DOI: http://dx.doi.org/10.7551/978-0-262-32621-6-ch102

Constrained Group Counseling Optimization

Mohammad A. Eita[1], Amin B. Shoukry[1,2] and Hitoshi C. Iba[3]

[1]Comptuer Science and Engineering Department, Egypt-Japan University of Science and Technology (E-JUST), Alex, EGYPT
[2]Computer and Systems Engineering Department, Alexandria University, Alex, EGYPT
[3]Graduate School of Information Science and Technology, The University of Tokyo, Tokyo, Japan
mohammad.eita@ejust.edu.eg

Abstract

Group Counseling Optimization (GCO) has recently been proposed in an attempt to emulate the human social behavior in solving life problems through *counseling within a group*. After its promising results in solving unconstrained single-objective and multi-objective optimization problems, in this paper, GCO is extended to solve the constrained optimization problems for the first time. Also, a hybrid parameter-less constraint handling technique is proposed, which uses two well-known constraint handling techniques: feasible rules and penalty function. The Constrained Group Counseling Optimization (CGCO) uses gradient-based mutation only in the case of all-equality-constraints COPs to reach the extremely small feasible region easily. Moreover, CGCO performance is tested by solving the constrained benchmarks function of the CEC 2010 competition. The results demonstrate that CGCO is competitive to other state-of-the-art algorithms and consistently reaches feasible solutions.

Introduction

Many real-world applications necessitate the solution of Constrained Optimization Problems (COPs). The solution of such problems means optimizing a given objective function while satisfying a set of imposed constraints. A COP can be defined as follows (Mezura-Montes and Coello C., 2011):

$$\begin{aligned} &\textit{minimize} && f(x) \\ &\textit{subject to} && g_j(x) \leq 0,\ j = 1,...,q \\ &&& h_j(x) = 0,\ j = q+1,...,m \\ &&& L_d \leq x_d \leq U_d,\ d = 1,...,D \end{aligned} \quad (1)$$

where $x = (x_1, x_2, ..., x_D) \in R^D$ is a real-valued D-dimensional vector, f is the real valued objective function, g_j and h_j are q inequality constraints and $(m-q)$ equality constraints, respectively, L_d and U_d are the lower and upper bounds of x_d, respectively.

To solve COPs, researchers used well-known nature-inspired algorithms such as Particle Swarm optimization (PSO) (Kennedy et al., 1995), and Differential Evolution (DE) (Storn and Price, 1997), etc. Originally, all of these algorithms are mainly proposed to deal with unconstrained optimization problems so that it should add a Constraint Handling Technique (CHT) to enable them to deal with COPs.

Currently, seven categories of CHTs are known in the literature (Mezura-Montes and Coello C., 2011): Feasibility Rules (FR) (Deb, 2000), Stochastic Ranking (SR) (Runarsson and Yao, 2000), ε-constrained method (Takahama et al., 2005), Novel Penalty Functions, Novel special operators, Multi-objective concepts, Ensemble of constraint-handling techniques. In FR, a solution with less constraint violation is preferred. If two solutions have the same value of constraint violation, the fitter solution is preferred. In SR, a user-defined parameter called pf determines which criterion to use when comparing infeasible solutions: (1) based on their sum of constraints violation or (2) based only on their objective function values. In ε-constrained method, the value of ε> 0 relaxes the limit of considering a solution as feasible. In Novel Penalty Functions, researchers recently proposed two penalty-based approaches namely adaptive penalty function and dynamic penalty function. Adaptive penalty function (Tessema and Yen, 2009) calculates the penalty factor based on the status of the candidate solutions in the search space. Dynamic penalty functions (Tasgetiren and Suganthan, 2006) adopts the current generation number to decrease the penalty factor. In Novel special operators, operator such as boundary operator (Leguizamón and Coello C., 2009) is suggested. In Multi-objective concepts, a COP is turned into a bi-objective optimization problem (objective function and sum of constraints violation) (Wang et al., 2007). In ensemble of constraint-handling techniques (Mallipeddi and Suganthan, 2010a), more than one of the aforementioned categories are hybridized to get the advantages of each category.

For simplicity, the Constrained Group Counseling Optimization (CGCO) proposes the use of two parameter-less CHTs: FR and penalty function without any penalty factor. Gradient-based mutation operator (Takahama and Sakai, 2006) is also used to deal with COPs that have only equality constraints. CGCO is applied to a set of standard benchmark

COPs of the IEEE CEC 2010 competition (Mallipeddi and Suganthan, 2010b) and the results are compared with εDEag (Takahama and Sakai, 2010), the winner of this competition, and another recently proposed algorithm Co-CLPSO (Liang et al., 2010).

