Group Counseling Optimization: A Novel Approach

M.A.Eita[1] and M.M.Fahmy[2]

Abstract A new population-based search algorithm, which we call Group Counseling Optimizer (GCO), is presented. It mimics the group counseling behavior of humans in solving their problems. The algorithm is tested using seven known benchmark functions: Sphere, Rosenbrock, Griewank, Rastrigin, Ackley, Weierstrass, and Schwefel functions. A comparison is made with the recently published comprehensive learning particle swarm optimizer (CLPSO). The results demonstrate the efficiency and robustness of the proposed algorithm.

1 Introduction

One of the most fundamental principles in our world is the search for an optimal situation. Many scientific, engineering, and economic problems involve the optimization of particular objective functions. These problems include examples like minimizing the losses in a power grid by finding the optimal configuration of the components, or training a neural network to recognize images of people's faces. Numerous optimization algorithms have been proposed, with varying degrees of success.

Over the decades, traditional optimization techniques, such as linear programming and steepest-decent methods, are used. Because of certain drawbacks of these techniques and the increasing complexity of real-world optimization problems, there is an urgent need for better optimization algorithms. Many heuristic algorithms are therefore developed to solve various optimization problems. These algorithms combine rules and randomness to mimic natural phenomena. Examples are: the Genetic Algorithm [9,11], Evolution Strategies [18] such as Differential Evolution [22], Ant Colony Optimization [4], Particle

1 M.A.Eita, Faculty of Engineering, University of Tanta, EGYPT
e-mail: eta1232002@yahoo.com

2 M.M.Fahmy, Faculty of Engineering, University of Tanta, EGYPT
e-mail: mfn_288@hotmail.com

M. Bramer et al. (eds.), *Research and Development in Intelligent Systems XXVI*,
DOI 10.1007/978-1-84882-983-1_14,

Swarm Optimization [5,6,12], Bees Colony Optimization [17], Memetic Algorithms [16], and Cultural Algorithms [20].

The Genetic Algorithm mimics natural selection and genetic recombination. The algorithm works by choosing solutions from the current population and then applying genetic operators – referred to as mutation and crossover – to create a new population. Crossover is the partial swap between two parent strings to produce two offspring strings. Mutation is the occasional random inversion of bit values, generating non-recursive offspring.

The Evolution strategies are heuristics-based optimization techniques exploiting the ideas of adaptation and evolution. The essential idea behind Differential Evolution is the way the (ternary) recombination operator 'deRecombination' is defined for creating new solution candidates. The difference $x1-x2$ of two vectors $x1$ and $x2$ is weighted with a weighted and added to a third vector $x3$ in the population.

The Ant Colony Optimizer is based on the metaphor of ants seeking for food. It imitates the behavior of ants in laying a trail of pheromone to find the shortest path from the food source to their nest. Each ant that finds the food will excrete some pheromone on the path. By time, the pheromone density of the path will increase and more and more ants will follow it to the food and back to the nest. The higher the pheromone density, the more likely will an ant stay on a trail. The probability that a passing stray ant will follow this trail depends on the quantity of pheromone laid.

The Particle Swarm Optimizer emulates a biological social system like a flock of birds or a school of fish. When a swarm looks for food, its individuals will spread in the environment and when one of them finds food, it announces this to its neighbors. These neighbors can then approach the source of food, too.

The Bees Colony Optimization is inspired from the natural foraging behavior of honeybees to find the optimal solution in nectar collection [17]. The algorithm performs a kind of neighborhood search combined with random search.

The Memetic Algorithms are population-based approaches for heuristic search in optimization problems. They have been shown to be orders of magnitude faster than traditional genetic algorithms for some problem domains. Basically, they combine local search heuristics with crossover operators. For this reason, some researchers have viewed them as *hybrid* genetic algorithms.

The Cultural Algorithms are a branch of evolutionary computation where there is a knowledge component that is called the belief space in addition to the population component. In this sense, cultural algorithms can be seen as an extension to conventional genetic algorithms.

In this paper, we propose a new optimization approach that *emulates the human behavior in problem solving through counseling within a group*. The approach is called a *Group Counseling Optimizer* (GCO). The iterations involved in the solution algorithm are visualized as counseling sessions. Candidate solutions are progressively improved by either counseling with other members in the group or by self-counseling. This line of thinking, we believe, holds much promise since the human's behavior has, or should have, the highest quality when compared with the behavior of other (lower-class) creatures.

