Artif Intell Rev (2014) 41:301–315
DOI 10.1007/s10462-011-9309-8

An optimization algorithm inspired by musical composition

Roman Anselmo Mora-Gutiérrez · Javier Ramírez-Rodríguez · Eric Alfredo Rincón-García

Published online: 3 January 2012

Abstract In this paper we propose a new multiagent metaheuristic based in an artificial society that uses a dynamic creative system to compose music, called "Method of musical composition" or MMC. To show the performance of our proposed MMC algorithm, 13 benchmark continuous optimization problems and the related results are compared with harmony search, improved harmony search, global-best harmony search and self-adaptative harmony search. The experimental results demonstrate that MMC improves the results obtained by the other metaheuristics in a set of multi-modal functions.

Keywords Global optimization · Metaheuristics · Social algorithms · System socio-cultural of creativity · Composition musical

1 Introduction

Mathematical optimization is a dynamic system where new and better methods have been developed, or adapted, to solve more complex problems, and heuristic algorithms are an attractive alternative that have been used to find high quality solutions in some optimization instances. However their design and adaptation implicates creativity and knowledge, and analogies are often used as starting point in this process, for example: physical phenomena (simulate annealing), biological processes and systems (genetic algorithms) and human memory (neuronal networks).

R. A. Mora-Gutiérrez (✉)
Posgrado de Ingeniería, Universidad Nacional Autónoma de México, C.P. 04360 México, D.F., Mexico
e-mail: ing.romanmora@gmail.com

J. Ramírez-Rodríguez · E. A. Rincón-García
Departamento de Sistemas, Universidad Autónoma Metropolitana, C. P. 02200 México, D.F., Mexico
e-mail: jararo@correo.azc.uam.mx

E. A. Rincón-García
e-mail: rigaeral@correo.azc.uam.mx

Recently has been developed a new meta-heuristic algorithm called "Harmony Search (HS)" algorithm, which mimicking the creative process of musical innovation (Geem et al. 2001; Geem 2009). This algorithm and its improved variants (global-best harmony search, self-adaptative harmony search etc) have been successfully applied to solve various benchmarking and real world optimization problems (Geem 2009, 2010). Numerical results reveal that these algorithms can find better solutions compared to other heuristic or deterministic methods.

In this paper we propose a new multiagent metaheuristic based in an artificial society that uses a dynamic creative system to compose music, called "Method of Musical Composition or MMC". The MMC mimics the creative process of musical composition, with agents that can create and change artworks. We designed this metaheuristic using 3 ideas: the main characteristics of artificial societies, the creative process of musical composition and optimization.

The study of the properties and characteristics of a metaheuristic, just as the comparison with other metaheuristics are often carried out empirically , by making use to sets of test functions, whose global and local optimum are known (Dreo 2006; Bersini et al. 1996). Therefore, in this paper we used to 13 benchmark continuous optimization functions (Bersini et al. 1996; Ali et al. 2005; Molga and Smutnicki 2005; Pohlheim 2006; Liang et al. 2006; Riley 2010; Yang 2010; Brest et al. 2006; de los Cobos Silva et al. 2010) to test the proposed procedure. The general structure of these problems is:

$$\min_{(x \in X)} f(x) \tag{1}$$

Such that:

$$x_i^L \leq x_i \leq x_i^U \text{ for all } i = 1, 2, 3, \ldots, n x_i \in \mathbb{R} \tag{2}$$

where

$$f : \mathbb{R}^n \rightarrow \mathbb{R}$$
$$X \subseteq \mathbb{R}^n$$
$$x_i^L \leq x_i \leq x_i^U \text{ feasible range of decision variables.}$$

Some of the 13 benchmark functions have been solved by exact methods like: Newton, Davidon-Fletcher-Powell and quasi-Newton methods (Luenberger 1984; Moudgalya and Joshi 2004), while some others have been solves by metaheuristics like: Tabu Search (TS) (Chelouaha and Siarry 2000), Genetic Algorithmos (GA) (Salomon 1996; Ali et al. 2005) , HS (Wang and Huang 2010; Pan et al. 2010), PSO (Liang et al. 2006) etc.

Numerical results demonstrate that MMC can find better solutions when compared to harmony search, improved harmony search, global-best harmony search and self-adaptative harmony search for the selected benchmark problems.

In next section we describe with more detail these ideas. In Sect. 3, we describe the process of musical composition as creative system. In Sect. 4, we review some related work, and in Sect. 5 are presented analyzed, evaluated and compared the results obtained by proposed algorithm.

2 Heuristic algorithms based on social system

The MMC is a social algorithm, in which composers exchange information among themselves and their environment, and use their knowledge to improve their musical works.

Social algorithms are a subset of evolutionary systems. These models are guided by changes in the social environment, where agents interact among themselves and the environment. These interactions produce a learning that is used to adapt the individual to the current environment faster than what biological evolution can do. Social algorithms include: (a) Cultural Algorithms (Reynolds 1994) introduced by Reynolds as a vehicle for modeling social evolution and learning, (b) Ant Colony Optimization algorithm (Dorigo et al. 1996) which is a metaheuristic inspired by natural systems of real ant colonies, which mimics the ants' behavior in finding the shortest path to reach food sources (c) Particle swarm optimization (Kenedy and Eberhart 1995) these algorithms are inspired by social behavior and movement dynamics of insect swarms, bird flocking and fish schooling, (d) Artificial immune system, these algorithms are inspired by biological immune systems, and exploit the immune system's characteristics of learning and memory to solve optimization problems, (e) Society and civilization (Ray and Liew 2003) this algorithm makes use of the intra and intersociety interactions within a society and the civilization model, (f) Social insects algorithms, metaheuristics inspired in social insects and their self-organized behavior with the idea of the specialization of agents.