This paper is organized as follows: section 2 outlines the related work. Section 3 provides an overview of GCO. Section 4 presents an overview of the CHTs and gradient-based mutation used by CGCO. Section 5 introduces the proposed CGCO. In section 6, CGCO is tested on COPs to evaluate its performance. Finally, the conclusions and future work are put forward in section 7.

Related Works

Here, some well-known nature-inspired algorithms, namely DE (Storn and Price, 1997), PSO (Kennedy et al., 1995) are briefly addressed.

In (Takahama and Sakai, 2006), an approach, called εDE, has been suggested to solve COPs using ε-constrained method as CHT and gradient-based mutation as a repair operator. Gradient-based mutation helps εDE handle COPs whose constraints are equality. This kind of COPs is difficult because the feasible region is very small. Afterwards, Takahama and Sakai added the concept of archive in their new approach εDEag (Takahama and Sakai, 2010). The archive increases the diversity of candidate solutions so that it gives better stability. In εDEag, a new controlling method of ε level is adopted. εDEag yielded promising results and was the winner of the CEC2010 competition (Mallipeddi and Suganthan, 2010b).

Liang et al. proposed an approach using PSO to solve COPs which is called cooperation comprehensive learning PSO (Co-CLPSO) (Liang et al., 2010). In Co-CLPSO, a novel CHT in which the population is divided into two sub-swarms is used. These two sub-swarms cooperate with each other in an attempt to solve the COPs. Particles of each swarm are responsible to deal with different constraints. Two swarms exchange their experiences to benefit each other. Sequential quadratic programming (SQP) is used as a local optimizer to improve the obtained solutions. Co-CLPSO ranked the fifth position in the CEC2010 competition.

Group Counseling Optimization

"Instead of mimicking the behavior of biological organisms such as birds, fish, ants, and bees, GCO is inspired by the human social behavior in solving life problems through *counseling within a group*" (Eita and Fahmy, 2010, 2014). Counseling (Burnard, 2002) is a well-established branch in sociology and psychology. This was the first time that a connection is found between population-based optimization and group counseling (Berg et al., 2006). Based on the group counseling concept, GCO is developed (Eita and Fahmy, 2010, 2014).

Four parameters affect the behavior of GCO:

- Number of group members acting as counselors, c, $(c \leq m{-}1)$.
- Counseling probability, cp.
- Search range reduction coefficient, *red*, set into the range [0,1].
- Transition rate from the stage of exploration to that of exploitation, tr.

The GCO algorithm is illustrated in the following steps:

Step 1

At the very beginning, the algorithm initializes randomly a population with m D-dimensional candidate solutions X^i in the search space according to a beta distribution (Gentle, 2003); (Owen, 2008),

$$\begin{aligned}\beta(x) &= \frac{x^{a-1}(1-x)^{b-1}}{B(a,b)} \quad 0<x<1 \\ B(a,b) &= \int_0^1 t^{a-1}(1-t)^{b-1}dt \qquad (2)\end{aligned}$$

The case that both shaping parameters a and b are equal and less than unity, $a = b < 1$, is adopted. According to the characteristics of beta distribution, the density function is symmetric and U-shaped. So, most of the candidate solutions lie near the search space boundaries. This implies that the global optimum is surrounded by these candidate solutions. Due to Other-members counseling strategy, as mentioned later in Step 3a, GCO moves from the search space boundaries to inside so that GCO most probably reaches global optimum. Shaping parameters a and b are often set to 0.1 as is the case in this paper.

Step 2

The objective function $f(X)$ is evaluated for each solution X^i , producing m fitness values $f(X^i)$.

Step 3

For each solution X^i, an alternative solution X'^i is generated

$$X'^i = (x'^i_1, x'^i_2, ..., x'^i_D) \qquad (3)$$

Each component x'^i_d is obtained via one of two counseling strategies:

(a) *Other-members counseling*

(b) *Self-counseling*

A random number in the range [0,1] is generated, according to a uniform distribution, for each component x^i_d. It is named a *counseling decisive coefficient* (cdc). If $cdc \leq cp$, other-members counseling is used; otherwise, self-counseling is used. In the following, the production process of x'^i_d, through these strategies, is demonstrated.