The remainder of the paper is organized as follows: Sect. 2 introduces counseling among humans as a problem-solving approach. In Sect. 3, the proposed algorithm, based on group counseling, is explained. In Sect. 4, the results of the experiments conducted on seven benchmark functions are given. The conclusions are finally discussed in Sect. 5.

2 Counseling as a Problem-Solving Approach

People with problems often seek out another person as a sounding board: someone with whom they can talk over their problems, experiment with various solutions and finally reach some resolution. Examples of this approach are seen when people have relationship difficulties or want to change jobs or places of residence. The person, for instance, who wants to change his or her job may be advised by another person who has experience of job opportunities that exist in related careers [2]. Hence, people start to seek help when they should make a decision or solve a problem.

Counseling can be thought of as a process of problem solving [3]. *Individual counseling* is an activity in which one person is helping (counselor) and one is receiving help (counselee) and in which the emphasis of that help is on enabling the other person to find solutions to problems [2].

However, individuals function most of their lives within groups. So, instead of the individual counseling there is another kind of counseling called *group counseling* that offers the unique advantages of providing group members with the opportunity to discover that their peers also have problems and to learn new ways of resolving problems by observing other members in the group deal with those problems. Unlike individual counseling relationships, a group provides each individual the opportunity to give as well as to receive help.

In the group, the members can discover that they are capable of understanding, accepting, and helping their peers, and that they can contribute to another person's life. Thus, members gradually begin to understand and accept themselves. The emerging trust in self and others facilitates the sharing of ideas and behaviors in a safe testing ground before applying those ideas and behaviors in relationships outside the group.

Group members come to function not merely as counselees, but they practically behave as counselees at certain times in the sessions and as counselors at other times. Unlike individual counseling, where information and care flow in a single direction, in a group, the flow of information and care is multi-directional, where each member participates in the giving and receiving of advice.

A list of rules exists for the group members [1]:

1. Let others know what your ideas are. What every member has to say is important. Sharing your thoughts and reactions with the group will stimulate other members and will help them to share what they are thinking.
2. Ask your questions. If you have a question or you want to know more about something, do not hesitate to ask.
3. Do not do all the talking. Others want to participate also, and they cannot if you take too long to express your ideas.
4. Help other members to participate. If someone looks as though he or she wants to say something but has not, encourage that person to do so.
5. Listen carefully to other members. Give a chance to the ideas of other persons, and try to understand what he or she is saying. Listen to other members in the way you would want them to listen to you.
6. Group members are here to help. Problems can be solved by working cooperatively together. In the process of helping others, you can help yourself. The information you have can be helpful to others. Suggesting alternatives or causes can help other members to make better decisions. This suggesting process of many alternatives is called *brainstorming*. In a brainstorming process, good ideas may be combined to form a new better idea [13].
7. Be willing to accept other viewpoints. Do not insist that you are right and everyone else is wrong. The other person just might be thinking the same thing. Try to help other members to understand rather than trying to *make* them understand.
8. Keep up with the discussion. If the discussion is confusing to you, say so.
9. In this group, to talk about your feelings and reactions is admissible.

In addition, individuals learn best when they become involved as participating and contributing members to the group. Each member needs to actively share in the group's decision making. Undoubtedly, group members will contribute to solving of the problem only by the best of their experiences.

Counseling can help some but not all people. Also, we should not assume that counseling can help in every situation [2].

In case when people depend on themselves in solving their problems, they exploit the best of their past experiences with some kind of *modification* seeking for a better, satisfactory solution.

The *group counseling optimization* approach developed here mimics the main ideas of human group counseling behavior illuminated above, without making use of all its details. This metaphor deserves utmost attention. The present paper, we hope, opens an avenue.

3 The proposed GCO Algorithm

The problem at hand is a single-objective, unconstrained, continuous optimization problem. Given a scalar function *f(X)*, where *X* denotes a set of *D* parameters x_d, *d=1,2,...D*, it is required to optimize *f(X)* through appropriate values of x_d; optimization means either minimization or maximization.