3 Creativity and algorithms

Musical composition is the artistic process of creating and innovating an artwork through a recursive process in a creative system. Musical composition is a creative system which implicates a mental phenomenon within creator "composer's creativity", and some events in composer's socio-cultural environment.

Composer's creativity results from making connections between disjoint ideas (de Bono 1993) and it may be produced through a moment of genius or for a recursive reasoning process about a thought, called "hard work" (Jacob 1996).

Creativity has been studied by many sciences like: biology, philosophy, education, socio-cultural studies, artificial intelligence, etc. However, creativity is still one of the most mysterious subjects in human thinking. We can understand creativity by distinguish two different levels: personal and social-cultural (Liu 2000).

These ideas can be used to model, simulate or replicate creativity using a computer, for example, artificial creativity, a branch of artificial intelligence, studies the creative behavior of individuals and artificial societies (Gessler 2010). The model of artificial creativity has been used in music from the start of the process of music composition (Jacob 1996), some examples are: (a) Genetic algorithms and Computer Assisted Music Composition (Horner and Goldberg 1991) (b) GenJam (Biles 1994) (c) Composing whit genetic algorithm (Jacob 1995) (d) Algorithm composition as a model creativity (Jacob 1996) (e) ALICE (Cope 2000) (f)SARA (Cope 2005) etc.

4 New heuristic based on the process of musical composition

4.1 Basic elements for the design

We create the *MMC* algorithm considering the following ideas:

– Musical composition can be considered as an algorithm, since this process use rules, principles and a finite number of steps to create original music of a particular style (Cope 2000).

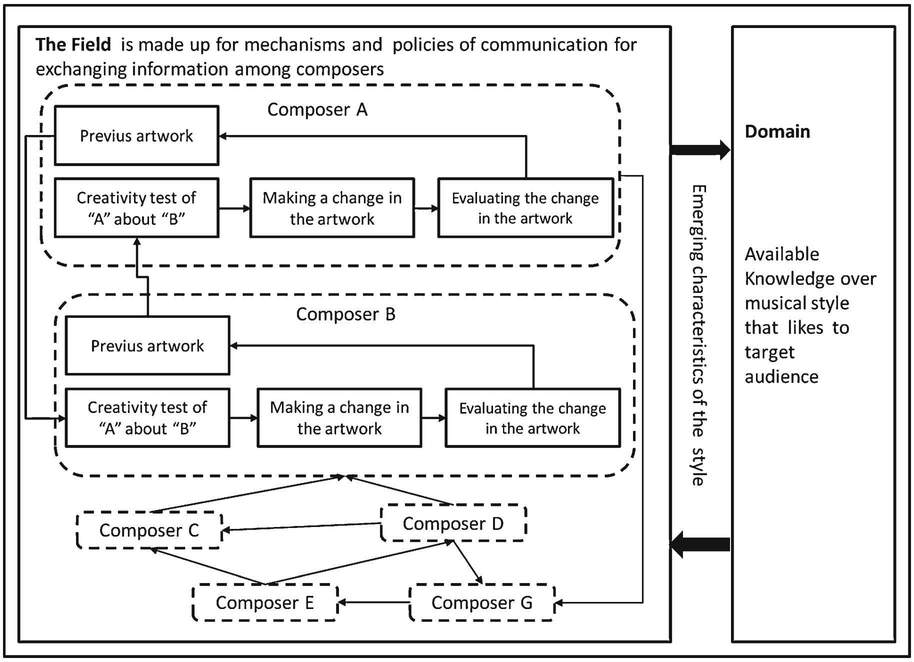
The Field is made up for mechanisms and policies of communication for exchanging information among composers
Composer A
Previus artwork
Creativity test of "A" about "B"
Making a change in the artwork
Evaluating the change in the artwork
Composer B
Previus artwork
Creativity test of "A" about "B"
Making a change in the artwork
Evaluating the change in the artwork
Composer C
Composer D
Composer E
Composer G
Emerging characteristics of the style
Domain
Available Knowledge over musical style that likes to target audience

Fig. 1 Model of the creative system for musical composition

– Musical composition is a system of creativity, so there are interactions among agents.
– The composition includes a set of agents that can learn and use their knowledge in future decisions.

We adapted the Csikszentmihalyis system model of creativity (Heller et al. 2000) to describe the musical composition process (see Fig. 1) This is a multiagents model, where the agents are composers that can create and change artworks. In this model, the interactions among agents produce emergent information, about general characteristics of the style, which is used to generate the domain. On the other side, the field is a set of mechanisms and policies for interaction among composers. These mechanisms are detailed in the relations between composers A and B see (Fig. 1).

Next, we built an analogy between optimization and musical composition (see Table 1).

4.2 *MMC* algorithm.

