

# A New Approach for Solving Equations Systems Inspired from Brainstorming

Liviu Octavian Maftciu-Scai  
West University of Timisoara, Romania  
lscai@info.uvt.ro

## ABSTRACT

This paper proposes a metaheuristic inspired from human brainstorming combined with concepts from graph theory for solving system of equations. The proposed method is able to find solutions of a given system of equations, even in cases where traditional methods fail. In the cases where no exact solution exists, an approximate solution is good and it can be obtained by the proposed method. Directions for future research are also provided.

## KEYWORDS

metaheuristics, brainstorming, k-partite graphs, cliques, quasi-cliques, system of equations

## 1. INTRODUCTION

### 1.1 Creativity and Brainstorming

Creativity is probably one of the most remarkable attributes of the human being, which essentially distinguishes it from other creatures on our planet. A definition of the concept can be found in the paper [1]. There are many works that describe and analyze human creativity in terms of psychology, such as for example [2, 3] and as a consequence we will not insist here on the subject. In the computer era it was a natural occurrence of a study domain aiming imitating human creativity called *computational creativity* [4, 5]. Computational creativity is a subfield of artificial intelligence who study how to build computational models of creative thought in science and arts. Computational creativity has been applied in many directions as well: design [6], games [7], culinary recipes [8], music [9] etc.

At the same time, human creativity has always played an essential role in solving new problems or in finding new and more effective solutions to solve old ones. For this, there are two forms of creativity in terms of the number of persons involved: singular and collective creativity [4]. One collective creativity technique is the *brainstorming* [10], which mainly aims to the

generation of a large number of creative ideas through intensive and freewheeling group discussion aiming to solve a problem.

The term *brainstorming* was introduced in 1963 by Alex Faickney Osborn [11]. There are four basic rules of brainstorming method:

- *focus on quantity;*
- *without criticism;*
- *welcome unusual ideas;*
- *combining and improving the ideas.*

The technique has generated much controversy in time, being many researchers who criticize it [12] and even some who entirely reject it. We will not do in this paper a complete description of brainstorming method and the problem generated by this because literature is very rich in this respect, a good survey being the work [13]. Anyway, over time, the technique proposed by Osborn has undergone improvements [14, 15]. One of the greatest innovations of the verbal brainstorming proposed by Osborne was the *electronic brainstorming* (EBS) [16].

In the second section of this paper we will review some psychological aspects of brainstorming that directly show interest in terms of the present work.

### 1.2 Optimization Problems Solved by Artificial Intelligence Techniques

Optimization is one of the most important real problems. Over time there have been developed many mathematical methods, mainly heuristic techniques. In recent years, techniques inspired by artificial intelligence have made beneficial contributions to solving optimization problems. Some of them, known as *evolutionary algorithms* were inspired by biological evolution: *genetic algorithms* [17], *genetic programming* [18] etc. Algorithms inspired from nature, particularly from collectivities animal life –known as swarm intelligence- have been proposed during the past time: *ant colony optimization* algorithm (ACO)

[19], *bee colony optimization* algorithm (BCO) [20], *particle swarm optimization* (PSO) [21] etc. A hybridization of evolutionary algorithms, known as *memetic algorithms*, inspired by cultural evolution was first proposed in [22].

Shi proposes in paper [23] an optimization method inspired from human brainstorming process (BSO) as a reply to swarm intelligence. In the proposed algorithm by Shi, the population is composed of individuals represented by the proposed potential solutions for the problem to be solved, randomly generated values. In the next phase, individuals are evaluated and grouped into clusters. After that, each cluster in part is evaluated and modified by positioning the best individual in the center. Using probabilistic methods, new individuals are generated according to the cluster centers, using a selection similar with genetic elitism. The process ends after generating a predetermined number of individuals or a predetermined number of iterations is reached. In this algorithm, the size of population represents the number of ideas generated and an iteration represents a round of idea generation in the brainstorming process. The tests and results on the two benchmark functions presented in paper [23] by author validate the effectiveness and usefulness of the proposed brainstorming optimization algorithm.

A similar approach for multiple optimization problems is proposed in paper [24]. A technical application of BSO was proposed in paper [25]. A working parameters investigation of BSO process is presented in paper [26]. Also, an improved variant of BSO algorithm is proposed in [26]. In [27] the authors propose a brainstorming process modeling using a genetic algorithm like a search technique. In [28] a hybridization between BSO and Teaching-Learning-Based Optimization algorithm (TLBO) with applying in optimal power flow problems is proposed.

