2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

Evaluating the performance of Group Counseling Optimizer on CEC 2014 problems for Computational Expensive Optimization

Subhodip Biswas, Mohammad A. Eita, Swagatam Das and Athanasios V. Vasilakos

***Abstract*— Group Counseling Optimizer (GCO) is a recently proposed population-based metaheuristics that simulates the ability of human beings to solve problems through counseling within a group. It is motivated by the fact that the human thinking ability is often predicted to be the most rational. This research article examines the performance of GCO on the benchmark test suite designed for the CEC 2014 Competition for Computational Expensive Optimization. Experimental results on 24 black-box optimization problems (8 test problems with 10, 20 and 30 dimensions) have been tabulated along with the algorithm complexity metrics. Additionally we investigate the parametric behavior of GCO based on these test instances.**

I. Introduction

DRAWBACKS of the existing derivative-based numerical methods for solving difficult computational problems have led scientists to be on the constant lookout for natural phenomenon that serves as a source of inspiration. The evolutionary computation (EC) paradigm developed as an intermediate field in relation to biological evolution serving as the model for optimizing problem domain. This has led researchers to rely on metaheuristic algorithms founded on simulations to solve engineering optimization problems. A common factor shared by these algorithms is that they combine rules and randomness to imitate some natural phenomena. Last few decades have seen an incredible growth in the field of nature-inspired metaheuristics. Two families of algorithms that primarily constitute this field today are the Evolutionary Algorithms (EAs) [1]-[3] and the algorithms based on Swarm Intelligence (SI) [4]-[7]. While the EAs involve change in distribution of the trial solutions over a search space following genetic operations like mutation, recombination, and natural selection principles from Darwinian evolution theory, SI algorithms mostly simulate the dynamics of an intelligent group of social beings to converge to the promising regions of the search space. The two families can very well complement each other and already there has been a good trend of hybridizing the algorithms from both domains.

S. Biswas is with the Department of Electronics and Tele-communication Engineering, Jadavpur University, Kolkata, WB 700032 INDIA (e-mail: sub17was@gmail.com).

M. A. Eita is with the Department of Computer Science and Engineering, Egypt-Japan University of Science and Technology (E-JUST), Egypt (e-mail: mohammad.eita@ejust.edu.eg).

S. Das is with the Electronics and Communication Sciences Unit, Indian Statistical Institute, Kolkata, WB 700108 INDIA (e-mail: swagatam.das@isical.ac.in).

A. V. Vasilakos is with the Department of Computer Science, Kuwait University, Kuwait (e-mail: th.vasilakos@gmail.com).

However majority of works involving SI have focused on insect colonies, fish schools, bird flocks, etc. Instead of focusing on such living organisms, we focus our attention on the behavior of human beings in problem solving through counseling within a group [8], [9]. The field of counseling occupies a pivotal role in sciences like psychology and sociology, but may appear rather obscure in relation to computational optimization. We would like to point out that there have been hardly any work applying the concept of counseling to optimization except in [10], where it was proposed for the first time by Eita and Fahmy. A recently proposed multiobjective variant of GCO [11] gave promising results in multi-objective optimization problems.

In this paper a connection is established between counseling and population-based algorithms through items of analogy based on which GCO is developed. Thus, we investigate some of the basic counseling concepts and procedures in an attempt to present counseling as an appealing metaphor for computational optimization. In this conceptual framework, we identify twenty striking items of significant analogy and utilize these metaphoric items to develop what we call a Group Counseling Optimizer and test it on recently proposed test problems and thus analyze the parameters involved.

The organization of the remainder of the paper is as follows: in Section II analogy has been drawn between group counseling and the optimization based on population-based metaheuristics. In Section III, we lay down the steps for the proposed algorithm and explain them in details. In Section IV, the details and results of experimentation have been analyzed. Finally Section V concludes the paper with discussions on future work.

II. Analogy between group counseling and population-based optimization

In our day to day life, when faced with problems, we often approach other people: someone with whom we can discuss our problems, get various solutions suggested by them, experiment with those solutions and ultimately arrive at a resolution. Such instances can be seen when people are faced with relationship difficulties or wish to switch jobs or alter place of residence. The person, for instance, who is willing to change his or her job seek help of another person and may be advised by someone who has experience of job opportunities that exist in related careers [12].

Fundamentally we can think of counseling as a method of solving a problem [8]. Individual counseling involves a counselor who helps another person, called counselee, to find solution to the problem in order to resolve it [12]. But from