In the *MMC* algorithm each solution is called "tune" and is represented by an n-dimensional vector:.

$$tune = [x_1 \; x_2 \; \ldots x_n] \tag{3}$$

The basic structure of *MMS* metaheuristic is presented in Algorithm 1. The *MMC* method consists of six steps : 1)initializing the process of optimization (from input to line 4), 2) exchanging of information among agents (from line 6 to line 7), 3) generating for each agent a new tune (from line 9 to line 10), 4) updating the artwork of each agent (from line 11 to

Table 1 Analogy between optimization and musical composition

Comparative question	Process of optimization	Process of musical composition
Why is performed?	Solving problems of mathematical programming	Generate musical works that satisfy desires of the goal audience
Which seeks?	Global optimum	Musical work with the best acceptance of the goal public
How is the quality of the solutions determined?	Objective function	Degree acceptance for part of target audience
Which are the decision elements?	Decision variables	Motives
What properties determine the quality of the solutions?	Values of variables	Sound characteristics of the motives
Is a finite process?	Yes	Yes
Which is the process unit?	Each iteration	Each of the arrangement to the track

Algorithm 1: The basic MMC algorithm

```
Input: $MMC$ method parameters and information about the instance to solve
Output: All the best tunes by composers generated
1  Creating an artificial society with rules of interaction among agents.
2  for each individual in society do
3      Randomly initialize an artwork (considering the upper and lower limits of each variable).
4  end
5  repeat
6      Updating artificial society.
7      Exchanging information among agents.
8      for each individual in society do
9          Updating the matrix of knowledge.
10         Generating and evaluating a new tune ($x_{\star,new}$)
11         if $x_{\star,new}$ is better than the worst tune ($x_{x-worst}$) in artwork of individual then
12             Replacing $x_{x-worst}$ with $x_{\star,new}$ in artwork
13         end
14     end
15     Building the set of solutions
16 until Until termination criterion is satisfied;
```

line 13), 5) building the set of solutions (line 15), and 6) repeating while the stop criterion is not to fulfilled (from line 5 to line 16).

Next, we describe the steps of the MMC algorithm.

4.2.1 Initialization

In this phase the algorithm introduces the characteristics of the instance to be solved and the value of used parameters, these parameters are shown in Table 2.

The number of evaluations (Ne) of the MMC algorithm are:

$$Ne = Nc * \max\_arrangement \tag{4}$$

Using the information introduced in this phase, the algorithm generates, for each composer, a score ($P_{\star,\star,i}$), which is used as memory. The scores are randomly generated, and are structured as shown in the following equation:

Table 2 Characteristics of the parameters of *MMC* algorithm

Description	Conditions of parameter
Maximum number of arrangements ($\max\_arrangement$)	$\max\_arrangement \in \mathbb{N}$
Factor of genius over innovation (ifg)	$ifg \in [0\ ,\ 1]$
Factor of genius over change (cfg)	$cfg \in [0\ ,\ 1]$
Factor of exchanging among agents ($fcla$)	$fcla \in [0, 1]$
Number of composers Nc	$Nc \in \mathbb{R}\backslash(-\infty, 2]$ since a society is a group of people which interact with each other, so there must be at least two agents
Number of chords that form of the artwork Ns	$Ns \in \mathbb{R}\backslash(-\infty, 3]$ For definition, harmony requires at least three chords executed simultaneously
s	r

$$P_{\star,\star,i} = \begin{pmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,n} \\ x_{2,1} & x_{2,2} & \dots & x_{2,n} \\ \vdots & \vdots & \vdots & \vdots \\ x_{Ns,1} & x_{Ns,2} & \dots & x_{Ns,n} \end{pmatrix} \tag{5}$$

where $P_{\star,\star,i}$ is the score of the $i-th$ composer and $x_{j,l}$ is the $l-th$ decision variable of $j-th$ tune.

Algorithm 2: Generating a set of initial scores

```
Input: $n, Nc, Ns, x_l^U$ for all $l = 1, 2, \dots, n$ and $x_l^L$ for all $l = 1, 2, \dots, n$
Output: $P_{\star,\star,i}$
1 for $i = 1 : Nc$ do
2     for $j = 1 : Ns$ do
3         for $l = 1 : n$ do
4             $P_{\star,\star,i} = x_l^L + (rand * (x_l^U - x_l^L))$
5         end
6     end
7 end
```

where: $r\ and\ \sim U[0, 1]$

To produce the scores for each agent was used to Algorithm 2.

4.2.2 Exchange of information among agents

In this phase, the composers exchange information using the policy of interaction. In this work the policy of interaction is: "composer i exchange a tune with composer k if and only if there is a link between them and the worst tune of composer k is better than the worst tune of composer i". This phase is divide in two subphases, a) update of links among composers and b) exchange of information.

(a) Update of links among composers. The objective of this subphase is to generate a change in the social network in time "t" whit respect to the network in time "t−1", see Fig. 2. For this activity the MMC used the procedure shown in the Algorithm 3.

where: v is the $v-th$ arrangements.