### 1.3 Some Used Concepts from Graph Theory

The next section proposes a modeling of creativity like a search space using graph theory, namely the concept of *k-partite* graph. It is known that a *k-partite* graph is a graph whose vertices can be partitioned into *k* disjoint sets so that no two vertices within the same set are adjacent. If *k=2* we

have a *bi-partite* graph and so on. First we describe the modeling proposed for *k=2* followed by an extension to the general case.

The following definitions are known in graph theory and used in our approach. So, in the case of a graph  $G(S, A)$ , where  $S$  is the vertices set and  $A$  is the edges set we have:

**Def.1:** A *biclique*  $C = (S', A')$  is a subgraph of  $G$  induced by a pair of two disjoint subsets  $S' \subseteq S$ ,  $A' \subseteq A$ , such that  $\forall s \in S', a \in A', (s, a) \in E$ , meaning that a biclique in a bipartite graph is a complete bipartite subgraph that contains all permissible edges.

**Def.2:** A *maximum biclique* is a largest biclique in a bipartite graph. From this point of view there are two distinct variants of this problem: the *vertex maximum biclique* problem and the *edge maximum biclique* problem.

**Def.3:** A biclique within another bipartite graph is called a *maximal biclique* if it is not contained in a larger biclique.

The problem can be regarded distinctly in terms of the vertices respectively edges of graphs. The *edge maximum biclique* is often used in biological applications, web community discovery, and text mining because it models more balanced connectivity between the two vertex classes [29] and therefore over time many algorithms have been developed in this respect [30].

**Def.4:** A *quasi-biclique* is similar to biclique but contains almost of its edges. *Maximal quasi-bicliques* are proposed in [31] and are motivated by real-world applications where errors and missing data/information (edges in graph) are present. Also it was shown in paper [31] the versatility and effectiveness of maximal quasi-bicliques to discover correlated data sets.

Figure 1 illustrates the previous concepts as follows:

- graph:  
 $G = (V, E) = \langle \{x, y, a, b, c, d, e\}, \{(x, b), (x, c), (x, d), (y, a), (y, b), (y, c)\} \rangle$

which have a

- bi-partite subgraph:  
 $G_B = (V_1 \cup V_2, E) = \langle \{x, y\} \cup \{a, b, c, d\}, \{(x, b), (x, c), (x, d), (y, a), (y, b), (y, c)\} \rangle$

with

- bi-cliques:  
 $G_{BC1} = \langle \{x\} \cup \{b, c, d\}, \{(x, b), (x, c), (x, d)\} \rangle$   
 $G_{BC2} = \langle \{x\} \cup \{a, b, c\}, \{(y, a), (y, b), (y, c)\} \rangle$

$$G_B = \langle \{x, y\} \cup \{b, c\}, \{(x, b), (x, c), (y, b), (y, c)\} \rangle$$

the last being a maximum biclique from the *edge maximum biclique* problem point of view.

At the same time  $G_{BC1}$  and  $G_{BC2}$  are *vertex maximum bicliques*.

A quasi-biclique example is:

$$G_{BQ} = \langle \{x\} \cup \{a, b, c, d\}, \{(x, b), (x, c), (x, d)\} \rangle$$

where the edges  $(x, a)$  and  $(y, d)$  are missing.

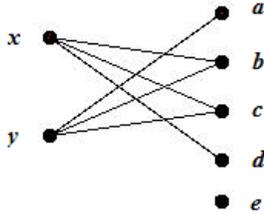


Figure 1 A bi-partite graph example

The previous concepts can be easily extended to the case of *k-partite* graphs, as can be seen in Figure 2. As previously mentioned, a *k-partite* graph is a graph whose vertices can be partitioned into *k* disjoint sets so that no two vertices within the same set are adjacent. We mention that for us it is sufficient to consider a particular case that the bicliques relate to a particular partition, hereinafter called the *main partition*, while the rest are called *secondary partitions*. The red edges from Figure 2 belong to a 4-clique graph. It is known the definition according to which a *n-clique* of an undirected graph is a maximal subgraph in which every pair of vertices is connected by a path of length *n* or less.