TABLE I: Analogy between Group Counseling and Population-based Optimization

Item	Group counseling	Population-based optimization
(1) Problem solving.	A life problem is to be solved.	A computational problem is to be solved
(2) Purpose of solution.	The most appropriate way out of a difficulty is sought.	The optimum value of an objective function is sought.
(3) Method of solution.	There is no one specific method for doing counseling. The approach varies from one situation to another.	There is no one specific method for achieving optimality. The approach is problem-dependent.
(4) Increments of solution.	Counseling is affected in increments through a number of successive counseling sessions.	Optimization is effected in increments through a number of successive computational iterations.
(5) Work-force.	Counseling is carried out by a group of participating members, a mini-society.	Optimization is carried out by a population of candidate vectors, a subspace of the real vector space.
(6) Role of participants.	Group members act as counselor at times and as counselee at other times.	Candidate vectors contribute to the improvement of a solution at times and receive improving contributions at other times.
(7) Best possible participation.	Group members use their best possible experiences in doing counseling.	Candidate vectors use their best possible positions in solution improvement.
(8) Equal chances.	All group members are eligible to give and receive help.	All candidate vectors are eligible to give and receive contributions. Contributing vectors are chosen according to a uniform distribution probability measure.
(9) Active members.	Not every group member necessarily participates in doing counseling - only verbally active members. Also, participating members vary with the variation of problems being discussed.	Not every candidate vector necessarily contributes to the improvement of vectors - only a subset of vectors chosen according to a probability measure. Also, contributing vectors vary with the variation of vector components being improved.
(10) Brainstorming.	Good ideas provided by counselors may be combined together to form a better idea.	Good contributions from candidate vectors may be added together in specific proportions to form a better contribution.
(11) From general terms to details.	Life problems are first discussed in general terms and then fine details are dealt with.	The search technique proceeds first in relatively large steps in an exploration stage, and then in small steps in an exploitation stage.
(12) Thorough investigation.	Immature solutions are to be avoided.	Local optima should not be trapped in; premature convergence is to be avoided.
(13) Problem complication.	Multidimensional life problems need a compromise between conflicting solutions.	Multi-objective function optimization needs a compromise between conflicting solutions.
(14) Inferior solution.	Counseling, in the domain of support, can only help to contain the counselee's distress.	A suboptimal solution can, in certain cases, be all what is obtainable.
(15) Dependence on other members.	A group member can solve his problem after receiving help from other members.	A candidate vector component can be improved through contributions from other vectors.
(16) Self-dependence.	Sometimes, a group member can solve his problem without receiving explicit help from other members through self-counseling, after self-discovery.	Sometimes, a candidate vector component can improve itself without receiving contributions from other vectors through self-improvement, according to a probability measure.
(17) Remaining unchanged.	Sometimes, a group member is persuaded to stay where he is; otherwise, things will worsen.	Sometimes, a candidate vector is kept unchanged; otherwise, the objective function will take on worse values.
(18) Standardization.	Counseling approaches and counselor's competencies conform to standard norms for a trustworthy counseling job.	Optimization approaches are tested using benchmark functions, and rotated benchmark functions, to demonstrate efficiency and robustness.
(19) Judgment.	The group progress is evaluated to check that the counseling job has been satisfactorily done.	The objective function is evaluated to check that convergence to the global optimum has successfully occurred.
(20) Termination.	The counseling process terminates at the end of sessions. A termination policy is followed.	The optimization process terminates at the end of iterations. A stopping criterion is used.

our daily experience, we can say that we spend majority of our time in the vicinity of more than one individual, or in groups. So there is another kind of counseling known as group counseling [9]. The unique advantage here is that group members get to know that their peers also have problems, and can advise on new ways of resolving problems by observing other group members deal with those problems over time. Contrary to individual counseling, a group enables each individual in availing assistance from others as well as in providing the same.

In a group, the members develop a sense of mutual co-operation so that they can understand and help their peers towards the betterment of life. The emerging trust in self and others allows the sharing of ideas and behaviors in a safe testing ground before applying them in relationships outside the group. Group members come to function not merely as counselees, but they practically act as counselees at certain times in the sessions to resolve their problems and as counselors to advice on such problems likewise. Unlike individual counseling, where unidirectional information and care flow occurs, it is multi-directional in a group, where each member participates in giving and receiving of advice.

Population-based optimization algorithms have been applied to a plethora of practical applications by designing an objective function that characterizes our main goal. It is mostly complex and is computed by a solution vector of parameters that can scale across varying range of dimensionality. Considering the nature and procedural phases of counseling among people, we opine that there exist many highly interesting aspects of correlation between a group counseling process and a population-based optimization process, in spite of both being in completely different settings. We bring out the resemblance through items of analogy between the two processes in Table I that well justifies the adoption of group counseling as a metaphor for population-based optimization. This is an important contribution of this article.

III. The GCO Algorithm

GCO is a population-based, derivative-free, optimization algorithm designed for detecting the global optimum. As stated earlier, our work is inspired from the problem solving ability of humans through counseling within a group. The various points of analogy between the group counseling process and a population-based optimization process, consolidated and itemized in Table I, pave way for a novel optimization approach with group counseling being the metaphor. Reference to the items of Table I will be made whenever the need arises. When we say, for instance, *cf.* Item 1 (problem solving), we refer to Item 1, entitled 'problem solving' in Table I, for comparison of a certain computational aspect with a resembling group-counseling aspect in relation to problem solving. The functioning of the computational scheme involved in the proposed algorithm is as follows.

The problem under investigation is a single-objective, multivariate, unconstrained, continuous optimization problem. Given a scalar objective function $f(\vec{X})$, the task here is to obtain a solution vector $\vec{X}^*$, which optimizes $f(\vec{X})(f : \Omega \subseteq \Re^n \rightarrow \Re)$ such that $f(\vec{X}^*) \leq f(\vec{X}) \forall \vec{X} \in \Omega$, where Ω is the search space domain.