The main idea of the proposed approach is as follows. Like other heuristics-based approaches, we begin with a certain number, *m*, of initial candidate solution vectors X^i in the *D*-dimensional search space, $X^i = (x_1^{\,i}, x_2^{\,i}, \ldots x_D^{\,i}); i = 1,2,\ldots, m$. These solution vectors are then improved through successive iterations. We regard such *m* solution vectors in (each of) the different iterations as *m* members (persons) in a group. Member *i* is represented by the vector X^i, which in turn contains *D* components x_d^i, *d=1,2,...D,* designating what we consider the best experiences of the member. Note that the representation of a specific member generally varies from iteration to iteration. A new value of each component in a vector is produced by invoking the current values (experiences) of corresponding components in other vectors or by modifying the current value of the component itself. These are two strategies, each having distinct behavior properties as will be soon discussed. The situation is interestingly analogous to what happens in group counseling, where a person - in solving a problem - asks other people for help or, sometimes, depends on himself only.

Each iteration is visualized as a group counseling session, with m members. We obtain m candidate solutions X^i from each session (except the last one) which are improved successively in subsequent sessions. The final session is the *decision-making* session. It receives the eventual m candidate solutions and compares them with each other so that the best solution is determined, the solution X^* that optimizes the objective function *f(X)*.

The proposed group counseling optimizer (GCO) is a search algorithm inspired by the group counseling approach to solve problems. The algorithm requires a number of parameters to be set, namely: number of group members representing the population size (*m*), number of group members used as counselors (*c*), counseling probability (*cp*), maximum value of modification (*mdf_max*), and transition rate from exploration to exploitation (*tr*). The significance of these parameters will become apparent as we proceed.

The algorithm is executed through the following steps:

Step 1

The algorithm starts with m initial candidate solutions $X^i = (x_1^i, x_2^i, \ldots x_D^i); i = 1,2,\ldots,m$, being placed randomly in the search space. We choose to locate the values of x_d^i in accordance with a beta distribution. As Fig. 1 shows, the beta probability density function *g(x)*, with its two parameters *a* and *b* being equal and less than unity, is of a symmetric U-shaped form. This implies that, most probably, the candidate solutions lie near the boundaries of the search space and that the global optimum is within this candidate solution set. For details of the beta distribution, see [8,10].

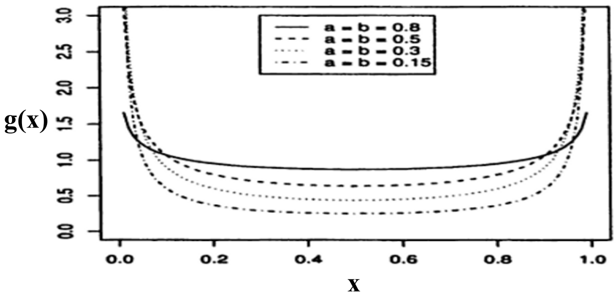

Fig. 1 U-shaped, symmetric beta probability density function

Step 2

The solution vectors X^i are substituted respectively into the objective function *f(X)*, yielding m values for $f(X^i)$, called fitness values.

Step 3

This is the first iterative step. For each solution X^i, we produce an alternative solution $X'^i = (x_1'^i, x_2'^i, \ldots, x_D'^i)$. The production process is carried out *component-wise*. Each component x'^i_d is obtained through one of two counseling strategies; namely,

(a) *Other-members counseling*

(b) *Self-counseling*

For each component x'^i_d, we start with generation of a random number, in the range [0,1], according to a uniform distribution. This number is here called a *counseling decisive coefficient* (*cdc*). If *cdc* is less than or equal to *cp* (set in the range [0,1]), we do other-members counseling; otherwise, we do self-counseling. In the following, we explain how to calculate x'^i_d in each of these strategies.

Step 3a: Other-members counseling (*cdc* ≤ *cp*)

In this strategy, member *i* (X^i) is regarded as a counselee. It counsels *c* other members (counselors), chosen randomly out of the population, so that another (hopefully better) alternative component x'^i_d is obtained. The value of x'^i_d is calculated by summing *weighted* values of the corresponding components (best experiences) of the *c* counselors. These are the contributions of the relevant counselors, in a *brainstorming* process.

The weight, denoted by ω_k, of component *d* in counselor *k* (*k*=1,2,….*c*) is a random number in the range [0, 1] with a uniform distribution,

$$\omega_k = rand\ (0,1) \tag{1}$$

Bear in mind that counselor k is some member i.