Step 3a: Other-members counseling ($cdc \leq cp$)

Here, candidate solution X^i acts as a counselee. It asks for counseling of c other members, that act as counselors, selected randomly out of the population. The value of $x_d^{\prime i}$ is computed as *weighted* summation of the corresponding components (best experiences) of the c counselors. The weights are the contributions of the relevant counselors, in a brainstorming process. The weight, w_q, of component d of counselor $q(q = 1,2,...,c)$ is a random number in the range [0,1] according to a uniform distribution. Counselor q may be any member i. Eq. (4) constrains the summation of the c weights to unity,

$$\sum_{q=1}^{c} w_q = 1 \tag{4}$$

The component $x_d^{\prime i}$ is calculated as

$$x_d^{\prime i} = \sum_{q=1}^{c} w_q . x_d^{int\_rand_q} \tag{5}$$

for all components of all candidate solutions. int_rand_q is an integer random number in the range $[1,m]$, generated according to a uniform distribution. Eq. (5) indicates that c random numbers are generated for each component x_d^i . Also, $x_d^{int\_rand_q}$ is the value of component d of counselor q being member int_rand_q. For each component, the set of c counselors differs. Furthermore, both w_q and int_rand_q varies for each value of i and d.

Step 3b: Self-counseling $(cdc > cp)$

An alternative component $x_d^{\prime i}$ is produced by searching around the current component x_d^i (best experience). The search range length $range\_length_d$ is computed as,

$$range\_length_d = U_d - L_d \tag{6}$$

The searching process is achieved through modification of the component within a *reduced* range with length red. red is the search range reduction coefficient, set in the range [0,1]. We modify component d in a *maximum* range $[-mdf\_max_d, mdf\_max_d]$ about the current component, where

$$mdf\_max_d = 0.5 \ \ red \ \ range\_length_d \tag{7}$$

The maximum modification value at the current iteration, $mdf\_max_d^{itr}$, is calculated as

$$mdf\_max_d^{itr} = mdf\_max_d \, (1 - \frac{itr}{itr\_max})^{tr} \tag{8}$$

where itr is the current iteration number, itr_max is the maximum number of iterations, tr is the transition rate at which GCO behavior transfers from exploration to exploitation. Eq. (8) indicates that $mdf\_max_d^{itr}$ varies from mdf_max_d to zero. This means that when itr proceeds, GCO converts from exploration phase to exploitation phase.

For each component x_d^i, a random number is generated in the range $[-mdf\_max_d^{itr}, mdf\_max_d^{itr}]$ and then added to x_d^i to get the modified value $x_d^{\prime i}$ as follows

$$x_d^{\prime i} = x_d^i + rand_d^i(-mdf\_max_d^{itr}, mdf\_max_d^{itr}) \tag{9}$$

Step 4

The fitness value of $X^{\prime i}$, $f(X^{\prime i})$, is evaluated. If $f(X^{\prime i})$ is better than $f(X^i)$, then $X^{\prime i}$ replaces X^i; otherwise, $X^{\prime i}$ is ignored and X^i is retained unchanged.

Repetition Steps (iterations)

Steps 3 and 4 are repeated until the stopping criterion is met.

Final Step

In the last iteration, GCO makes the decision by selecting the best solution out of m solutions.

Constrained Handling Techniques

In this paper, the sum of constraints violation of a solution x is defined as

$$CV(x) = \sum_{j=1}^{q} max(0, g_j(x)) + \sum_{j=q+1}^{m} max(0, |h_j(x)| - \delta) \tag{10}$$

where δ is a very small positive number. In the experimental section of this paper, δ is set to 0.0001.

Here, we address three CHTs used in our proposed algorithm: namely FR, parameter-less penalty function, and gradient-based mutation.

Feasibility Rules (FR)

In (Deb, 2000), the author suggested FR as a CHT. FR is popular in this field because it can be added to any nature-inspired algorithm without adding any new parameter (parameter-less). To compare two solutions, FR employs three feasibility criteria as follows: (1) for two feasible solutions, the solution with the better objective function is preferred, (2) for a feasible and an infeasible solution, the feasible solution is preferred, (3) for two infeasible solutions, the solution with the lower sum of constraint violation is preferred.

Parameter-less Penalty Function

Penalty function transforms COPs to unconstrained optimization problems which are easy to deal with. Due to its simple implementation, penalty function technique is popular and it is used by a lot of researchers.

In (Debchoudhury et al., 2013), the authors used the simple penalty function without any penalty factor that needs to be adjusted, as follows

$$\phi(x) = f(x) + CV(x) \tag{11}$$

where $f(x)$ is the objective function, $CV(x)$ is the sum of constraints violation, $\phi(x)$ is the modified objective function.

Gradient-based Mutation

Gradient-based mutation (Takahama and Sakai, 2006, 2010) is a repair operator, by which an infeasible solution is converted to a feasible solution. The constraint functions vector $C(x)$ and the constraint violation vector $\Delta C(x)$ are as follows,

$$C(x) = [g_1(x), ..., g_q(x), h_{q+1}(x), ..., h_m(x)]^T \quad (12)$$

$$\Delta C(x) = [\Delta g_1(x), ..., \Delta g_q(x), h_{q+1}(x), ..., h_m(x)]^T \quad (13)$$

where $\Delta g_j(x) = max(0, g_j(x))$.