Population-based metaheuristics rely on parallel search by instantiating a trial population of, say m individuals, each of which can be represented as

$$\vec{X}_i = [x_1, ...x_j, ...x_{n-1}, x_n]; i = 1, 2, ...m; j = 1, 2...n \quad (1)$$

where m is the population size and n is the dimensionality of the problem (function) in hand.

Each component x_j has its own bounded interval of values

$$\begin{aligned} \vec{x}_{\min} &= [x_{1,\min}, x_{2,\min}, ..., x_{n,\min}] \\ \vec{x}_{\max} &= [x_{1,\max}, x_{2,\max}, ..., x_{n,\max}] \end{aligned} \quad (2)$$

which is considered the search range of x_j and is expressed as $[x_{j,\min}, x_{j,\max}]$. This means that the length of the range of variation of x_j is

$$range_j = x_{j,\max} - x_{j,\min} \quad (3)$$

The problem at hand is to be computationally solved in the conceptual framework of the solution of a life problem through group counseling; *cf.* Item 1 (problem solving). The solution vector $\vec{X}^*$ and the corresponding global optimum $f(\vec{X}^*)$ are obtained as the most appropriate way out of a difficulty found in group counseling; *cf.* Item 2 (purpose of solution). As said, there is no one unique method for achieving optimality. The parameters of the approach taken are somewhat problem-dependent, in both optimization and group counseling; *cf.* Item 3 (method of solution).

This vector subspace corresponds to the mini-society of participating members (persons) in group counseling; *cf.* Item 5 (work-force). The population is gradually improved upon in increments through successive iterations just as counseling is incrementally affected through sequential counseling sessions; *cf.* Item 4 (increments of solution). Thus, an iteration with the m solution vectors, is analogous to an ongoing session of group counseling with m members participating in it. Member i is represented by a vector $\vec{X}_i$, which is composed of n components ($x_{i,j}$), designating what we consider the best experiences (so far) gained by the member (*selection mechanism*, detailed later). Candidate vectors contribute towards solution improvement by using their best possible positional values similar to group members using their best possible experiences in doing counseling; *cf.* Item 7 (best possible participation). Also note that candidate vectors, at times, help in solution betterment of other members (act as counselors) and at other times receive improving contributions (act as counselees) likewise; *cf.* Item 6 (role of participants). Thus every member (candidate vector) is eligible to give and receive information; *cf.* Item 8 (equal chances).

The representation of a specific member usually varies from one iteration to other [experiences vary (presumably improve) from session to session]. The components of candidate vector is updated (hopefully improve) by invoking contributions (experiences) of the corresponding components in some (not necessarily all) other vectors, or by directly altering the current value of the component itself. Hence there are two strategies at work, each with their distinct properties. The situation can be compared to what happens in group counseling, where a person – in solving his problem – asks other people for help; *cf.* Item 15 (dependence on other members) or, sometimes, he depends on himself only after self-discovery; *cf.* Item 16 (self-dependence).

Maintaining close resemblance to group counseling, where life problems are first discussed generally and then fine details are dealt with, the search for optimal solution in the GCO algorithm proceeds first in relatively large steps (exploration stage), and then in small steps (exploitation stage); *cf.* Item 11 (from general terms to details). The search is terminated when a stopping criterion is met identical to the ending of a group counseling session following a termination policy; *cf.* Item 20 (termination). Although there are many stopping criterion proposed in literature, we prefer to employ a certain maximum number of function evaluations (FEs) of the objective function which is closely followed in practical applications (offline) with a fixed FEs budget.

The proposed GCO algorithm employs four algorithmic parameters as given below:

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

The role of these parameters will become apparent as we proceed. To prevent repetition, we would like to define few notations that have been constantly maintained throughout.

m : Number of search agents (population members) used.

n : Dimensionality of the input vector to objective function $f(\vec{X}^{*})$.

$i \in \{1, 2, ...m\}$: Refers to the i^{th} member.

$j \in \{1, 2, ...n\}$: Refers to the j^{th} component of a vector.

$x_{i,j}$: Refers to the j^{th} component of member i i.e. $\vec{X}_i$.

$\vec{X}'_i$: Modified solution corresponding to $\vec{X}_i$.

$itr \in \{1, 2, ...itr\_max\}$: Refers to the present iteration number with maximum being itr_max.

$rand(L, U)$: An uniform random number in the continuous range (L, U) unless otherwise mentioned.

The algorithm starts with the initialization of the vector components $x_{i,j}$ being placed randomly in the search space. This is done in accordance with a beta probability distribution [13], [14],

$$\beta(x) = \frac{x^{a-1}\,(1-x)^{b-1}}{\mathrm{B}(a,b)}, \quad 0 < x < 1 \tag{4}$$

where the function in the denominator is a beta function defined as

$$B(a,b) = \int_0^1 t^{a-1}\,(1-t)^{b-1}\,dt \tag{5}$$

Here we have set both the shaping parameters a and b to 0.1, such that they satisfy the condition $a = b < 1$. In such a case, a symmetric U-shape of the density function is obtained. This affects the candidate solutions by, most probably, locating them near the boundaries of the search space such that the global optimum is assumed to be well within this candidate solution set. This is followed by the evaluation of the trial population.