The c weights should sum to unity,

$$\sum_{k=1}^{c} \omega_k = 1 \tag{2}$$

The form of x''^i_d is expressed as

$$x'^i_d = \sum_{k=1}^{c} \omega_k * x_d^{rand\_int_k(1,m)} \tag{3}$$

where the superscript *rand_int(*1*,m)* is an *integer* random number in the range [1*,m*] with a uniform distribution, and $x_d^{rand\_int_k(1,m)}$ is the value of component d of counselor k. Note particularly that set of c counselors in general varies from component to component (as d varies from 1 to D). It should also be clear that ω_k and *rand_int(1,m)* are both dependent on the values of i and d; these symbols are not superimposed on Eq.(3) for notational simplicity.

<u>Step 3b: Self-counseling (*cdc* > *cp*)</u>

In this strategy, an alternative component x''^i_d is obtained through modification of the current component x^i_d. This situation may be interpreted as follows. Member i, being involved in the counseling group, discovers that it is capable of suggesting a new component x''^i_d depending on its own best experience x^i_d with some specific modification. The value x^i_d is modified by adding a term $rand_d^{\,i}(-mdf, mdf)$. Here, *mdf* and *–mdf* are the greatest positive and negative values of modification, respectively. That is, the value of x^i_d will change in the range [-*mdf,mdf*]. The value of *mdf* plays a central role in whether the optimization algorithm performs 'exploration' or 'exploitation'. An equation which can be used to estimate *mdf* is

$$mdf = mdf\_\max * \left(1 - \frac{itr}{itr\_\max}\right)^{tr} \tag{4}$$

where *mdf_max*, as stated previously, is a set value, *itr* is the iteration number, and *itr_max* is the total number of iterations. The exponent *tr* in Eq.(4) refers to a transition rate at which the search method changes from exploration to exploitation. Consequently, the form of x''^i_d is given by

$$x'^i_d = x^i_d + rand_d^{\,i}(-mdf, mdf) \tag{5}$$

As a further illustration, Fig. 2 shows the variation of *mdf* from *mdf_max* (at the very beginning of iterations) to zero (at *itr_max*) for different values of *tr*. It is evident that at a certain iteration, the value of *mdf* decreases as *tr* increases. In other words, as *tr* increases, exploration tends to exploitation in a smaller number of iterations. It is well known that all optimization algorithms have to compromise between exploration to exploitation so that the global optimum is eventually attained.

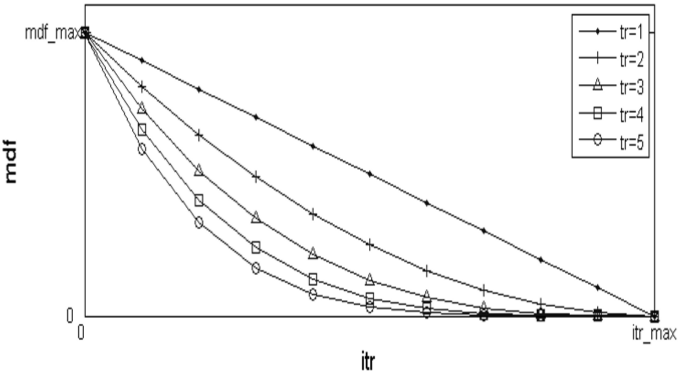

Fig. 2 Effect of transition rate (*tr)* on modification

The result of Step 3 is a set of m solution vectors X'^{i}. We emphasize the fact that, in general, some of the components x'^{i}_{d} of X'^{i} are produced by other-members counseling while the remaining components (of the same vector) are produced by self-counseling.

Step 4

Step 2 is repeated for X'^{i} (instead of X^{i}) and the fitness value, $f(X'^{i})$, is evaluated. If $f(X'^{i})$ is better than $f(X^{i})$, then X'^{i} replaces X^{i} ($X^{i} \leftarrow X'^{i}$); otherwise, X'^{i} is ignored and X^{i} remains for possible subsequent improvement.

Repetition Steps (iterations)

Step 3 and 4 are repeated until a *stopping criterion* is met.

Final Step

This is a decision-making step. The *m* solutions, resulting from the last repetition step, are compared with each other based on the fitness values of the objective function. The best solution is taken as the optimum solution X^* (with acceptable error).

4 Experiments

The proposed GCO algorithm is tested in *minimization* problems using seven benchmark functions: two unimodal functions (Sphere and Rosenbrock) and five multimodal functions (Griewank, Rastrigin, Ackley, Weierstrass, and Schwefel) [7,14]. Also, it is compared with the comprehensive learning particle swarm optimizer (CLPSO) developed by Liang *et al.* [15].