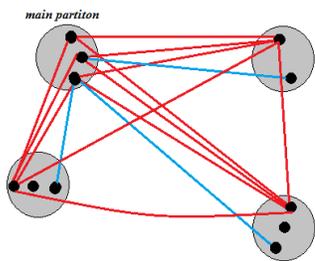


Figure 2 A 4-partite graph example

### 1.4 Solving the Systems of Equations

A frequent problem in numerical analysis is solving the systems of equations. That problem has generated in time a great interest among mathematicians and computer scientists, as evidenced by the large number of numerical methods developed. Besides the classical numerical methods, in the last years were proposed

methods inspired by techniques from artificial intelligence. Hybrid methods have been also proposed along the time [32].

The classical methods are usually divided into exact/direct methods and iterative/indirect methods. In exact methods, the solution is obtained after a fixed number of operations, a number that is directly proportional to system size. In these cases the solutions are affected by rounding errors and are used only when the size of the system is small. The most known direct methods are: Cramer, Gaussian elimination, Gauss-Jordan elimination, LU factorization, QR decomposition etc. In iterative methods the iterative process will be stopped after a preset number of steps or after fulfilling certain conditions. The process involves truncation errors, that is the main disadvantage of these methods. However, because the rounding errors are not cumulated and because requiring a smaller number of operations, iterative methods are preferred especially for large systems of equations. A disadvantage of iterative methods occurs when the system size is smaller than the number of unknowns (under-determined system) when the classical iterative algorithms are not applicable in general. The most popular iterative methods are Gauss-Seidel, Jacobi, Newton, Conjugate Gradient and Broyden.

In the last years, techniques and methods from artificial intelligence have been used to solve systems of equations. In [33] is proposed an algorithm that uses a genetic approach to solve linear systems of equations. This problem is viewed in terms of an optimization task. In [34] the authors investigate the applicability and effectiveness of genetic algorithms in finding the solutions of systems of linear equations. Solving a system of equations is regarded again as an optimization problem. In the case of an infinite number of solutions the proposed algorithm is able to produce more than one set of solutions for a system of equations. In terms of the objective function, the approach is similar to that in [33] but the problem is interpreted as a multiobjective optimization task. Such a multiobjective optimization approach is not new, since similar approaches were proposed also in [35, 36, 37 and 38]. In [39] the proposed method for solving a linear system of equations is based on using a

particle swarm optimization algorithm and an artificial fish swarm algorithm, especially designed for ill conditioned linear systems equations. In paper [40] is proposed a memetic algorithm (MA) to solve linear systems of equations, by transforming the linear system of equations into an optimization problem. Such exploitation of knowledge obtained in a local search/optimization allows the evolutionary programming implementation to produce very good results at a relatively low computational cost.

## 2. THE BRAINSTORMING IN THE PROPOSED METHOD

At the base of the proposed algorithm in this paper are a few elements from modern process of human brainstorming. These will be described briefly below.

### a) the need for experts

After a brainstorming process it is expected to result a better idea/solution than ideas/solutions proposed by each participant in part. In our opinion if all participants are unskilled in the working domain, it is unlikely to reach a good result. At the same time, the presence of experts would allow to remove those ideas that are not suitable from a technical point of view.

In our proposed method the main *expert* role is done by the fitness function of the evolutionary process. The chosen fitness function tells the algorithm how good a particular solution is. That is in fact the difference between left and right side of each equation in part after replacing the possible solution vector (an individual in population) in all the equations. If this difference is equal to zero we have an exact solution. If this difference is smaller than a preset value we say that we have an approximate solution

### b) the diversity of participants

It is necessary to have a great diversity of groups from which participants arise, because the same origin of organizational groups can lead to a premature convergence of the ideas issued. The brainstorming ideas in the proposed method are represented by individuals in the population, where each individual represents a possible solution to an

equation. The diversity is ensured by a controlled random number generator and by evolutionary processes (crossover and mutation).

### c) starting point

The use in the early stages of brainstorming process the known solutions for similar problems can be beneficial in the sense that can position the starting point of new ideas in a favorable position. Obviously there is a risk of obtaining a contrary effect.

The *starting point* in our proposed algorithm is represented by an initial search space that is an arithmetic interval. This initial search space is used in first stage of the process and is determined by computing the degree of dissimilarity [41] of the current system of equations and other similar systems of equations solved, systems that are stored in a database. If the degree of dissimilarity isn't favorable, random real numbers are generated and assigned to individuals. So, in our algorithm, this means clustering and storage of the basic characteristics and solution for each equations system solved in a database.