The iterative steps are concerned with the sequential solution improvement. For each solution $\vec{X}_i$, we produce a modified solution $\vec{X}'_i$ on a component basis involving a decision making process. This is implemented by instantiating a counseling decisive coefficient such that $cdc \in rand(0, 1)$ and comparing its value to the counseling probability cp. We choose to do other-members counseling when cdc is less than or equal to cp. Otherwise self-counseling is implemented. We explain how to calculate $x'_{i,j}$ for a given xi, j in each of these strategies.

Other-members counseling $(cdc \leq cp)$

In this strategy, $\vec{X}'_i$ is regarded as the counselee and asks for counseling of c other members (counselors), chosen randomly out of the population, so that a modified component $x'_{i,j}$ is obtained; *cf.* Item 9 (active members). The value of $x'_{i,j}$ is calculated by summing *weighted values* of the corresponding j^{th} component (best experiences) of the c counselors. These are the contributions of the relevant counselors, in a brainstorming process; *cf.* Item 10 (brainstorming). The form of $x'_{i,j}$ is expressed as

$$x'_{i,j} = \sum_{q=1}^{c} w_q . x_{c_q,j} \tag{6}$$

where $w_q \in rand[0, 1]$ is the weight factor of component j in counselor q $(q \in \{1, 2....c\})$ and they add up to unity $\sum_{q=1}^{c} w_q = 1$, c_q is a random integer in the range $[1,\ m]$ with a uniform distribution. Note that, according to Eq. 6, we generate c such random numbers for each component $x_{i,j}$. The component denoted by $x_{c_q,j}$ is the value of component j of counselor q (which is member c_q). It should be evident that the set of c counselors in general varies from component to component (as j varies from 1 to n).

Self-counseling $(cdc > cp)$

Here, $x'_{i,j}$ is obtained through modification of the current component $x_{i,j}$. Thus $x'_{i,j}$ depends on the best experience $x_{i,j}$ of member i with a specific modification. In the problem statement, each component in $\vec{X}_i$ is assigned an overall bounded range. Let the length of this range for component j be denoted by $range_j$ as in Eq. 3. We choose to search for a modification value of the component in a reduced range with length $red \times range_j$, where red is the *search range reduction coefficient*. The value of $red \in [0, 1]$ is fixed for all components throughout the simulation.

Component j is updated iteratively in a maximum range $\left[-mdf\_max_j^{itr}, mdf\_max_j^{itr}\right]$ about the current component, where

$$mdf\_max_j = \frac{1}{2} \times red \times range_j. \tag{7}$$

The above equation implies that the maximum modification range is divided into two halves about the current component. This results from adding the current component to the modification range. The maximum modification value in a certain iteration, is estimated from the relation

$$mdf\_max_j^{itr} = mdf\_max_j \left(1 - \frac{itr}{itr\_max}\right)^{tr}. \tag{8}$$

The exponent tr above refers to the rate at which the search method undergoes transition from exploration to exploitation. Relation 8 is a *rule of thumb*, supported by the mathematical illustration of Fig. 1, which shows the variation of $mdf\_max_j^{itr}$ from mdf_max_j (at the very beginning of iterations) to 0 (at itr_max) for different values of tr.

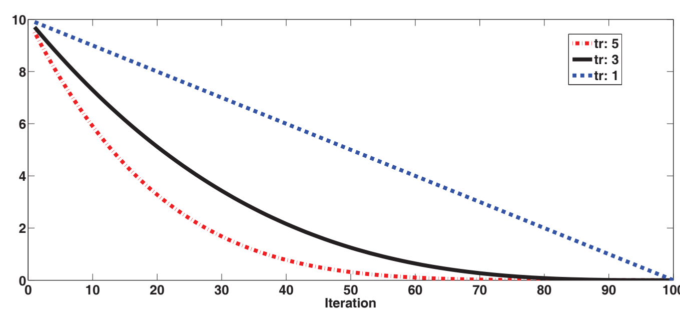


Fig. 1: Effect of transition rate tr on $mdf\_max_j^{itr}$.

It is seen that at a certain iteration, the value of $mdf\_max_j^{itr}$ decreases as tr increases. In other words, as

tr increases, the transition from exploration to exploitation occurs in a smaller number of iterations. Here the value of $mdf\_max_j^{itr}$ determines whether the algorithm performs exploration or exploitation in a certain stage. This is in accordance with the well-accepted principle that all optimization algorithms have to attain a tradeoff between exploration and exploitation so that the global optimum is eventually attained. For each component $x_{i,j}$, we generate a random number in the range $\left[-mdf\_max_j^{itr}, mdf\_max_j^{itr}\right]$ and this is added to $x_{i,j}$ to obtain the modified value of the form

$$x'_{i,j} = x_{i,j} + rand\left(-mdf\_max_j^{iter}, mdf\_max_j^{iter}\right). \quad (9)$$

For solutions exceeding the bounds, they are reset using a repair operator as

$$\begin{array}{l} if \ \ x'_{i,j} > x_{j,\max} \\ \qquad then \ \ x'_{i,j} = x_{i,j} + rand(0, x_{j,\max} - x_{i,j}) \\ if \ \ x'_{i,j} < x_{j,\min} \\ \qquad then \ \ x'_{i,j} = x_{i,j} + rand(x_{j,\min} - x_{i,j}, 0) \end{array} \quad (10)$$

Following the generation of $\vec{X}'_i$, its fitness is evaluated as $f(\vec{X}'_i)$; *cf.* Item 19 (judgment). A one to one comparison takes place between $\vec{X}'_i$ and $\vec{X}_i$. If $f(\vec{X}'_i)$ is better (less for minimization or greater for maximization) than $f(\vec{X}_i)$, then $\vec{X}'_i$ replaces $\vec{X}_i$; otherwise, $\vec{X}_i$ is kept unchanged for subsequent improvement; *cf.* Item 17 (remaining unchanged). The steps are repeated till a *termination criterion* is met. Finally, out of the m solutions, the best solution is taken as the optimum solution $\vec{X}^*$. The outline of the GCO algorithm is presented in the form of pseudocode in Algorithm 1.