4.1 Test Functions

The definitions of the benchmark functions used for testing are as follows:

1) Sphere function

$$f_1(x)=\sum_{i=1}^{D}x_i^2$$

2) Rosenbrock function

$$f_2(x)=\sum_{i=1}^{D-1}(100(x_i^2-x_{i+1})^2+(x_i-1)^2)$$

3) Ackley function

$$f_3(x)=20+e-20\cdot e^{-0.2\cdot\sqrt{\frac{1}{D}\sum_{i=1}^{D}x_i^2}}-e^{\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)}$$

4) Griewanks function

$$f_4(x)=\sum_{i=1}^{D}\frac{x_i^2}{4000}-\prod_{i-1}^{D}\cos(\frac{x_i}{\sqrt{i}})+1$$

5) Weierstrass function

$$f_5(x)=\sum_{i=1}^{D}(\sum_{k=0}^{k\max}[a^k\cos(2\pi b^k(x_i+0.5))])-D\sum_{k=0}^{k\max}[a^k\cos(2\pi b^k.0.5)],$$

$$a=0.5,\quad b=3,\quad k\max=20$$

6) Rastrigin function

$$f_6(x) = \sum_{i=1}^{D} (x_i^2 - 10 \cdot \cos(2\pi x_i) + 10)$$

7) Schwefel function

$$f_7(x) = 418.9829 \cdot D + \sum_{i=1}^{D} x_i \cdot \sin(\sqrt{|x_i|})$$

All the above functions are tested for dimension D=30. The global optima X^*, fitness values $f(X^*)$, and search ranges are given in Table 1.

Table 1. Global Optima and Search Ranges of Test Functions

f	X^*	$f(X^*)$	**Search Range**
f_1	[0,0,……..,0]	0	$[-100,100]^D$
f_2	[1,1,……..,1]	0	$[-2.048,2.048]^D$
f_3	[0,0,……..,0]	0	$[-30,30]^D$
f_4	[0,0,……..,0]	0	$[-600,600]^D$
f_5	[0,0,……..,0]	0	$[-0.5,0.5]^D$
f_6	[0,0,……..,0]	0	$[-5.12,5.12]^D$
f_7	[420,420,……..,420]	0	$[-500,500]^D$

4.2 Parameter Settings

In conducting the experiments, we use the following parameter values for the GCO algorithm: number of group members m=40; parameters of beta distribution: a=b=0.25; number of counselors c=2; maximum number of fitness evaluations FEs=200,000. All experiments are run 30 times. The three parameters *cp, tr*, and *mdf_max* are set differently for the test functions, as indicated in Table 2. The parameters of the CLPSO algorithm are taken as specified in [15].

Table 2. Parameter settings for test functions

f	*cp*	*tr*	*mdf_max*
f_1	0.01	30.0	10.0
f_2	0.008	1.0	0.01
f_3	0.12	20.0	3.0
f_4	0.025	15.0	50.0
f_5	0.07	30.0	0.1
f_6	0.018	15.0	1.0
f_7	0.04	18.0	100.0

4.3 Results

Table 3 gives the mean values and standard deviations of the 30 runs of the test functions for the GCO algorithm, together with the CLPSO algorithm. Figure 3 illustrates the convergence characteristics, for the two algorithms, through the variation of the best function value as a function of FEs. The comparison demonstrates the success and effectiveness of the proposed GCO algorithm. Specifically, in the experiments conducted, the GCO outperforms the CLPSO for the first six benchmark functions and is well comparable to it for the seventh function (Schwefel).

Table 3. Mean and standard deviation of GCO and CLPSO algorithms

f	**GCO**	**CLPSO**
f_1	1.48881e-020 ± 8.05893e-020	2.3060e-019 ± 1.4236e-019
f_2	2.60961e-3 ± 3.20126e-3	19.0364 ± 3.2650
f_3	7.10543e-015 ± 0.0	1.367e-10 ± 5.389e-11
f_4	0.0 ± 0.0	7.785e-12 ± 3.076e-11
f_5	0.0 ± 0.0	5.095e-13 ± 2.176e-13
f_6	0.0 ± 0.0	2.823e-10 ± 3.513e-10
f_7	7.27596e-013 ± 9.06353e-013	1.819e-13 ± 5.55029e-13

