716	5004	60.64
f_{ACK}	1.42800e-04 1.79218e-05	1.34120e-04 1.07299e-05	259410	154471	40.45
f_{SWF}	7.28960e-04 3.85744e-06	7.19992e-04 5.36615e-06	366570	164846	55.03
f_{GW}	4.62272e-05 9.03396e-06	4.78183e-05 9.31092e-06	224910	128013	43.08
f_{ZA}	4.50199e-05 7.48947e-06	4.79686e-05 1.09593e-05	214890	123047	42.73

TABLE II. SOLUTION FOR WATER PUMPING SYSTEM.

	DE	ACDE	Branch & Reduce[14]
x1	6.29343	6.29343	6.293429
x2	3.82184	3.82184	3.821839
x3	201.159	201.159	201.159334
g1	8.78565e-05	6.79518e-05	N/A
g2	7.8936e-05	2.58852e-05	N/A
f(X)	201.159	201.159	201.159334
NFE	2436	1846	N/A
Time	0.110	0.100	0.30

TABLE III. SOLUTION FOR WATER QUALITY MODEL.

	DE	ACDE	GA[11]	PSO[11]
x1=u	0.530027	0.530003	0.52856	0.53021
x2=D	22.0001	22.0001	20.36604	21.9254
x3=K	0.0669998	0.0670001	0.07087	0.065397
f(X)	3.17559e-11	2.24631e-16	7.4072e-07	9.3019e-09
NFE	6000	6000	6000	6000
Time	0.8	0.82	N/A	N/A

APPENDIX:

1. Goldstein problem (GP):

$$f_{GP}(x) = [1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 (15 - 32x_1 + 12x_1^2 + 42x_2 - 36x_1x_2 + 27x_2^2)]$$

With $-2 \leq x_i \leq 2$, $\min f_{GP}(0, -1) = 3$

It is a multimodal, non separable function having one global minimum and four local minima.

2. Hartman 3 function (H3):

$$f_{H3}(x) = -\sum_{i=1}^4 c_i \exp \left[-\sum_{j=1}^3 a_{ij} (x_j - p_{ij})^2 \right]$$

With $0 \leq x_i \leq 1$, $\min f_{H3}(.114614, .555649, .852547) = -3.862782$

I	c _i	a _{ij}			p _{ij}		
		j=1	2	3	j=1	2	3
1	1	3	10	30	0.3689	0.117	0.2673
2	1.2	.1	10	35	0.4699	0.4387	0.747
3	3	3	10	30	0.1091	0.8732	0.5547
4	3.2	.1	10	35	0.3815	0.5743	0.8828

It's a multimodal no separable function having four local minima and one global minimum.

3. Colville function:

$$f_{CV}(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 + 10.1((x_2 - 1)^2 + (x_4 - 1)^2) + 19.8(x_2 - 1)(x_4 - 1)$$

With $-10 \leq x_i \leq 10$, $\min f_{CV}(1, 1, 1, 1) = 0$

It's a Unimodal no separable function. Its global optimum lies inside a long, narrow parabolic shaped flat valley. It serves a good example for testing the premature convergence of an optimization algorithm.

4. Ackley's function:

$$f_{ACK}(X) = -20 * \exp \left(-2 \sqrt{1/n \sum_{i=1}^n x_i^2} \right) - \exp \left(1/n \sum_{i=1}^n \cos(2\pi x_i) \right) + 20 + e$$

With $-32 \leq x_i \leq 32$, $\min f_{ACK}(0, \dots, 0) = 0$

It is a multimodal function. the presence of an exponential term makes its surface covered with several local minima.

5. Schwefel's problem (SWF):

$$f_{SWF}(x) = 418.9829 \times n - \sum_{i=1}^n x_i \sin(\sqrt{|x_i|})$$

With $-500 \leq x_i \leq 500$, $\min f_{SWF}(s, \dots, s) = 0$

Where $s=420.97$

It is a multimodal function with very deep sinusoidal interactions. It is generally considered to be difficult for optimization algorithms.

6. Griewank function:

$$f_{GW}(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \quad \text{With } -600 \leq x_i \leq 600, \min f_{GW}(0, \dots, 0) = 0$$

It is a continuous multimodal function considered difficult to optimize because of its non-separable nature.

7. Zakharov function (ZA):

$$f_{ZA}(x) = \sum_{i=1}^n x_i^2 + \left(\sum_{i=1}^n 0.5ix_i\right)^2 + \left(\sum_{i=1}^n 0.5ix_i\right)^4 \quad \text{With } -5 \leq x_i \leq 10, \min f_{ZA}(0, \dots, 0) = 0$$

It is a multimodal, separable function having several local minima (exact number unspecified in literature) and one global minimum.

REFERENCES

- [1] R.Storn, and K.Price, DE-a simple and efficient adaptive scheme for global optimization over continuous space, Technical Report TR-95-012, ICSI, March 1995. Available via the Internet: ftp.icsi.berkeley.edu/pub/techreports/1995/tr-95-012.ps.Z, 1995.
- [2] S.Paterlini, T.Krink, High performance clustering with differential evolution. In: Proceedings of the IEEE Congress on Evolutionary Computation, vol. 2, 2004, pp. 2004–2011.
- [3] M.Omran, A.Engelbrecht and A.Salman, Differential evolution methods for unsupervised image classification. In: Proceedings of the IEEE Congress on Evolutionary Computation, vol. 2, 2005a, pp. 966–973.
- [4] R.Storn, Differential evolution design for an IIR-filter with requirements for magnitude and group delay. Technical Report TR-95-026, International Computer Science Institute, Berkeley, CA 1995.
- [5] J. Vesterstroem and R. Thomsen, A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems, *Proc. Congr. Evol. Comput.*, vol. 2, pp. 1980–1987, 2004.
- [6] S. Rahnamayan, H.R. Tizhoosh, and M. M. A. Salama, Opposition-Based Differential Evolution, *IEEE Transactions on Evolutionary Computation*, Vol. 12, Issue 1, pp. 64 – 79, 2008.
- [7] U. K. Chakraborty (Ed.) *Advances in Differential Evolution*, Springer-Verlag, Heidelberg, 2008.
- [8] P.S.Shelokar P.Siarry, V.K.Jayaraman and B.D.Kulkarni, Particle swarm and ant colony algorithm hybridized for continuous optimization, *Applied mathematics and computation* 188, 129-142, 2007.
- [9] W. F. Stoecker, *Design of Thermal Systems*. 3rd ed., McGraw-Hill International edition, Singapore, pp 117-121, 1971
- [10] J. Leibman, L. Lasdon, L. Schrage and A. Waren, *Modeling and optimization with GINO. The Scientific Press*, Palo Alto, CA, 1986.
- [11] K.Wang, X.Wang, J.Wang, G.Lv, M.Jiang, C.Kang and L.Shen, particle swarm optimization for calibrating stream water quality model, second international symposium on intelligent information technology application, pp 682-686, 2008.
- [12] E. M. O.Loughlin, K. H. Bowmer, Dilution and decay of aquatic herbicides in flowing channels, *J. Hydro*, Vol. 26, pp. 217-235, 1975.
- [13] K. Deb, an efficient constraint handling method for genetic algorithm, computer method in applied mechanics and engineering, 186(2/4), pp 311-338, 2000.
- [14] H. S. Ryoo, and N. V. Sahinidis, Global optimization of nonconvex NLPs and MINLPs with Applications in Process Design. *Computers & Chemical Engineering*, 19(5), 551-566, 1995.