In the proposed algorithm the individuals are represented by vectors of real numbers with dimension equal to the number of unknowns of the equations system to be solved.

In the next stages the search space is expanded, the expanding having two purposes: to avoid the process of population degeneration and to try to find new solutions in a larger search space. In reality the search space is infinite  $(-\infty, +\infty)$ , but this approach has proven to lead to a very poor convergence of the algorithm, so that was chosen a method of successive expansion of initial arithmetic interval to the right and to the left with a preset value  $r$ . The  $r$  value is set by user at the beginning of the process.

### d) ideas combination

Combining similar or redundant ideas can lead to improved brainstorming process.

In proposed algorithm, an iteration represents a phase in the brainstorming process. Except for the first iteration, in the rest of iterations only a part of the population will be generated randomly, the other part being obtained by genetic crossover and mutation of the best individuals of the previous

population, respecting the principle of genetic elitism.

e) ideas classification

The classification/clustering of ideas is necessary to determine which should take priority in next brainstorming ie next iteration in algorithm.

f) validation of ideas

Brainstorming can be viewed as a method for generating ideas without worrying about solving or not solving the problem. After ideas generation, can begin implementing and validating them as solutions for problem to be solved. So, at the beginning, ideas have to be generated and after that they must be validated to see if these can solve the problem. As mentioned in [42], generating new ideas in a software program is not difficult with artificial intelligence techniques, but the hard problem is their automating validation.

In the proposed algorithm, after replacing the values from possible solution vector in all the equations, the difference between left and right side of each equation in part shows how good a solution is and it is also an expertise in terms of a brainstorming process.

g) creativity like a search space

In work [42] the creative process is seen as a search process in a given space, space which can then be extended. May be, the term “explore” is more adequate than the classical “search”, but we prefer the last in the approach of searching new ideas for a given problem. So, we have in a given space a set of objects, and between these, relationships can be established. Linking a group of objects through relationships will be a new idea. Validation of these new ideas/paths represents new solutions for a given problem. But -through an initial expertise- it’s known that there are objects which cannot be linked by relationships or objects that only by some relationships can or cannot be linked. In this context it is necessary a first expertise to remove those pathways that contain impossible/forbidden links between objects in search space. This first expertise process will eliminate the bad ideas and no longer necessary their validation. The modeling of creativity like a search space in proposed method is made by using the *k-partite* graph concept .

h) termination of the process

A brainstorming process can be closed when it was found a solution for the problem or a preset number of stages was reached.

In the proposed algorithm the termination criteria are: finding an exact solution (fitness function satisfying), finding an approximate solution or reaching a preset number of iterations.

From solution finding point of view the differences mentioned at point g) represents one of the stopping criteria of the algorithm. The ideal case is when all the differences are zero, but in some cases a small enough value (preset by user) can be satisfactory.

### 3. ALGORITHM AND EXPERIMENTS

In accordance with the basic rules of brainstorming and considerations presented in the previous section, a metaheuristic algorithm designed to solve systems of equations (linear and nonlinear), inspired from human brainstorming process is proposed and described below.

In first variant of proposed algorithm, the problem is viewed as an optimization problem. In this case, finding a solution for the system of equation  $f(X)$  -where  $X=\{x_1, x_2, \dots, x_n\}$  vector represents the unknowns- involves finding a solution so that every equation in the system is zero i.e.:

$$\begin{cases} f_1(x_1, x_2, \dots, x_n) = 0 \\ f_2(x_1, x_2, \dots, x_n) = 0 \\ \cdot \\ \cdot \\ \cdot \\ f_n(x_1, x_2, \dots, x_n) = 0 \end{cases} \quad (1)$$

A solution is an assignment of values to the variables  $x_1, x_2, \dots, x_n$  such that each of the equation is satisfied. The solution set is the set of all possible solution for equations system and there are three possible situations: unique, infinitely or no solution for system. In our approach the method is viewed as an optimization problem (finding a solution by minimized a given objective function). This function in our method is  $abs(f_i(X))$  that in term of evolutionary process is named *fitness* function. This fitness function tells the algorithm how good a particular solution is.

The size of population ie the  $X$  vectors, represents the number of ideas generated from

brainstorming point of view. An iteration in proposed algorithm represents a round of idea generation in the brainstorming process.