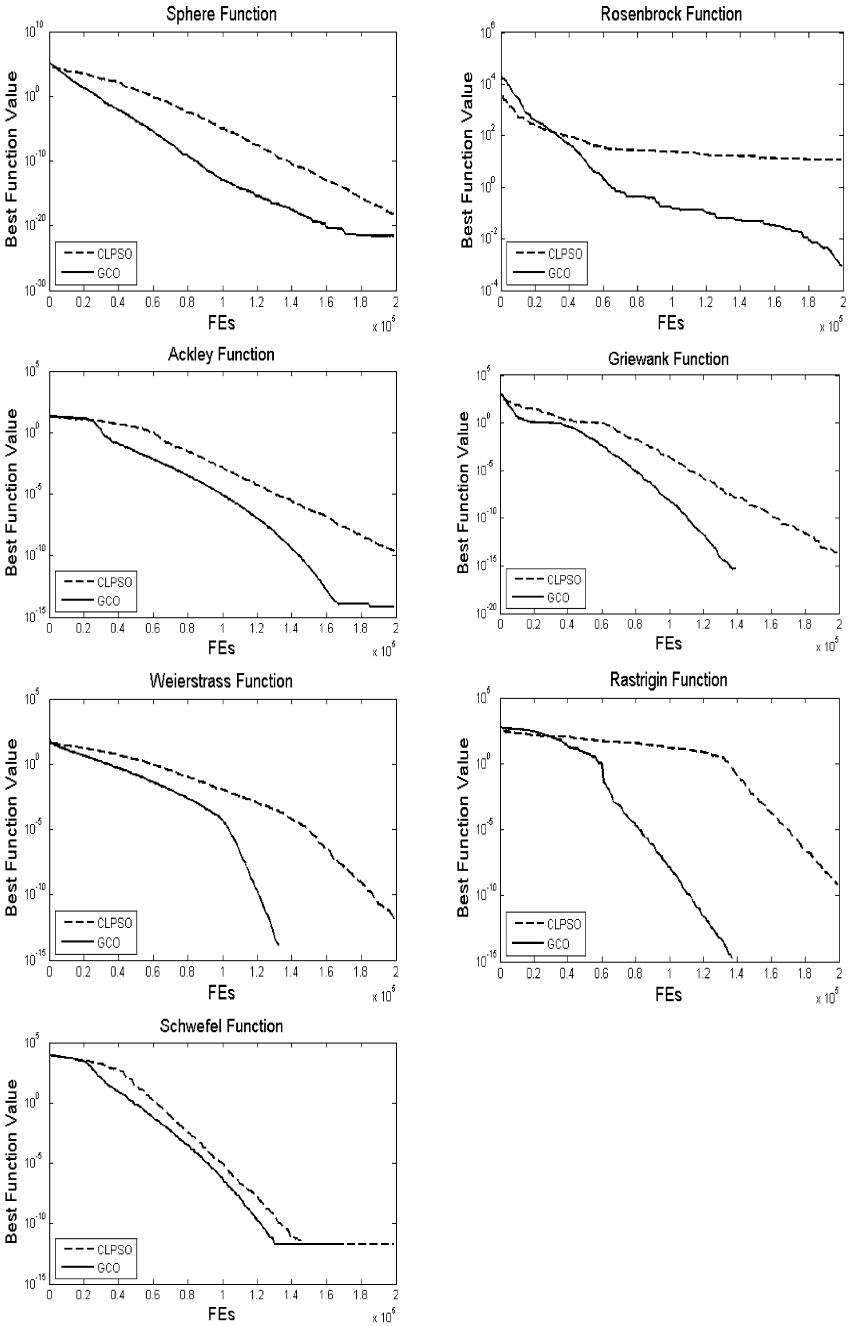

Fig. 3 Convergence characteristics of test functions

5 Conclusions

This paper introduces a novel heuristics-based, derivative-free optimization approach for single-objective functions, which we call a group counseling optimizer (GCO). Instead of mimicking the behavior of living organisms such as birds, fish, ants, and bees, we choose to emulate the behavior of the humans in solving their problems through group counseling. This is motivated by the fact that the human's thinking is, or should be, the most reasonable and influential, and group counseling is in essence a problem-solving technique.

The algorithmic iterations are visualized as counseling sessions, with counselees and counselors. Candidate solutions are progressively improved by means of one of two strategies: (a) other-members counseling or (b) self-counseling.