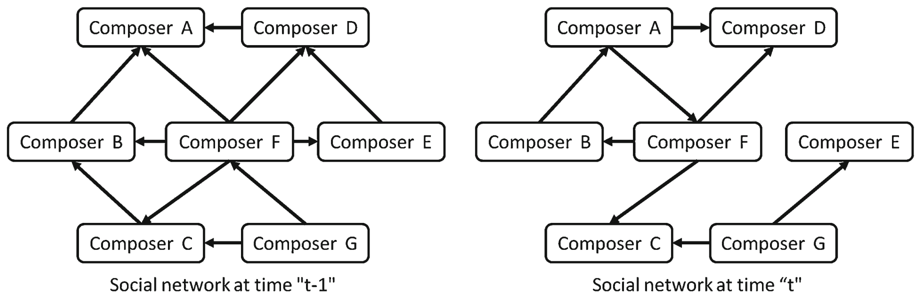

Fig. 2 Dynamic social network

Algorithm 3: Updating of links among composers

```
Input: $v$, $Nc$, $fcla$, previous artificial society
Output: Updating artificial society
1  if $v = 1$ then
2      for $i = 1 : Nc$ do
3          for $k = i + 1 : Nc$ do
4              if $rand < 0.5$ then
5                  Create a link between $i$ composer and $k$ composer.
6              end
7          end
8      end
9  else
10     for $i = 1 : Nc$ do
11         if $rand < fcla$ then
12             Randomly chose a $k$ composer, such that $i \neq k$.
13             Changing the relation between both composers.
14         end
15     end
16 end
17 Checking that each agent has at least one link.
```

After updating the links among composer, the MMC algorithm will perform the next subphase.

(b) Exchange of information. The objective of this subphase is to provide each composer of information about their environment. For this purpose the MMC algorithm used the procedure shown in the Algorithm 4.

4.2.3 Generating a new tune

In this phase each composer will create a new tune using his knowledge. This phase is divided in two subphases, a) building the background and b) creating a new tune.

(a) Building the matrix of knowledge. In this subphase the algorithm creates a matrix that represents the background for each composer ($KM_{\star,\star,i}$). This matrix contains the knowledge of composer i and the information that he got from environment. The MMC algorithm used the routine shown in Algorithm 5.

(b) Creating a new tune. In this subphase each composer will create a new tune using his background and his innovative ideas. The MMC algorithm used the routine shown in Algorithm 6.

Algorithm 4: Exchanging of environmental information

```
   Input: $P_{\star,\star,i}$, $Nc$ and objective function of the instance ($f(x)$)
   Output: Matrix of knowledge of environment of $i-th$ composer ($SC_{\star,\star,i}$)
1  for $i = 1 : Nc$ do
2      $x_{i-worst} \leftarrow$ vector with worst value of $f(x)$ in $P_{\star,\star,i}$.
3      for $k = 1 : Nc \wedge k \neq i$ do
4          $x_{k-worst} \leftarrow$ Vector with worst value of $f(x)$ in $P_{\star,\star,k}$.
5          if there is a link between composer i and composer k then
6              if $f(x_{i-worst})$ is worse than $f(x_{k-worst})$ then
7                  The composer i randomly takes a tune of $P_{\star,\star,k}$ and add this information to $ISC_{\star,\star,i}$.
8              end
9          end
10     end
11 end
```

Algorithm 5: Building and weighting of background knowledge

```
   Input: $P_{\star,\star,i}$, $ISC_{\star,\star,i}$, $Nc$ and $f(x)$
   Output: Matrix of weighting knowledge of $i-th$ composer ($fitness(KM_{j',\star,i})$
1  for $i = 1 : Nc$ do
2      $KM_{\star,\star,i} = P_{\star,\star,i} \bigcup ISC_{\star,\star,i}$.
3      for $k = 1 : Nc \wedge k \neq i$ do
4          $x_{k-worst} \leftarrow$ Vector with worst value of $f(x)$ in $P_{\star,\star,k}$.
5          $a_i = \sum_{j'=i}^{r} f(KM_{j',\star,i})$.
6          for $j' = 1 : r$ do
7              $fitness(KM_{j',\star,i}) = \frac{a_i - f(KM_{j',\star,i})}{a_i * (Nc-1)}$.
8          end
9      end
10 end
```

4.2.4 Updating $P_{*,*,i}$

In this phase each composer decide if replaces the worst tune in his memory for the new tune generate, this decision is based on value of objective function obtained by each. The MMC algorithm used the routine shown in Algorithm 7.

4.2.5 Building the set of solutions

In this phase, the MMC takes the best tune of each composer so this algorithm used the routine shown in Algorithm 8.

In the next section we show the applied tests and the obtained results for MMC algorithm over 13 NLP instances of optimization.

5 Experimental methodology and test problem

In this study the MMC algorithm was tested on thirteen benchmark functions NLP. According to their properties, these functions are divided into three groups: unimodal problems, unrotated multimodal problems and rotated multimodal problems. The properties and the formulas of these functions are presented below:

Algorithm 6: Creating a new tune

```
Input: $KM_{\star,\star,i}, ifg, n, Nc, f(x), fitness(KM_{j',\star,i}), x_l^U$ for all $l = 1, 2, \ldots, n$ and
    $x_l^L$ for all $l = 1, 2, \ldots, n$
Output: A new tune $(x_{\star,new})$
1  for $i = 1 : Nc$ do
2      if $rand < (1 - ifg)$ then
3          for $l = 1 : n$ do
4              $x_l^{max} \leftarrow$ maximum value of $x_l$ into $KM_{\star,l,i}$.
5              $x_l^{min} \leftarrow$ minimum value of $x_l$ into $KM_{\star,l,i}$.
6              $KM_{j,l,i} \leftarrow$ randomly take the $j$ tune of $KM_{\star,l,i}$, considering $fitness(KM_{j,\star,i})$.
7              $KM_{j',l,i} \leftarrow$ randomly take the $j'$ tune of $KM_{\star,l,i}$, considering $fitness(KM_{j',\star,i})$.
8              if $rand < (1 - cfg)$ then
9                  $x_{l,new} = KM_{j,l,i} + (rand * (KM_{j',l,i} - KM_{j,l,i}))$.
10             else
11                 if $rand < 0.5$ then
12                     $x_{l,new} = x_l^{min} + (rand * (KM_{j,l,i} - x_l^{min}))$.
13                 else
14                     $x_{l,new} = x_l^{max} - (rand * (x_l^{max} - KM_{j,l,i}))$.
15                 end
16             end
17         end
18     else
19         for $l = 1 : n$ do
20             $x_{l,new} = x_l^U - (rand * (x_l^U - x_l^L))$
21         end
22     end
23     Determine the value of the objective function of $x_{\star,new}$ $(f(x_{\star,new}))$.
24 end
```

Algorithm 7: Updating of artwork of $i - th$ composer

```
Input: $P_{\star,\star,i}$, $f(x)$ and $Nc$
Output: matrix $P_{\star,\star,i}$ updated
1  for $i = 1 : Nc$ do
2      $x_{x-worst} \leftarrow$ element within $P_{\star,\star,i}$ with the worst fitness value.
3      if $f(x_{x-worst})$ is worse that $f(x_{\star,new})$ then
4          Replacing $x_{x-worst}$ with $x_{\star,new}$ within $P_{\star,\star,i}$.
5      end
6  end
```

Algorithm 8: Building the set of solutions

```
Input: $P_{\star,\star,i}$, $f(x)$ and $Nc$
Output: Set with the best found solutions by composers $(S\star, \star)$
1  for $i = 1 : Nc$ do
2      $Si, \star \leftarrow$ element within $P_{\star,\star,i}$ with the best value of $f(x)$.
3  end
```

5.0.6 Unimodal problems

(A) Rosenbrock function:

$$\begin{array}{c} \min f(x) = \sum_{l=1}^{n-1}(100(x_{l+1} - x_l^2)^2 + (x_l - 1)^2) \\ -30 \leq x_l \leq 30 \text{ for all } l = 1, 2, \ldots .n \\ \text{where } f(x^\star) = 0 \text{ with } [x_1 \ldots x_n] = [1 \ldots 1] \end{array} \tag{6}$$

(B) Step function:

$$\begin{array}{c} \min f(x) = \sum_{l=1}^{n}(\lfloor x_l + 0.5 \rfloor)^2 \\ -100 \leq x_l \leq 100 \text{ for all } l = 1, 2, \ldots .n \\ \text{where } f(x^\star) = 0 \text{ with } [x_1 \ldots x_n] = [0 \ldots 0] \end{array} \tag{7}$$

5.0.7 Unrotated multimodal problems

(C) Schwefels problem 2.26:

$$\begin{array}{c} \min f(x) = 418.9829 - \sum_{l=1}^{n} x_l \sin(\sqrt{|x_l|}) \\ -500 \leq x_l \leq 500 \text{ for all } l = 1, 2, \ldots .n \\ \text{where } f(x^\star) = 0 \text{ with } [x_1 \ldots x_n] = [420.9687 \ldots 420.9687] \end{array} \tag{8}$$

(D) Rastrigin function:

$$\begin{array}{c} \min f(x) = \sum_{l=1}^{n} x_l^2 - 10\cos(2\pi * x_l) + 10 \\ -5.12 \leq x_l \leq 5.12 \text{ for all } l = 1, 2, \ldots .n \\ \text{where } f(x^\star) = 0 \text{ with } [x_1 \ldots x_n] = [0 \ldots 0] \end{array} \tag{9}$$

(E) Ackley's function:

$$\begin{array}{c} \min f(x) = -20\exp^{-0.2(\sqrt{\frac{1}{n}\sum_{l=1}^{n} x_l^n})} - \exp^{\frac{1}{n}\sum_{l=1}^{n}\cos(2\pi * x_l)} + 20 + \exp^1 \\ -32 \leq x_l \leq 32 \text{ for all } l = 1, 2, \ldots .n \\ \text{where } f(x^\star) = 0 \text{ with } [x_1 \ldots x_n] = [0 \ldots 0] \end{array} \tag{10}$$

(F) Griewank function

$$\begin{array}{c} \min f(x) = \frac{1}{4000}\sum_{l=1}^{n} x_l^2 - \prod_{l=1}^{n}\cos(\frac{x_l}{\sqrt{l}}) + 1 \\ -600 \leq x_l \leq 600 \text{ for all } l = 1, 2, \ldots .n \\ \text{where } f(x^\star) = 0 \text{ with } [x_1 \ldots x_n] = [0 \ldots 0] \end{array} \tag{11}$$

5.0.8 Rotated multimodal problems

(G) Rotate hyper-ellipsoid function:

$$\begin{array}{c} \min f(x) = \sum_{l=1}^{n}(\sum_{ll}^{l} x_{ll})^2 \\ -100 \leq x_l \leq 100 \text{ for all } l = 1, 2, \ldots .n \\ \text{where } f(x^\star) = 0 \text{ with } [x_1 \ldots x_n] = [0 \ldots 0] \end{array} \tag{12}$$