These considerations are the base of the first method proposed for solving systems of equations inspired by human brainstorming process. The general structure of this algorithm is presented below:

```

Begin
  Read system and other working parameters
  Search Space Initialization
  Generating Random and Guided Initial Population
  Repeat
    Evaluate individuals by their fitness
    Put and Sort individuals in clusters
    Select for each cluster in part the elite group
    Crossover and Mutation for each elite group
    Add new individuals to population
  Repeat
    Random generating new individuals
    If Degenerate
      Expand Search Space
    Else
      Complete Population with new individuals
  Until Not Degenerated
  Compute Average deviation of solutions for each
  -equation in part and for entire system
Until Termination Conditions
End
    
```

From an evolutionary process point of view, the crossover process explores the search space around the already found good or approximate solutions and the mutation process explores the search space for new solutions. In the crossover process it is used a middle point crossover. Two types of mutation were used in the proposed algorithm: first a *pure-random* mutation type to ensure a high diversity of the population in the search space and second, a *non-random* mutation with small random variations of promising chromosomes to increase the likelihood of promising chromosomes. In initial population generation stage, when there isn't a similar system in database, it is used a *selective initialization* that means: a large number of random solutions are randomly generated and then the initial population is selected from these, according to genetic elitism.

The termination conditions used in proposed algorithm were: finding an exact solution (fitness function satisfying), finding an approximate solution or reaching a preset number of iterations.

After experiments it was observed that the proposed method always finds a solution to a given system of equations (linear and nonlinear), an exact solution or an approximate solution, even in situations when conventional methods fail

(determinant null, system is dependent, system doesn't satisfy the convergence conditions, convergence condition satisfied but an infinite number of solutions found/not found, etc.).

In the second approach the elements from graph theory presented in the first section were used to improve the first proposed method.

So, for a  $n$  dimensional system of equations a  $(n+1)$ -partite graph is generated.

The first partition, hereinafter called *main partition* is fixed from number of vertices point of view, the dimension being equal to  $n$ . In this partition the vertices are represented by equations.

The rest of partitions ( $n$  partitions), hereinafter called *secondary partitions*, the vertices are represented by numerical values of possible solution vectors, which are obtained by the evolutionary process described before. These partitions are variable from the number of vertices point of view, due to values coincidence of possible solutions. Each partition corresponds in fact to all possible solutions for each unknown of system in part.

The edges of the graph are determined by the fitness function and suggest that a numerical value from a particular secondary partition is a possible solution to some unknown and some equation of the system.

A solution vector for equations system from this graph representation of the problem is in fact a  $(n+1)$ -clique. Multiple solution are represented by multiple and distinct  $(n+1)$ -cliques.

The *quasi-cliques* represent a starting point for a new crossover process which aims to achieve more quickly the system solution. These quasi-cliques are sorting descending by number of edges and a first part of them are subject to a new evolutionary process, ie crossover and mutation.

In fact, this approach, based on evolutionary process overlapped by a heuristic based on graph theory, is a complex metaheuristic.

A sample example of this new approach can be seen in Figure 3. The graph representation is the final form of the associated graph, after the evolutionary process for a 3-dim system of linear equations:

$$\begin{cases} x + 2y + 3z = 14 \\ x + y + z = 6 \\ 3x + 2y + z = 10 \end{cases} \quad (2)$$

The system equation composed of by  $e1$ ,  $e2$  and  $e3$  equations has multiple solutions, two of them were represented like cliques in Figure 3 (edges in red and blue) and quasi-cliques (one of them colored in yellow).

The main task consists of determining the cliques and if they don't exist, the task continues to determining the quasi-cliques and select the best of these after a greedy selection. In our approach the selected quasi-cliques will be subjects of a new evolutionary process.