The linear approximation of the constraint violation vector $\Delta C(x)$ through Taylor expansion is defined as follows,

$$\nabla C(x)\ \Delta x = -\Delta C(x) \quad (14)$$

$$\Delta x = -\nabla C(x)^{-1}\ \Delta C(x) \quad (15)$$

where Δx is the change in x required to make solution x feasible, $\nabla C(x)$ is the gradient of the constraint functions which is numerically approximated as follows:

$$\nabla C(x) = \frac{1}{\eta}[C(x+\eta e_1) - C(x), ..., C(x+\eta e_n) - C(x)] \quad (16)$$

where η is a very small number, e_i is the unit vector in the direction of dimension i. Generally, $\nabla C(x)$ matrix is not square, so the pseudo inverse (Campbell and Meyer, 2009) is the best numerical approximation of $\nabla C(x)^{-1}$. The new value of x is calculated by the following formula,

$$x^{new} = x^{old} + \Delta x \quad (17)$$

Proposed CGCO

After GCO gave promising results in solving unconstrained single-objective (Eita and Fahmy, 2010, 2014) and multi-objective function (Ali and Khan, 2013), GCO is extended, here, to solve COPs. In the CGCO algorithm, FR and parameter-less penalty function are hybridized as a proposed novel CHT. The hybridization is adopted to gain the advantages of FR and parameter-less penalty function and to discard their disadvantages. **Table 1** summarizes both the advantages and disadvantages of various CHTs (Mani and Patvardhan, 2009). It should be obvious that in addition to the merits of the penalty function technique, the parameter-less penalty function does not need to adjust any penalty factor.

Table 1: Comparison Among Various CHTs (Mani and Patvardhan, 2009)

Item	Gradient based mutation	Penalty Function	Parameter-less Penalty Function	FR	Proposed CHT
Computationally expensive	Yes	No	No	No	No
More feasible solution wins	Yes	No	No	Yes	Yes
Exploration	Good	Good	Good	Poor	Good
Parameter-less	No	No	Yes	Yes	Yes
Simple implementation	No	Yes	Yes	Yes	Yes

CGCO possesses a good exploration capability, avoids premature convergence of FR (Mezura-Montes and Coello C., 2011), using the parameter-less penalty function and can pick the more feasible solutions using FR and stores it into X_FR. X_FR replaces a solution from the population other than the best solution according to modified objective function so that evolution process benefits from the more feasible solution obtained by FR.

Gradient-based mutation is a computationally expensive repair operator. When the feasibility region is extremely small as is the case in COPs which have only equality constraints, gradient-based mutation helps iteratively the CGCO to reach feasibility. Due to the nonlinearity of the constraints and the approximation given in Eq.(14), gradient-based mutation is applied to a solution iteratively to increase the feasibility of the solution. N is the maximum number of iterations of gradient-based mutation.

Algorithms 1 and 2 describe the pseudo code of the CGCO algorithm and gradient-based mutation operator respectively.

Experiments and Results

In this section, the CGCO performance is tested through solving 18 10-D and 18 30-D constrained benchmark functions of IEEE CEC 2010 competition (Mallipeddi and Suganthan, 2010b). Then the comparison is made with two competitive algorithms: εDEag (Takahama and Sakai, 2010) and Co-CLPSO (Liang et al., 2010). εDEag and Co-CLPSO ranked the first and the fifth position in this competition, respectively. For each function, CGCO is run for 25 independent runs. The algorithm is coded in C++, and run on a PC with Windows 7 OS, a 4 GB Ram, Intel 2.67 GHz core i7 processor. The stopping criterion is the maximum number of function evaluations FE_{max}. For 10-D functions, FE_{max} is equal to 200,000 FEs. For 30-D functions, FE_{max} is equal to 600,000 FEs.

CGCO uses the following parameters: Population size $m = 40, c = 2, cp = 0.01$ (this value has been experimen-

Algorithm 1 The CGCO algorithm, where FE is the number of current function evaluation, $u(a,b)$ is a uniform random number in $[a,b]$