(H) Shifted sphere function:

$$\begin{array}{c} \min f(x) = \sum_{l=1}^{n} z_l^n + f_{bais} \\ -100 \leq x_l \leq 100 \text{ for all } l = 1, 2, \ldots .n \\ z_l = x_l - 1 \text{ for all } l = 1, 2, \ldots .n \\ f_{bais} = -450 \\ \text{where } f(x^\star) = -450 \text{ with } [x_1 \ldots x_n] = [1 \ldots 1] \end{array} \tag{13}$$

(I) Shifted Schwefel's problem:

$$\begin{gathered}\min f(x)=\sum_{l=1}^{n}(\sum_{ll}^{l} z_{ll})^2+f_{bais}\\ -100\leq x_l\leq 100 \text{ for all } l=1,2,\ldots .n\\ z_l=x_l-1 \text{ for all } l=1,2,\ldots .n\\ f_{bais}=-450\\ \text{where } f(x^\star)=-450 \text{ with } [x_1\ldots x_n]\ =\ [1\ldots 1]\end{gathered}\tag{14}$$

(J) Shifted Rosenbrock function:

$$\begin{gathered}\min f(x)=\sum_{l=1}^{n-1}100(z_{l+1}-z_l^2)^2+(z_{l-1})^2+f_{bais}\\ -100\leq x_l\leq 100 \text{ for all } l=1,2,\ldots .n\\ z_l=x_l-1 \text{ for all } l=1,2,\ldots .n\\ f_{bais}=390\\ \text{where } f(x^\star)=390 \text{ with } [x_1\ldots x_n]\ =\ [1\ldots 1]\end{gathered}\tag{15}$$

(K) Shifted Rastrigin function:

$$\begin{gathered}\min f(x)=\sum_{l=1}^{n}((z_l^2-10\cos(2\pi * z_l)+10)+f_{bais}\\ -5\leq x_l\leq 5 \text{ for all } l=1,2,\ldots .n\\ z_l=x_l-1 \text{ for all } l=1,2,\ldots .n\\ f_{bais}=-330\\ \text{where } f(x^\star)=-330 \text{ with } [x_1\ldots x_n]\ =\ [1\ldots 1]\end{gathered}\tag{16}$$

(L) Shifted rotated Griewank function:

$$\begin{gathered}\min f(x)=\frac{1}{4000}\sum_{l=1}^{n}z_l^2-\prod_{l=1}^{n}\cos(\frac{z_l}{\sqrt{l}})+1+f_{bais}\\ -100\leq x_l\leq 100 \text{ for all } l=1,2,\ldots .n\\ z_l=(x_l-1)M \text{ for all } l=1,2,\ldots .n\\ M \text{ is a linear transformation matrix}\\ f_{bais}=-180\\ \text{where } f(x^\star)=-180 \text{ with } [x_1\ldots x_n]\ =\ [1\ldots 1]\end{gathered}\tag{17}$$

(M) Shifted rotated Rastrigin function:

$$\begin{gathered}\min f(x)=\sum_{l=1}^{n}((z_l^2-10\cos(2\pi * z_l)+10)+f_{bais}\\ -100\leq x_l\leq 100 \text{ for all } l=1,2,\ldots .n\\ z_l=(x_l-1)M \text{ for all } l=1,2,\ldots .n\\ M \text{ is a linear transformation matrix}\\ f_{bais}=-330\\ \text{where } f(x^\star)=-330 \text{ with } [x_1\ldots x_n]\ =\ [1\ldots 1]\end{gathered}\tag{18}$$

We used the MMC algorithm to solve the above problems in instances of thirty-dimensions. For each function we made 30 independent experiments, determined the runtime and obtained the best solution found. An experiment consisted in 50000 evaluations of objective function.

We programmed the MMC algorithm in Matlab 7.10.0, which was executed in a computer DELL inspiron 1720. After some experiments we decided that the best values for parameters are: $\max\_arrangement = 12{,}500$, $ifg = 0.01$, $cfg = 0.09$, $fcla = 0.045$, $Nc = 4$ and $Ns = 5$.

5.0.9 Experimental desing

In optimization, Z. W. Geem proposed the first metaheuristic based on personal creativity, Harmonic Search (HS), inspired by the improvisation process of Jazz musicians, see (Geem 2009, 2010; Geem et al. 2001; Lee and Geem 2004, 2005).As the proposed algorithm mimics the musical composition process, we decided to compare its performance against HS and some variants because these heuristics mimic a creative process.

Should be noted the MMC is not a HS variant. The MMC is a social algorithm, therefore is a model of dual-inheritance as it allows for a two-way system of learning and adaptation to take place. On the other hand HS only has one-inheritance mechanism.

Besides, the MMC algorithm generates a new solution using a process that is more similar to the technique used by PSO than the technique used by HS and its variants, as for this activity are take random elements based on their fitness, which will be parents of a new tune. An new tune is create for assigning a value for each variable decision trough to approximate the value of its parents.

In this study, thirteen benchmark numerical optimization problems were chosen to evaluate and compare the performance of the proposed MMC algorithm against harmonic search (HS) the improved harmony search (IHS), the global-best harmony search (GHS) and the self-adaptative harmony search (SHS). The data about HS, HIS, GHS and SHS were taken of Pan et al. (2010).