The approach is successfully applied to seven benchmark functions: two unimodal functions (Sphere and Rosenbrock) and five multimodal functions (Griewank, Rastrigin, Ackley, Weierstrass, and Schwefel). Global optima are reached without being trapped at local optima. Convergence characteristics are empirically studied in terms of the best function values *versus* fitness evaluations. Furthermore, a comparison is made with the comprehensive learning particle swarm optimizer (CLPSO). It is demonstrated that the GCO outperforms the CLPSO for six benchmark functions and is well comparable to it for the seventh function (Schwefel).

The proposed algorithm is seen to be interesting, promising, and readily applicable to many vital areas of optimization. We are currently investigating extension of the GCO to rotated benchmark functions [21] and multi-objective optimization problems [19].

Acknowledgements The authors would like to thank anonymous reviewers for helpful comments.

References

1. Berg, R.C., Landreth, G. L., Fall, K. A.: Group counseling: Concepts and procedures (4th ed.). Philadelphia (1998)
2. Burnard, P.: Practical counselling and helping. Routledge, London (1999)
3. Dixon, D.N., Glover, J.A.: Counseling: A problem-solving approach. Wiley, New York (1984)
4. Dorigo, M., Maniezzo, V., Colorni A.: The ant system: Optimization by a colony of cooperating agents. IEEE Trans. on Systems, Man, and Cybernetics Part B: Cybernetics, 26(1), 29–41 (1996)
5. Eberhart, R.C., Kennedy, J.: A new optimizer using particle swarm theory. In: Proc. of the Sixth International Symposium on Micro Machine and Human Science MHS'95, IEEE Press, 39–43, (1995)

6. Eberhart, R.C., Shi, Y., Kennedy, J.: Swarm Intelligence. Morgan Kaufmann, San Francisco, (2001)
7. Esquivel, S.C., Coello Coello, C. A.: On the use of particle swarm optimization with multimodal functions. In: Proc. Congr. Evol. Comput., vol. 2, Canberra, Australia, 1130–1136 (2003)
8. Gentle, J.E.: Random number generation and Monte Carlo methods — (Statistics and computing). Springer Science and Business Media, Inc. (2003)
9. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison Wesley, Boston, MA (1989)
10. Gupta, A. K., Nadarajah, S.: Handbook of beta distribution and its applications, Marcel Dekker (2004)
11. Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, MI (1975)
12. Kennedy, J., Eberhat, R.C.: Particle swarm optimization. In: Proc. of IEEE International Conference on Neural Networks, No. IV, IEEE Service Center, Piscataway, NJ, 1942–1948 (1995)
13. Kratcer, J., Leende, R.TH.A.J., van Engelen, J.M.L., Kunest, L.: InnovationNet: the Art of Creating and Benefiting from Innovation Networks. Van Gorcum (2007)
14. Lee, C.Y., Yao, X.: Evolutionary programming using mutations based on the levy probability distribution. IEEE Trans. Evol. Comput., vol. 8, 1–13 (2004)
15. Liang, J.J., Qin, A.K., Suganthan, P.N., Baskar, S.: Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. IEEE Trans. Evolutionary Computation 10(3), 281-295 (2006)
16. Moscato, P.: On evolution, search, optimization, genetic algorithms and martial arts: towards memetic algorithms. Technical Report C3P 826, Caltech Concurrent Computation Program 158-79, California Institute of Technology, USA, Pasadena, CA (1989)
17. Pham, D.T., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S., Zaidi, M.: The bees algorithm – a novel tool for complex optimization problems. In: Proc. of 2nd Virtual International Conference on Intelligent Production Machines and Systems IPROMS (2006)
18. Rechenberg, I.: Cybernetic Solution Path of an Experimental Problem. Royal Aircraft Establishment, Farnborough (1965)
19. Reyes-Sierra, M., Coello, C.A.C.: Multi-objective particle swarm optimizers: a survey of the state-of-the-art. International Journal of Computational Intelligence Research, 2(3), 287–308 (2006)
20. Reynolds, R.G.: An introduction to cultural algorithms. In: Proc. of the 3rd Annual Conference on Evolutionary Programming, World Scienfific Publishing, 131-139 (1994)
21. Salomon, R.: Reevaluating genetic algorithm performance under coordinate rotation of bencmark functions. *BioSystems*, vol. 39, 263–278 (1996)
22. Storn, R., Price, K.: Differential evolution – a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report TR-95-012, International Computer Science Institute, Berkeley, CA (1995)