1. Initialization of the population of solutions $(X^1,\ldots,X^m)$ according to beta distribution
2. For each solution, evaluation of objective function $f(X^i)$, sum of constraint violation $CV(X^i)$, modified objective function $\phi(X^i)$
3. According to FR, determining the index of the best solution r, and assigned X^r to X_FR ($X\_FR = X^r$)
4. **repeat**{
5.   **for**($i = 1; i \le m; i{+}{+}$){
6.   **for**($j = 1; j \le D; j{+}{+}$){
7. $cdc = u(0,1)$
8.   **if**($cdc \le cp$){
9.   Excluding solution i, select randomly different c solutions
10. $x'^i_d = \sum_{q=1}^{c} w_q . x_d^{int\_rand_q}$
11.   }**else**{
12. $mdf\_max_d^{FE} = mdf\_max_d\,(1 - \frac{FE}{FE\_max})^{tr}$
13.   **repeat**{
14. $x'^i_d = x^i_d + rand^i_d(-mdf\_max_d^{FE}, mdf\_max_d^{FE})$
15.   }**until**(x'^i_d inside the allowable range)
16.   }
17.   Evaluation of $f(X'^i), CV(X'^i), \phi(X'^i)$; $FE = FE + 1$
18.   **if**($\phi(X'^i) < \phi(X^i))X^i = X'^i$
19.   According to FR, **if**(X'^i better than X_FR){$X\_FR = X'^i$}
20.   **if**(all constraints are equality && $\phi(X^i)$ is the best value){
21.   Doing gradient-based mutation to X'^i as in **Algorithm 2**
22.   }
23.   }
24. Determining the index of the best solution of the population according to modified objective function, *best*
25. **if**($r == best$){   // Injection of X_FR into the population
26.   Generating randomly a new value of $r \in [1,m]$ such that $r \neq best$
27. $X^r = X\_FR$
28. }**else**{
29. $X^r = X\_FR$
30. }
31. }**until**($FE == FEmax$)
32. According to FR, comparing(X_FR, X^{best}) and output the best one

Algorithm 2 Gradient-based mutation to X'^i

1. **for**($n = 1$; X'^i is infeasible && $n \le N; n{+}{+}$){
2. $\Delta X'^i = -\nabla C(X'^i)^{-1}\ \Delta C(X'^i)$
3.   **if**($X'^i + \Delta X'^i$ is inside the allowable range){
4. $X'^i = X'^i + \Delta X'^i$
5.   }**else**{
6.   Exit from the loop
7.   }
8.   Evaluation of $f(X'^i), CV(X'^i), \phi(X'^i)$
9.   **if**($\phi(X'^i) < \phi(X^i)$){ $X^i = X'^i$ }
10.   According to FR, if(X'^i better than X_FR){$X\_FR = X'^i$}
11. $FE = FE + D + 1$
12. }

tally found suitable for all functions except 30-D C01&C02 for which the value 0.1 has given better performance), $red = 0.5, tr = 7, N = 100$. The results and feasibility rate for 10-D and 30-D are given in **Appendix A**.

From **Appendix A**, it is significant to mention that the feasibility rate of CGCO is 100% for all 18 10-D and 18 30-D test functions. On the other hand, the feasibility rate of εDEag is 100% for only 35 test functions and 12% for 30-D C12. Also, the feasibility rate of Co-CLPSO is 100% for only 31 test functions and 0% for 10-D C11, 0% for 30-D C03, 80% for 30-D C04, 0% for 30-D C11, and 92% for 30-D C12. Also, **Appendix A** shows that εDEag results are slightly better than CGCO results for only 10-D test functions.

Table 2 outlines the comparison of CGCO vs. εDEag and Co-CLPSO. Wilcoxon Signed Rank Test for 5% significance level is performed. Three symbols (+, -, and =) are used in this respect. The symbol ”+” denotes that the first algorithm is significantly better than the second one, the symbol “-“ indicates that the first algorithm is significantly worse than the second, and the symbol “=” means that there is no significant difference between the two algorithms. The table shows that, for 10-D functions, CGCO is competitive to Co-CLPSO. But for 30-D functions, CGCO is competitive to εDEag and outperforms Co-CLPSO based on the average value. This case is expected according to *no free lunch* theorem. Here, it is important to report the following points: 1) CGCO does not use any local optimizer and depends only on its behavior to reach the optimum, while Co-CLPSO uses a local optimizer called Sequential Quadratic Programming (SQP) to improve the quality of the solutions (Liang et al., 2010). 2) CGCO does not use any archive, while

Table 2: COMPARISON OF CGCO *vs.* εDEag and Co-CLPSO

Dim	Algorithms	Objective Function	Better	Equal	Worse	Test
10 D	CGCO vs. εDEag	Best	3	3	12	-
		Average	2	2	14	-
	CGCO vs. Co-CLPSO	Best	5	0	13	=
		Average	10	0	8	=
30 D	CGCO vs. εDEag	Best	11	0	7	=
		Average	9	0	9	=
	CGCO vs. Co-CLPSO	Best	9	0	9	=
		Average	14	0	4	+

εDEag uses an archive to preserve the diversity of the solutions (Takahama and Sakai, 2010). 3) CGCO employed the gradient-based mutation only for 14 all-equality-constraints COPs (C03, C04, C05, C06, C09, C10, and C11) with 10-D and 30-D, while εDEag employed the gradient-based mutation for all 36 test functions (Takahama and Sakai, 2010). 4) CGCO adopts a parameter-less CHT for 22 test functions and needs to set only one parameter 'N' of gradient-based mutation for 14 all-equality-constraints COPs, while εDEag needs to set 4 parameters for its CHT (2 parameters for ε-constrained method and 2 parameters for gradient-based mutation) (Takahama and Sakai, 2010).