The no-parametric Wilcoxon rank sum test was applied among the results obtained with MMC algorithm and the other heuristics of the test parameters were obtained p (test results of symmetry and mean of the distribution) and h (test results hypothesis, where the null hypothesis that data in vectors x and y are independent, a return value of $h = 1$ indicates a rejection of the null hypothesis at the 5% significance level, while $h = 0$ indicates a failure to reject the null hypothesis at the 5% significance level).

5.0.10 Numerical result

The underline data in the Table 3 are the best solution found by HS, HIS, GHS, SHS and MMC for the set benchmark functions.

In the Table 3 are see that the MMC algorithm generates 3 of best results out of 13 functions. More specifically, comparing with both HS and IHS algorithms, the MMC algorithm produces much better results for all the test functions. While comparing with GHS algorithm, the MMC algorithm obtains better results in seven tests, which are: Rotate hyper-ellipsoid function, Shifted sphere function, Shifted Schwefel's problem, Shifted Rosenbrock function, Shifted Rastrigin function, Shifted rotated Griewank function, Shifted rotated Rastrigin function. Whereas contrasting with SHS the MMC algorithm obtains better results in four tests, which are: Ackley's function, Shifted Rosenbrock function, Shifted rotated Griewank function, Shifted rotated Rastrigin function. Should be noted that the solution found for the MMC algorithm for the problems: Shifted sphere function and Shifted Rastrigin function, are close to the best reported solutions for these instances.

The MMC algorithm has obtained good results for the set of rotated multimodal problems, as the solutions generated by this algorithm are for all instances of test, the best or the second-best option. In the Table 4 we present the data generated after applying the non-parametric Wilcoxon rank sum test.

In the Table 5 we show the running time of the algorithm for each instance.

Table 3 Mean and standard deviation of the benchmark function optimization

Problem	Global optimum	HS	HIS	GHS	SHS	MMC
A	0	350.297 ± 266.691	624.323 ± 559.847	$\underline{49.669 \pm 59.161}$	150.93 ± 131.055	245.43 ± 297.927
B	0	4.233 ± 3.030	3.333 ± 2.196	$\underline{0 \pm 0.000}$	$\underline{0 \pm 0.000}$	1.0667 ± 1.143
C	0	30.262 ± 11.960	34.531 ± 10.400	0.042 ± 0.0503	$\underline{0.004 \pm 0.006}$	3.259 ± 1.245
D	0	1.391 ± 0.824	3.499 ± 1.829	$\underline{0.0086 \pm 0.0153}$	0.0177 ± 0.0675	0.465 ± 0.371
E	0	1.130 ± 0.407	1.893 ± 0.315	$\underline{0.0209 \pm 0.0217}$	0.484 ± 0.357	0.246 ± 0.0686
F	0	1.119 ± 0.041	1.121 ± 0.041	0.102 ± 0.176	$\underline{0.0505 \pm 0.035}$	0.720 ± 0.191
G	0	4297.816 ± 1362.148	4313.65 ± 1062.106	5146.176 ± 6348.79	$\underline{11.766 \pm 7.454}$	220.560 ± 154.72
H	−450	−443.553 ± 2.777	−438.815 ± 3.704	1353.211 ± 361.763	$\underline{-450 \pm 0.00}$	−449.142 ± 0.355
I	−450	3888.18 ± 1115.26	3316.60 ± 1519.41	18440.50 ± 4537.944	$\underline{-431.096 \pm 17.25}$	−224.97 ± 160.814
J	390	3790.70 ± 3271.57	5752.12 ± 3762.54	35046942 ± 22136432	2511.7 ± 3966.5	$\underline{1756.72 \pm 1383.8}$
K	−330	−329.129 ± 0.809	−328.057 ± 0.667	−263.272 ± 9.356	$\underline{-329.861 \pm 0.350}$	−329.319 ± 0.526
L	−180	547.869 ± 0.501	494.756 ± 6.717	546.62 ± 8686070.61	−47.189 ± 13.313	$\underline{-178.509 \pm 0.160}$
M	−330	−274.687 ± 12.863	−270.695 ± 16.223	−192.096 ± 18.646	−232.14 ± 30.033	$\underline{-283.677 \pm 10.416}$

Table 4 Results of Wilcoxon rank test

Parameter	HS	IHS	GHS	SHS
p	0.1119	0.1008	0.3051	0.9183
h	0	0	0	0

Table 5 Mean and standard deviation of running time of MMC algorithm over each instance

Instance	Time in s
A	42.370 ± 5.075
B	39.439 ± 2.366
C	50.326 ± 1.874
D	44.159 ± 4.196
E	49.344 ± 1.840
F	45.598 ± 4.643
G	46.153 ± 4.960
H	40.643 ± 3.744
I	61.071 ± 3.032
J	43.986 ± 3.067
K	41.458 ± 2.383
L	48.442 ± 4.982
M	44.898 ± 2.982

6 Conclusions

In this paper we presented the "Method of Musical Composition" (MMC) which is a multi-agent optimization algorithm. The optimization performance of the MMC algorithm on solving optimization problems was investigated trough of testing 13 NLP instances. The experimental results illustrate that the MMC is an attractive option to work out the set of rotated multimodal problems. The numerical results over Shifted Rosenbrock function, Shifted rotated Griewank function and Shifted rotated Rastrigin function showed that the MMC improved the solutions found by HS, IHS, SHS and GHS.