Algorithm 1: Group Counseling Optimizer

input : i) Control parameters: c, cp, red, tr.
ii) Objective function $f(\vec{X}_i)$.
iii) Dimensionality n and population size m.
output: Solution $\vec{X}^*$ and functional value $f(\vec{X}^*)$.

begin
  Initialize trial population $\mathbf{X}$ using Eq. (4) and (5). ;
  Evaluate the candidate solutions using $f(\vec{X})$;
  Compute itr_max and set $itr \longleftarrow 0$;
  while *termination criteria is not met* **do**
    Update $mdf\_max_j^{itr}$ $\forall j \in 1, 2, ...n$;
    Set $\mathbf{X'} \longleftarrow \emptyset$;
    for [1]$x \in \mathbf{X}$ **do**
      Instantiate cdc. [2]$cdc \longleftarrow rand(0, 1)$;
      if $cdc \leq cp$ **then**
        Generate x' using other-members counseling.
      else
        Generate x' using self counseling;
        Apply repair operator using Eq. (10);
      end
      $\mathbf{X'} \longleftarrow \mathbf{X'} \cup x'$;
      Evaluate x';
    end
    for $x \in \mathbf{X}$ **do**
      if $f(x') \leq f(x)$ **then**
        Replace x by x' and $f(x)$ by $f(x')$;
      end
    end
    $itr \longleftarrow itr + 1$;
  end
end

[1] x here actually represents $\vec{X}_i$.

[2] This action is instantiated n times for each component.

IV. Experimentation

The performance of GCO is evaluated on the benchmark set proposed in the Technical Report titled *Problem Definitions and Evaluation Criteria for Computational Expensive Optimization* [15]. The test suite, *cf.* Item 18 (standardization), is composed of 24 benchmark functions (8 test problems spanning across 10, 20 and 30 dimensions) and are treated as black-box optimization problems.

A. Simulation Environment

The experimentation has been performed under the following system configuration:

- Windows 7 Home Premium SP1
- Processor: Intel(R) Core(TM) i5-2450 @ 2.5 GHz
- RAM: 4.00 GB (usable 3.90 GB)
- Language: MATLAB R2010a

B. Parametric settings

The issue of control parameters has been an area of active research in optimization algorithms and is usually problem dependent. Although different problems require varying parametric setting, it is preferable for an algorithm to use a common parametric setting. Our aim here is not to fine tune the parameters for each problem of varying dimensionality but to select a general setting. As stated above, GCO algorithm makes use of 4 control parameters : number of counselors (c), counseling probability (cp), search range reduction coefficient (red) and transition rate (tr).

We would like to state that, other member counseling permits exchange of information among the members and leads to diversity determined by the value of number of counselors (c). A high value of c and cp means a high probability of information sharing among a greater group leading to slow convergence and is preferable for multimodal landscapes. Whereas low value of c and cp ensures local search and enhances convergence rate (unimodal landscape preferred). Likewise population size (m) is another factor. Although a high value of m ensures good exploration but considering the serious limitation of FEs here, we need a tradeoff between exploration and exploitation (even though tr also controls this rate through nonlinear variation).

Based on these limitations and considerations, we set the control parameters to the following setting.

Population size (m) : 25
Number of counsellors (c) : 2
Counselling Probability (cp) : 0.7
Reduction co-efficient (red) : 0.4
Transition Rate (tr) : 2.

C. Performance

The experimental setting used here is according to the guidelines established in [15]. The FEs budget (MaxFES) has been fixed at $50 \times n$ where n is the problem dimensionality and take values 10, 20 and 30. The error value has been recorded for 20 independent sample runs. The runs are terminated on expending MaxFEs and the final error value obtained is presented for 8 test problems categorized as $10D$,

$20D$ and $30D$ functions and enlisted in Table II, III and IV respectively. Additionally, we have plotted the best-of-the-run error values in Fig. 2 and showed the variation in obtained error with the increasing iterations.

From the results obtained, we can see that the performance of GCO mostly deteriorates with increasing dimensionality. This is quite evident from the exponential rise in the search space which makes it difficult for any Cartesian co-ordinate based search process to detect optimum efficiently. But this is not the case entirely; for *Shifted Step function* ($F10-F12$) and *Shifted Griewank's function* ($F16-F18$), the mean error value remains constant approximately. The performance fluctuates to some extent in *Shifted and Rotated Rosenbrock's functions* which may be due to the presence of narrow valley.