Conclusions

Group Counseling optimization (GCO) has basically been introduced to solve unconstrained optimization problems. In this paper, the Constrained Group Counseling Optimization (CGCO) is presented to solve COPs. CGCO adopts a new parameter-less CHT in which hybridization between FR and penalty function techniques is implemented. In addition to this, gradient-based mutation is used in the case of COPs that have only equality constraints so that the feasible solution is obtained quickly. The proposed algorithm is applied to the constrained benchmark functions of the CEC 2010 competition. A comparison is made between CGCO and two well-known algorithms. The results show that the performance of CGCO is competitive and the feasibility rate is always 100% for all functions.

For future work, the behavior of CGCO needs to be studied with other CHTs such as SR and ε-constrained method. The applicability of the CGCO to real-world applications should be, also, investigated.

Acknowledgment

This research is supported by the Ministry of Higher Education (MoHE) of Egypt through PhD fellowships. Our sincere thanks to E-JUST University for guidance and support.

References

Ali, H. and Khan, F. A. (2013). Group counseling optimization for multi-objective functions. In *Evolutionary Computation (CEC), 2013 IEEE Congress on*, pages 705–712. IEEE.

Berg, R. C., Landreth, G. L., and Fall, K. A. (2006). *Group Counseling: Concepts and Procedures Fourth Edition*. Routledge.

Burnard, P. (2002). *Practical counselling and helping*. Routledge.

Campbell, S. L. and Meyer, C. D. (2009). *Generalized inverses of linear transformations*, volume 56. SIAM.

Deb, K. (2000). An efficient constraint handling method for genetic algorithms. *Computer methods in applied mechanics and engineering*, 186(2):311–338.

Debchoudhury, S., Biswas, S., Kundu, S., Das, S., Vasilakos, A. V., and Mondal, A. (2013). Modified estimation of distribution algorithm with differential mutation for constrained optimization. In *Evolutionary Computation (CEC), 2013 IEEE Congress on*, pages 1724–1731. IEEE.

Eita, M. A. and Fahmy, M. M. (2010). Group counseling optimization: A novel approach. In *Research and Development in Intelligent Systems XXVI*, pages 195–208. Springer.

Eita, M. A. and Fahmy, M. M. (2014). Group counseling optimization. *Applied Soft Computing*.

Gentle, J. E. (2003). *Random number generation and Monte Carlo methods*. Springer.

Kennedy, J., Eberhart, R., et al. (1995). Particle swarm optimization. In *Proceedings of IEEE international conference on neural networks*, volume 4, pages 1942–1948. Perth, Australia.

Leguizamón, G. and Coello C., C. A. (2009). Boundary search for constrained numerical optimization problems with an algorithm inspired by the ant colony metaphor. *Evolutionary Computation, IEEE Transactions on*, 13(2):350–368.

Liang, J. J., Zhigang, S., and Zhihui, L. (2010). Coevolutionary comprehensive learning particle swarm optimizer. In *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pages 1–8. IEEE.

Mallipeddi, R. and Suganthan, P. N. (2010a). Ensemble of constraint handling techniques. *Evolutionary Computation, IEEE Transactions on*, 14(4):561–579.

Mallipeddi, R. and Suganthan, P. N. (2010b). Problem definitions and evaluation criteria for the cec 2010 competition and special session on single objective constrained real-parameter optimization. *Nangyang Technological University, Singapore*.

Mani, A. and Patvardhan, C. (2009). A novel hybrid constraint handling technique for evolutionary optimization. In *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*, pages 2577–2583. IEEE.

Mezura-Montes, E. and Coello C., C. A. (2011). Constraint-handling in nature-inspired numerical optimization: past, present and future. *Swarm and Evolutionary Computation*, 1(4):173–194.

Owen, C. (2008). Parameter estimation for the beta distribution. Master's thesis, Brigham Young University, Utah, USA.

Runarsson, T. P. and Yao, X. (2000). Stochastic ranking for constrained evolutionary optimization. *Evolutionary Computation, IEEE Transactions on*, 4(3):284–294.

Storn, R. and Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359.

Takahama, T. and Sakai, S. (2006). Constrained optimization by the &# 949; constrained differential evolution with gradient-based mutation and feasible elites. In *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, pages 1–8. IEEE.