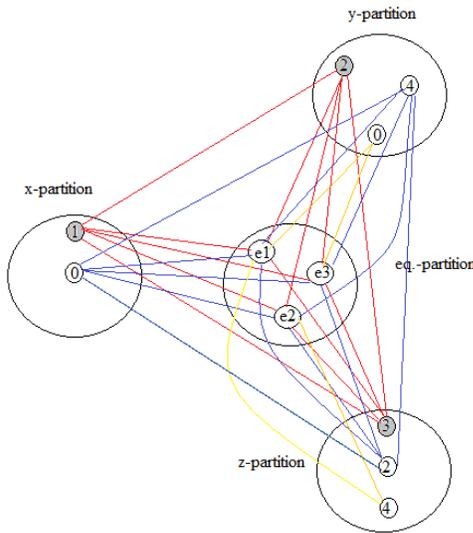


Figure 3: k-partite graph of equations system (1)

So, the first algorithm will be completed with the sequence:

```

...
If cliques found then
    print solution
else
    search for quasi-cliques
    sort quasi-cliques
    Apply crossover to elite of quasi-cliques
...
    
```

The values of the parameters used in the experiments were:

- $r = 10$ ;
- population size of each generation = 200
- maximum number of iteration = 1000
- value for maximum acceptable deviations =  $10^1$

A comparison of system (1) solutions obtained with other methods is given in Table 1. Note that system (1) has an infinite number of solutions and Wolfram Mathematica software can produce a set of linearly independent vectors from which system

solutions can be constructed. This test confirms the validity of the solutions provided by our algorithm.

It has been observed that using the *mass-mutations*, in which is kept a part of the initial configuration of elites, are the most productive evolutionary process in case of our method.

Also, we observed that using a more random population, increases the solution quality but also increases the execution time.

Table 1. Comparative solutions for system (1)

Gauss-elimination	Gauss-Jordan	IU Factorization	Genetic Algorithm	Conjugate gradient	Wolfram Mathematica 10	Proposed Algorithm Inspired from human brainstorming
fail	{4,-4,6}	fail	{1,2,3} {0,4,2} {2,0,4}	{4,-16,14} fail	$x = -2 + z$ $y = 8 - 2z$	{1, 2, 3}, {0, 4, 2}, {2, 0, 4}, {-2,8, 0}, {4, -4, 6}, {1.45, 1.08, 3.45}, {-4.13, 12.26, -2.13}, {75.2, -146.3, 77.2}

Another running example of the proposed method is given further. We consider three nonlinear equations:

$$\begin{aligned}
 (3x - 2)^{1/4} + (15x + 1)^{1/4} - 3 &= 0 & (a) \\
 2^x + 3^x + 6^x - 3x^2 - 5x - 3 &= 0 & (b) \\
 2^{3x+1} - 13 \cdot 2^{2x} + 11 \cdot 2^{x+1} - 8 &= 0 & (c)
 \end{aligned}
 \tag{2}$$

In Table 2, for all possible systems of equations formed by these equations are given the exact and the approximate solutions obtained with the improved proposed method. Through "deviation" we mean the difference between left and right side of each equation in part after the approximate solution replaces the unknowns and show how good the solution is. To note that only integer solutions were searched.

Table 2. Solutions of non-linear system (2)

	The equations of the system	Exact solution	Approximate solution (deviation $\leq 5$ )	
			solution	deviation
1	(a)	1	2	0.77
			5	1.85
2	(b)	1 0 -1	-2	4.61
			0	
			-2	
3	(c)	2 1 -1	0	3
			-2	3.28
			-2	
4	(a)+(b)	1	2	-
5	(a)+(c)	1	2	-
6	(b)+(c)	1	-2	-
		-1		
7	(a)+(b)+(c)	1	2	-

#### 4. CONCLUSION AND FUTURE WORK

The main conclusion is that the proposed method inspired from brainstorming process combined with concepts from graph theory can be very helpful in finding solution for equation and system of equations, linear and nonlinear. The proposed method can find solutions of a given system of equations, even in cases where traditional methods fail (Gauss-elim., Gauss-Jordan, Conjugate gradient etc). In the cases where no exact solution exists, an approximate solution is good and it can be obtained by the proposed method, solutions whose fitness is very close to 0. This approximate solution is good/acceptable in many real life application. Also, systems with multiple solutions can be solved, the proposed method has been able to find all the solutions inside of a given interval preset by user.

The major disadvantage of the proposed algorithm is the running time that is prohibitive in many cases for systems with dimension greater than 50. Even for smaller systems the execution time is greater if the values of the solution vector are distributed in a large interval. So, the proposed method should be refined, and especially made more effective through parallelization and these will be our future concerns.