TABLE II: Result table for 10D

Function	Best	Worst	Median	Mean	Std
$F1$	3.2252e+0	2.9585e+1	1.0863e+1	1.2255e+1	6.3649e+0
$F4$	8.4590e+0	2.2170e+2	2.6284e+1	4.1351e+1	4.6139e+1
$F7$	1.5615e+1	2.0940e+2	8.4916e+1	8.8506e+1	5.5348e+1
$F10$	3.0000e+0	2.7000e+1	7.0000e+0	1.0095e+1	6.9347e+0
$F13$	3.9171e+0	9.9356e+0	5.9863e+0	6.3547e+0	1.7083e+0
$F16$	1.2409e+0	4.5086e+0	1.8906e+0	2.1126e+0	6.7736e-1
$F19$	4.4196e+1	1.7999e+2	8.6354e+1	9.2782e+1	3.2193e+1
$F22$	3.3953e+0	6.4780e+1	4.7905e+1	4.7361e+1	1.0320e+1

TABLE III: Result table for 20D

Function	Best	Worst	Median	Mean	Std
$F2$	3.5974e+1	2.1703e+1	1.0735e+1	1.1871e+1	5.8804e+0
$F5$	7.7910e+1	1.7877e+2	6.8882e+1	9.3442e+1	4.7450e+1
$F8$	3.0000e+0	2.6173e+2	1.2994e+2	1.4443e+2	4.7573e+1
$F11$	3.3285e+0	1.4000e+1	9.0000e+0	8.4762e+0	3.0103e+0
$F14$	1.2826e+0	8.9372e+0	4.3345e+0	4.5777e+0	1.1552e+0
$F17$	4.9111e+1	2.8568e+0	1.7657e+0	1.8898e+0	4.9443e-1
$F20$	3.0956e+1	1.7232e+2	1.0844e+2	1.1264e+2	2.6665e+1
$F23$	3.3953e+0	1.2478e+2	8.5884e+1	7.9049e+1	2.2262e+1

TABLE IV: Result table for 30D

Function	Best	Worst	Median	Mean	Std
$F3$	3.4771e+0	1.8021e+1	7.2885e+0	8.2618e+0	3.7792e+0
$F6$	6.6760e+1	3.0964e+2	1.6566e+2	1.7135e+2	5.6276e+1
$F9$	3.0804e+2	1.6252e+3	5.9978e+2	7.6153e+2	3.8806e+2
$F12$	2.0000e+0	1.6000e+1	6.0000e+0	7.7619e+0	3.8975e+0
$F15$	2.9418e+0	5.1773e+0	4.0533e+0	4.0955e+0	5.6399e-1
$F18$	1.2928e+0	2.7759e+0	1.8292e+0	1.8695e+0	3.9111e-1
$F21$	1.7427e+2	4.7116e+2	2.4427e+2	2.4857e+2	6.5074e+1
$F24$	1.0438e+2	2.4569e+2	1.8084e+2	1.8128e+2	3.5581e+1

But considering the small value of m, the performance of GCO may be ideally suited to online problems where time is a serious concern rather then the accuracy of solutions obtained. The time complexity analysis is discussed next.

D. Complexity analysis

The algorithmic complexity of GCO has been computed as per [15] and the data has been tabulated in Table V. The value reported for $10D$ functions are roughly 0.5 and it

TABLE V: Computational Complexity

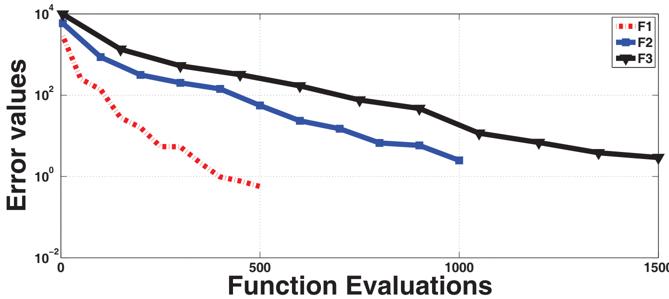
Function	$\widehat{T1}/T0$
$F1$	0.452
$F2$	0.954
$F3$	1.52
$F4$	0.457
$F5$	0.988
$F6$	1.47
$F7$	0.516
$F8$	1.04
$F9$	1.53
$F10$	0.471
$F11$	0.976
$F12$	1.55
$F13$	0.477
$F14$	1
$F15$	1.54
$F16$	0.485
$F17$	1.01
$F18$	1.58
$F19$	0.492
$F20$	1.01
$F21$	1.61
$F22$	0.481
$F23$	1.03
$F24$	1.58

rises to around 1.0 and 1.5 when the problem dimensionality n scales up by 2 and 3 respectively. There is a strong uniformity in the values reported which represents a roughly linear dependence on n. This is indicative of the fact that no serious computational overhead is incurred during internal evaluations as a result of rising n.