Takahama, T. and Sakai, S. (2010). Constrained optimization by the ε constrained differential evolution with an archive and gradient-based mutation. In *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pages 1–9. IEEE.

Takahama, T., Sakai, S., and Iwane, N. (2005). Constrained optimization by the ε constrained hybrid algorithm of particle swarm optimization and genetic algorithm. In *AI 2005: Advances in Artificial Intelligence*, pages 389–400. Springer.

Tasgetiren, M. F. and Suganthan, P. N. (2006). A multi-populated differential evolution algorithm for solving constrained optimization problem. In *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, pages 33–40. IEEE.

Tessema, B. and Yen, G. G. (2009). An adaptive penalty formulation for constrained evolutionary optimization. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 39(3):565–578.

Wang, Y., Cai, Z., Guo, G., and Zhou, Y. (2007). Multiobjective optimization and hybrid evolutionary algorithm to solve constrained optimization problems. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 37(3):560–575.

Appendix A

FUNCTION VALUES ACHIEVED BY CGCO, εDEag, Co-CLPSO FOR THE CEC2010 TEST PROBLEMS

Problem	Algorithm	10D				30D			
		Feasibility Rate	**Best**	**Mean**	**SD**	**Feasibility Rate**	**Best**	**Mean**	**SD**
C01	CGCO	100	-7.473104E-01	-7.452169E-01	4.991886E-03	100	-8.218384E-01	-8.189027E-01	4.635651E-03
	εDEag	100	-7.473104E-01	-7.470402E-01	1.323339E-03	100	-8.218255E-01	-8.208687E-01	7.103893E-04
	Co-CLPSO	100	-7.4731E-01	-7.3358E-01	1.7848E-02	100	-8.0688E-01	-7.1598E-01	5.0252E-02
C02	CGCO	100	-2.277704E+00	-2.246991E+00	1.264116E-02	100	-2.214588E+00	-2.158273E+00	5.783516E-02
	εDEag	100	-2.277702E+00	-2.269502E+00	2.3897790E-02	100	-2.169248E+00	-2.151424E+00	1.197582E-02
	Co-CLPSO	100	-2.2777	-2.2666	1.4616E-02	100	-2.2809	-2.2029	1.9267E-01
C03	CGCO	100	3.032171E-06	2.737753E-05	1.542211E-05	100	2.540517E-03	4.620062E+00	1.072924E+01
	εDEag	100	0.00E+00	0.00E+00	0.00E+00	100	2.867347E+01	2.883785E+01	8.047159E-01
	Co-CLPSO	100	2.4748E-13	3.5502E-01	1.77510	0	N/A	N/A	N/A
C04	CGCO	100	-9.989973E-06	-8.374013E-06	1.031334E-06	100	-1.777536E-07	5.145977E-06	3.897395E-06
	εDEag	100	-9.992345E-06	-9.918452E-06	1.54673E-07	100	4.698111E-03	8.162973E-03	3.067785E-03
	Co-CLPSO	100	-1.00E-05	-9.3385E-06	1.0748E-06	80	-2.93E-06	1.1269E-01	5.6335E-01
C05	CGCO	100	-4.836106E+02	-4.836106E+02	0.00E+00	100	-4.764488E+02	-4.508943E+02	1.445262E+01
	εDEag	100	-4.836106E+02	-4.836106E+02	3.89035E-13	100	-4.531307E+02	-4.495460E+02	2.899105E+00
	Co-CLPSO	100	-4.8361E+02	-4.8360E+02	1.9577E-02	100	-4.8360E+02	-3.1249E+02	8.8332E+01
C06	CGCO	100	-5.786585E+02	-5.786703E+02	7.433578E-02	100	-5.208747E+02	-4.462454E+02	6.149371E+01
	εDEag	100	-5.786581E+02	-5.786528E+02	3.6271690E-03	100	-5.285750E+02	-5.279068E+02	4.748378E-01
	Co-CLPSO	100	-5.7866E+02	-5.7866E+02	5.7289E-04	100	-2.8601E+02	-2.4470E+02	3.9481E+01
C07	CGCO	100	1.344985E-07	2.703616E+00	1.454563E+00	100	9.906752E+00	2.302078E+01	1.368257E+01