#### REFERENCES

[1] Runco M.A., Jaeger G.J.: The Standard Definition of Creativity. *Creativity Research Journal*, 24(1), 92–96, ISSN: 1040-0419 (2012)

[2] Lytton H.: *Creativity and Education*. Ed. Routledge, 2012 edition, ISBN:978-0-415-67549-9 (2012)

[3] Sawyer R.K.: *Explaining Creativity. The Science of Human Innovation*, second edition, ISBN-10: 0199737576 (2012)

[4] Maher M.L.: *Computational and Collective Creativity: Who's Being Creative?*. International Conference on Computational Creativity (2012)

[5] Colton S., Wiggins G.A.: *Computational Creativity: the Final Frontier?*. ECAI 2012: 20th European Conference on Artificial Intelligence, eds. L. De Raedt, C. Bessiere, D. Dubois (2012)

[6] Gero J.S., Maher M.L.: *Modeling Creativity and Knowledge-Based Creative Design*. Lawrence Publ. ISBN:0-8058-1153-2 (1993)

[7] Merrick K.E., Isaacs A., Barlow M., Gu N.: A shape grammar approach to computational creativity and procedural content generation in massively multiplayer online role playing games. *Elsevier - Entertainment Computing*, Vol. 4, Issue 2, pp. 115–130, (2013)

[8] Pinel F., Varshney L.R.: Computational creativity for culinary recipes. *ACM- Proceeding CHI EA '14*, pp. 439-442, ISBN: 978-1-4503-2474-8 (2014)

[9] McDermott J.: *Functional Representations of Music*. Proceedings of the Third International Conference on Computational Creativity, ISBN: 978-1-905254668 (2012)

[10] Fen L.H.: A review on the pragmatic approaches in educating and learning creativity. *International Journal of Research Studies in Educational Technology*, ISSN: 2243-7738, Volume 1 Number 1, pp. 13-24, (2012)

[11] Osborn A.F.: *Applied imagination. Principles and procedures of creative problem solving* New York, NY: Charles Scribner's Sons (1963)

[12] Furnham A.: *The Brainstorming Myth*. Wiley Online Library, online: 6.01.2003, DOI: 10.1111/1467-8616.00154 (2003)

[13] Isaksen S.G.: *A Review of Brainstorming Research*. Creativity Research Unit, Creative Problem Solving Group, Buffalo, New-York, Monograph #302 (1998)

[14] Litchfield R.C.: *Brainstorming Reconsidered: A Goal-Based View*. *Academy of Management*, July 1, 2008 vol. 33 no. 3 pp.649-668 (2008)

[15] Mongeau P.A., Morr M.C.: *Reconsidering brainstorming*, ABI/INFORM Global, pg. 14 (1999)

[16] Dennis A.R., Williams M.L.: *Electronic Brainstorming: theory, research and future directions*. *Group Creativity : Innovation through Collaboration*, eds. B. Arlington, Oxford University Press (2003)

[17] Holland J.H.: *Adaptation in natural and artificial systems*. MIT Press Cambridge, MA, USA ©1992 ISBN:0-262-58111-6 (1992)

[18] Koza J.R.: Genetic programming as a means for programming computers by natural selection. *Springer, Statistics and Computing*, June 1994, vol. 4, issue 2, pp 87-112 (1994)

[19] Dorigo M., Maniezzo V., Colnari A.: Ant system: optimization by a colony of cooperating agents. *IEEE Systems, Man, and Cybernetics*, vol. 26 issue:1, pp. 29-41, ISSN: 1083-4419 (1996)

[20] Nakrani S., Tovey C.: On Honey Bees and Dynamic Allocation in an Internet Server Colony. *Proceedings of 2nd International Workshop on the Mathematics and Algorithms of Social Insects*, Atlanta, Georgia, USA (2004)

[21] Kennedy J., Eberhart R.C.: Particle Swarm Optimization. *Proc. IEEE International Conference on Neural Networks*, vol. IV, pp.1942-1948 (1995)

[22] Moscato P., Cotta C., Mendes A.: *Memetic Algorithms*. Springer, *New Optimization Techniques in Engineering Studies in Fuzziness and Soft Computing*, vol. 141, pp 53-85 (2004)

[23] Shi Y.: *Brain Storm Optimization Algorithm*. Springer, *Advances in Swarm Intelligence*, LNCS vol. 6728, pp 303-309 (2011)

[24] Xue J., Wu Y., Shi Y., Cheng S.: *Brain Storm Optimization Algorithm for Multi-objective Optimization Problems*. *Advances in Swarm Intelligence*, Springer, LNCS vol. 7331, pp 513-519 (2012)