The results also illustrate that the MMC has higher exploration capability of solution space throughout the whole iteration due to the utilization of interaction among agents. In sum, it is a promising optimization algorithm, and it may be improved.

Future works will also focus on studying the applications of MMC on nonlinear with constraints, combinatorial and discrete optimization problems.

Acknowledgments We thank to Dr. Carlos Gershenson and Dr. Antonin Ponsich for helpful comments in the preparation of this paper.

References

Ali MM, Khompatraporn C, Zabinsky ZB (2005) A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems. J Global Optim 31:635–672

Bersini H, Dorigo M, Langerman S, Seront G, Gambardella LM (1996) Results of the first international contest on evolutionary optimisation (1st iceo). International conference on evolutionary computation, pp 611–615. http://dblp.uni-trier.de

Biles JA (1994) Genjam: a genetic algorithm for generating jazz solos. International computer music conference. International Computer Music Association, Aarhus, pp 131–137

Brest J, Greiner S, Boskovic B, Mernik M, Zumer V (2006) Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. IEEE Trans Evol Comput 10:646–657

Chelouaha R, Siarry P (2000) Tabu search applied to global optimization. Euro J Oper Res 23:256–270

Cope D (2000) The algorithmic composer. A-R Editions Inc., Wisconsin

Cope D (2005) Computer model of musical creativity. MIT Press, London

de Bono E (1993) El pensamiento práctico, Editorial Paidos

de los Cobos Silva SG, Close JG, Andrade MAG, Licona AEM (2010) Búsqueda y exploración estocástica. Universidad Autónoma Metropolitana, Mexico

Dorigo M, Maniezzo V, Colorni A (1996) Ant system: optimization by a colony of cooperating agents. IEEE Trans Syst Man Cybernet 26:29–41

Dréo J, Pétrowski A, Siarry P, Taillard E (2006) Metaheuristics for hard optimization: methods and case studies. Springer, Berlin

Geem ZW (2009) Recent advances in harmony search algorithm. Springer, Berlin

Geem ZW (2010) Music-inspired harmony search algorithm. Springer, USA

Geem ZW, Kim JH, Loganathan GV (2001) A new heuristic optimization algorithm: harmony search. Simulation 76(2):60–68

Gessler N (2010) Fostering creative emergences in artificial cultures. In: Artificial life XII—proceedings of the twelfth international conference on the synthesis and simulation of living systems, MIT Press, pp 669–676

Heller K, Mönks F, Csikszentmihalyi M, Wolfe R (2000) The international handbook of giftedness and talent. Elsevier, New York

Horner A, Goldberg DE (1991) Genetic algorithms and computer assisted music composition. Music composition. In: ICMC91 proceedings, International Computer Music Association, San Francisco, pp 479–482

Jacob B (1995) Composing with genetic algorithms. International Computer Music Association, San Francisco 452–455

Jacob BL (1996) Algorithmic composition as a model of creativity. Organ Sound 1:157–165

Joshi MC, Moudgalya KM (2004) Optimization: theory and practice. Alpha Science International, Ltd., UK

Kenedy J, Eberhart RC (1995) Particle swarm optimization. International Conference Neuronal Networks, UK 1942–1948

Lee KS, Geem ZW (2004) A new structural optimization method based on the harmony search algorithm. Comput struct 82:781–798

Lee KS, Geem ZW (2005) A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. Comput Methods Appl Mech Eng 194:3902–3933

Liang JJ, Qin AK, Suganthan PN, Baskar S (2006) Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. IEEE Trans Evol Comput 10:281–295

Liu YT (2000) Creativity or novelty? Cognitive-computational versus social-cultural. Design Stud 23:261–276

Luenberger DG (1984) Linear and nonlinear programming. Addison-Wesley, Boston

Molga M, Smutnicki C (2005) Test functions for optimization needs. http://www.zsd.ict.pwr.wroc.pl/files/docs/functions.pdf

Pan QK, Suganthan PN, Tasgetiren MF, Liang JJ (2010) A self-adaptive global best harmony search algorithm for continuous optimization problems. Appl Math Comput 216:830–848

Pohlheim H (2006) Geatbx: genetic and evolutionary algorithm toolbox for use with matlab. http://www.geatbx.com/

Ray T, Liew KM (2003) Society and civilization: an optimization algorithm based on simulation of social behavior. IEEE Trans Evol Comput 7:386–396

Reynolds RG (1994) An introduction to cultural algorithms. In: Proceedings of the 3rd annual conference on evolutionary programming,World Scientific Publishing, pp 131–139

Riley MJW, Jenkins KW, Thompson CP (2010) A study of early stopping, ensembling, and patchworking for cascade correlation neural networks. IAENG Int J Appl Math 40(4):307–316

Salomon R (1996) Reevaluating genetic algorithm performance under coordinate rotation of benchmark functions; a survey of some theoretical and practical aspects of genetic algorithms. BioSystems 39:263–278

Wang CM, Huang YF (2010) Self-adaptive harmony search algorithm for optimization. Expert Syst Appl 37:2826–2837

Yang XS (2010) Test problems in optimization. Engineering optimization: an introduction with metaheuristic applications. Wiley, NJ