	εDEag	100	0.00E+00	0.00E+00	0.00E+00	100	1.147112E-15	2.603632E-15	1.233430E-15
	Co-CLPSO	100	1.0711E-09	7.9732E-01	1.6275	100	3.7861E-11	1.1163	1.8269
C08	CGCO	100	1.245218E-02	7.161994E+00	3.581421E+00	100	1.799808E+01	2.286629E+01	2.142798E+00
	εDEag	100	0.00E+00	6.727528E+00	5.560648E+00	100	2.518693E-14	7.831464E-14	4.855177E-14
	Co-CLPSO	100	9.6442E-10	6.0876E-01	1.4255	100	4.3114E-14	4.7517E+01	1.1259E+02
C09	CGCO	100	7.646936E-16	3.614285-08	6.843696E-08	100	3.888829E-09	1.25734E+01	3.2612E+01
	εDEag	100	0.00E+00	0.00E+00	0.00E+00	100	2.770665E-16	1.072140E+01	2.821923E+01
	Co-CLPSO	100	3.7551E-16	1.9938E+10	9.9688E+10	100	1.9695E+02	1.4822E+08	2.4509E+08
C10	CGCO	100	1.466726E-07	3.403369E+01	1.580944E+01	100	3.146641E+01	3.414345E+01	1.895779E+00
	εDEag	100	0.00E+00	0.00E+00	0.00E+00	100	3.252002E+01	3.326175E+01	4.545577E-01
	Co-CLPSO	100	2.3967E-15	4.9743E+10	2.4871E+11	100	3.1967E+01	1.3951E+09	5.8438E+09
C11	CGCO	100	-1.516158E-03	-1.047986E-03	6.982165E-04	100	-3.800417E-04	-3.754486E-04	3.625592E-06
	εDEag	100	-1.52271E-03	-1.52271E-03	6.3410350E-11	100	-3.268462E-04	-2.863882E-04	2.707605E-05
	Co-CLPSO	0	N/A	N/A	N/A	0	N/A	N/A	N/A
C12	CGCO	100	-1.973112E-01	-1.221012E-01	9.193794E-02	100	-1.983111E-01	8.844418E-02	5.089598E-01
	εDEag	100	-5.700899E+02	-3.367349E+02	1.7821660E+02	12	-1.991453E-01	3.562330E+02	2.889253E+02
	Co-CLPSO	100	-1.2639E+01	-2.3369	2.4329E+01	92	-1.9926E-01	-1.9911E-01	1.1840E-04
C13	CGCO	100	-6.842937E+01	-6.842936E+01	4.627696E-06	100	-6.842914E+01	-6.83411E+01	2.380819E-01
	εDEag	100	-6.842937E+01	-6.842936E+01	1.02596E-06	100	-6.642473E+01	-6.535310E+01	5.733005E-01
	Co-CLPSO	100	-6.842936E+01	-6.397445E+01	2.13408E+00	100	-6.2752E+01	-6.0774E+01	1.1176
C14	CGCO	100	1.790238E-01	3.734062E+00	1.164704E+00	100	1.800895E+01	2.566797E+01	1.331492E+01
	εDEag	100	0.00E+00	0.00E+00	0.00E+00	100	5.015863E-14	3.089407E-13	5.608409E-13
	Co-CLPSO	100	5.78E-12	3.1893E-01	1.1038	100	3.28834e-09	0.0615242	0.307356
C15	CGCO	100	1.800656E-02	3.155997E+00	1.253508E+00	100	2.08679E+01	2.365358E+01	1.199714E+00
	εDEag	100	0.00E+00	1.798980E-01	8.813156E-01	100	2.160345E+01	2.160376E+01	1.104834E-04
	Co-CLPSO	100	3.0469E-12	2.9885	3.3147	100	5.7499E-12	5.1059E+01	9.1759E+01
C16	CGCO	100	5.281739E-02	5.37987E-01	2.417903E-01	100	1.063727E+00	1.077787E+00	1.077612E-02
	εDEag	100	0.00E+00	3.702054E-01	3.7104790E-01	100	0.00E+00	2.168404E-21	1.062297E-20
	Co-CLPSO	100	0.00E+00	5.9861E-03	1.3315E-02	100	0.00E+00	5.2403E-16	4.6722E-16
C17	CGCO	100	1.395648E-11	2.498691E-03	3.819487E-03	100	5.2054121E-02	3.693843E-01	1.766043E-01
	εDEag	100	1.46318E-17	1.249561E-01	1.937197E-01	100	2.165719E-01	6.326487E+00	4.986691E+00
	Co-CLPSO	100	7.6677E-17	3.7986E-01	4.5284E-01	100	1.5787E-01	1.3919	4.2621
C18	CGCO	100	1.984565E-20	3.57925E-14	8.240038E-14	100	3.22061E-02	2.726909E+00	3.697505E+00
	εDEag	100	3.73144E-20	9.678765E-19	1.811234E-18	100	1.226054E+00	8.754569E+01	1.664753E+02
	Co-CLPSO	100	7.7804E-21	2.3192E-01	9.9559E-01	100	6.0047E-02	1.0877E+01	3.7161E+01