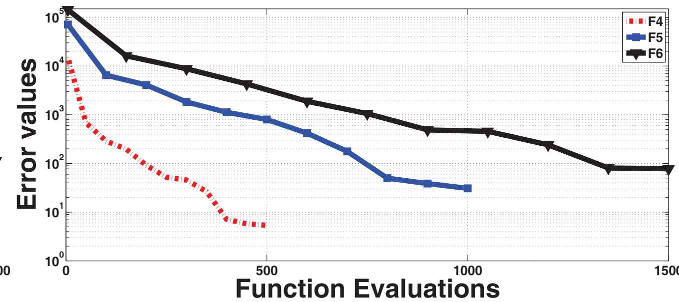
E. Discussions

To understand the effect of control parameters on the performance of GCO, we selected an intermediate n equal to 20 and observed the performance by varying a single control parameter while fixing the others. Note that population size m has not been studied here. We are more interested in red, tr, cp and c. For analysis, we used $red \in [0.1, 0.2, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.95]$,
$tr \in [2, 5, 7, 10, 15, 20, 25, 30]$,
$cp \in [0.01, 0.05, 0.1, 0.3, 0.5, 0.7, 0.9]$,
and $c \in [2, 3, 4]$.

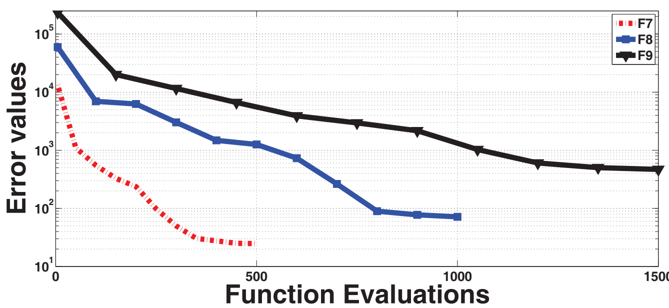
From the experimentation, we opine that the value of cp is crucial to the performance of the algorithm. This stems from the fact that the value of cdc is checked with cp before the effect of c (in other-members counseling), or of red and tr (in self-counseling) is felt. It acts as a control mechanism. It also affects the computational complexity. A high value of cp 1 can be approximated as $n.m.c$ calls to the random number generator in a single iteration. Although this effect is not felt in the problems considered here, it can become a significant overload when $n >> 1$. Then comes the mutual role of red and tr in deciding the search range at disposal and also the transition in phase. The value of tr 2 is ideal since the curve depicted in Fig. 1 approaches a linear shape representing a controlled transition. This method is preferable since the $a-$priori knowledge about the nature of landscape (modality) remains unavailable in most practical problems.



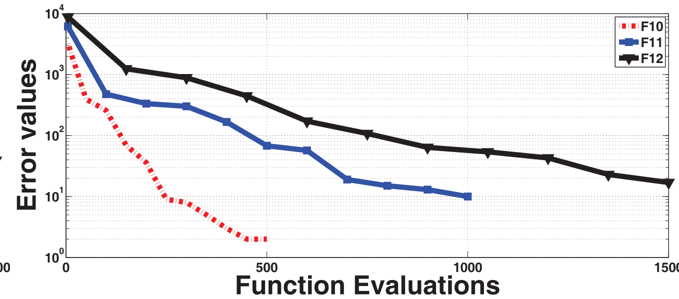
(a) Shifted Sphere function $F_1 - F_3$



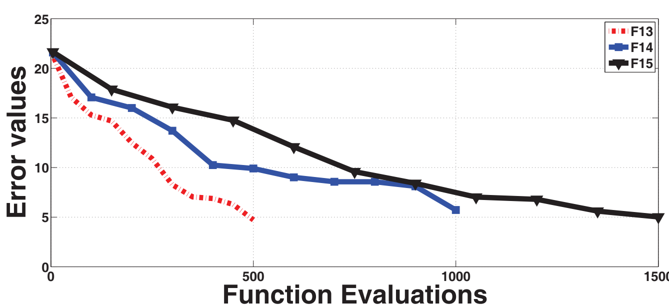
(b) Shifted Ellipsoid function $F_4 - F_6$



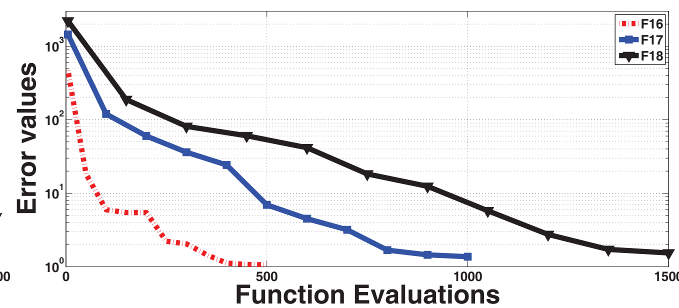
(c) Shifted and Rotated Ellipsoid function $F_7 - F_9$



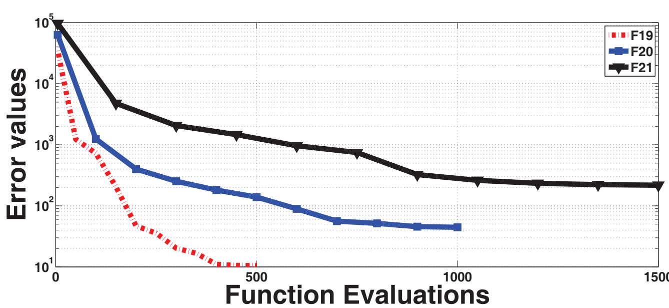
(d) Shifted Step function $F_{10} - F_{12}$