[25] Duan H., Li S., Shi Y.: *Predator-Prey Brain Storm Optimization for DC Brushless Motor*. *Magnetics, IEEE*

Transactions, vol. 49 issue:10, pp. 5336 – 5340, ISSN :0018-9464 (2013)

[26] Zhan Z., Chen W, Lin Y., Gong Y., Li Y., Zhang J.: Parameter investigation in brain storm optimization. IEEE, Swarm Intelligence (SIS), 2013 IEEE Symposium, 16-19 April 2013 Singapore, pp.103 – 110 (2013)

[27] Rees J., Koehler G.: Brainstorming, negotiating and learning in group decision support systems: an evolutionary approach. IEEE Proc. of. HICSS-32, ISBN:0-7695-0001-3 (1999 )

[28] Krishnanand K.R., Hasani S.M.F., Panigrahi B.K., Panda S.K.: Optimal Power Flow Solution Using Self-Evolving Brain-Storming Inclusive Teaching-Learning-Based Algorithm. Springer, Advances in Swarm Intelligence, LNCS vol. 7928, 2013, pp 338-345 (2013)

[29] Gilliss N., Glineur F.: A continuous characterization of the maximum-edge biclique problem, ACM DL, Journal of Global Optimization archive, vol. 58 issue 3, March 2014, pp. 439-464 (2014)

[30] Alexe G., Alexe S., Crama Y., Foldes S., Hammer P., Simeone B.: Consensus algorithms for the generation of all maximal bicliques. Elsevier, Discrete Applied Mathematics 145 , pp. 11 – 21, (2004)

[31] Sim K., Li J., Gopalkrishnan V., Liu G.: Mining maximal quasi-bicliques: Novel algorithm and applications in the stock market and protein networks. Statistical Analysis and Data Mining, vol. 2, issue 4, pp. 255–273 (2009)

[32] Maftciu-Scai L.O.: Improved the Convergence of Iterative Methods for Solving Systems of Equations by Memetics Techniques. International Journal of Computer Applications (0975 – 8887) vol. 64, no.17 (2013)

[33] Al-Dahoud A., El-Emary I.M.M., El-Kareem M. A.: Application of Genetic Algorithm in Solving Linear Equation System. MASAUM Journal of Basic and Applied Science, vol.1, no.2 (2009)

[34] Ikotun Abiodun M., Lawal Olawale N., Adelokun Adebowale P.: The Effectiveness of Genetic Algorithm in Solving Simultaneous Equations. International Journal of Computer Applications (0975 – 8887) vol. 14 no. 8 (2011)

[35] Grosan C., Abraham A: Multiple Solutions for a System of Nonlinear Equations. International Journal of Innovative Computing, Information and Control ICIC International , ISSN 1349-4198, (2008)

[36] El-Emary I.M.M., Abd El-Kareem M.M.: Towards Using Genetic Algorithm for Solving Nonlinear Equation Systems. World Applied Sciences Journal 5(3): pp. 282-289, ISSN 1818-4952, (2008)

[37] Grosan C., Abraham A: A New Approach for Solving Nonlinear Equations Systems. IEEE Transaction on Systems, Man and Cybernetics-part A: Systems and Humans, vol. 38, no. 3 (2008)

[38] Mastorakis N.E.: Solving Non-linear Equations via Genetic Algorithms. Proceedings of the 6th WSEAS Int. Conf. on Evolutionary Computing, Lisbon, Portugal, June 16-18, pp. 24-28 (2005)

[39] Zhou Y., Huang H., Zhang J.: Hybrid Artificial Fish Swarm algorithm for Solving Ill-Conditioned Linear Systems of Equations. ICICIS 2011 Proceedings, Part 1, Springer, pp. 656-662 (2011)

[40] Maftciu-Scai L.O., Maftciu-Scai E.J.: Solving Linear Systems of Equations using a Memetic Algorithm. International Journal of Computer Applications (0975 – 8887) vol. 58, no.13 (2012)

[41] Maftciu-Scai L.O.: A New Dissimilarity Measure between Feature-Vectors. International Journal of Computer Applications (0975 – 8887) vol. 64, no.17 (2013 )

[42] Boden M.A.: Creativity and artificial intelligence. Elsevier, Artificial Intelligence 103 pp. 347-356, (1998)