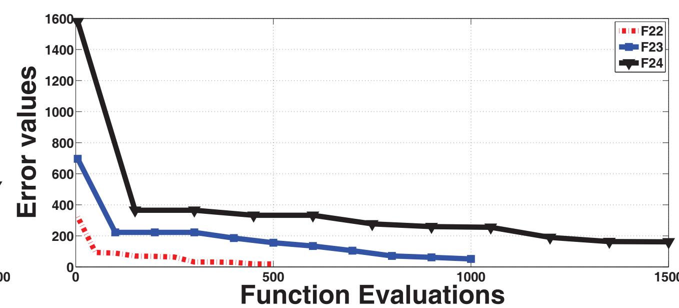
(e) Shifted Ackley's function $F_{13} - F_{15}$



(f) Shifted Griewank's function $F_{16} - F_{18}$



(g) Shifted Rotated Rosenbrock's function $F_{19} - F_{21}$



(h) Shifted Rotated Rastrigin's function $F_{22} - F_{24}$

Fig. 2: Convergence graph $F_1 \sim F_{24}$

It would not be wise to stress too much on either exploration or exploitation. The value of tr around 2 seems optimal.

Although we arrived at the chosen settings through a bit of both intuition and experimental verification, but this settings is bound to show poor performance in another benchmark set. This is in accordance with the No Free Lunch (NFL) theorem [16] and is the primary motivation behind a great deal of research. Recently there has been efforts to study the effect of GCO in the light of an extended benchmark set [17].

V. Conclusions

This research article dealt with a population-based meta-heuristics called Group Counseling Optimizer. The contribution of this paper is therefore twofold. First, we investigate some of the basic counseling concepts and procedures and identify twenty striking items of significant analogy. Second, we develop the GCO algorithm from these items and the proposed algorithm is evaluated on the test-suite for CEC–2014 benchmark set for expensive optimization. This was done through reporting of error values obtained for problems of varying dimensions. Alongside, experimentally calculated algorithmic complexity measurements have been reported as well.

During the process of experimentation we noticed the need for tuning of parameters in the basic GCO framework. The adaptation of parameters and parameter ensembles being highly promising areas of research, can as well be extended to GCO. We also noticed that in spite of a very small population size, GCO was able to maintain its performance across changing dimensionality. Study of this area along with real-life applications can also be undertaken by the researchers in near future.

References

[1] L. J. Fogel, A. J. Owens and M. J. Walsh. *Artificial Intelligence through Simulated Evolution*, John Wiley, 2006.

[2] H.-G. Beyer and H.-P. Schwefel. Evolution Strategies: A Comprehensive Introduction. In *Natural Computing*, vol. 1(1), pp. 3-52, 2002.

[3] R. Storn and K. V. Price, Differential evolution: A simple and efficient adaptive scheme for global optimization over continuous spaces, ICSI, USA, Tech. Rep. TR-95-012, 1995 [Online]. Available: http://icsi.berkeley.edu/ storn/litera.html.

[4] M. Clerc, *Particle Swarm Optimization*, ISTE Publications, 2008.

[5] M. Dorigo and T. Sttzle, *Ant Colony Optimization*, MIT Press, Cambridge, MA, 2004.

[6] E. Bonabeau, M. Dorigo and G. Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. New York, NY: Oxford University Press, Santa Fe Institute Studies in the Sciences of Complexity 1999.

[7] W. Banzhaf, G. Beslon, S. Christensen, J. A. Foster, F. Kps, V. Lefort, J. F. Miller, M. Radman and J. J. Ramsden. From artificial evolution to computational evolution: a research agenda. *Nature Reviews. Genetics*, Vol. 7, pp. 729-735, 2001.

[8] D. N. Dixon and J. A. Glover, *Counseling: A problem-solving approach*. Wiley, New York 1984.

[9] R. C. Berg, G. L. Landreth and K. A. Fall, *Group Counseling: Concepts and Procedures*, 4th edition, Routledge, Taylor and Francis, 2006.

[10] M. A. Eita and M. M. Fahmy, "Group Counseling Optimization: A Novel Approach", *29th SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence* (AI-2009), pp. 195-208, Cambridge, England, UK, December 15-17, 2009.

[11] H. Ali and F. A. Khan, "Group Counseling Optimization for Multi-objective Functions", in *IEEE Cong. Evol. Comput.* (CEC 2013), pp. 705 - 712, Cancun, Mexico, June 20-23, 2013.

[12] P. Burnard, *Practical Counseling and Helping*. Routledge, Taylor and Francis, 2002.

[13] R. C. H. Cheng, Generating Beta Variates with Non-integral Shape Parameters, *Communications ACM*, Vol. 21, No. 4, pp. 317-322, April 1978.

[14] A. K. Gupta, S. Nadarajah, *Handbook of Beta Distribution and its Applications*, Marcel Dekker, 2004.

[15] B. Liu, Q. Chen, Q. Zhang, J. J. Liang, P. N. Suganthan and B. Y. Qu, "Problem Definitions and Evaluation Criteria for Computational Expensive Optimization", in Tech. Report, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Tech. Report, Nanyang Technological University, Singapore, December 2013.

[16] D. H. Wolpret and W. H. Macready, "No free lunch theorems for optimization", *IEEE Transactions on Evolutionary Computation*, Vol. 1, Isuue 1, pp. 67-92, April, 1997.

[17] M. A. Eita and M. M. Fahmy, "Group Counseling Optimization", *Applied Soft Computing*, In Press.