Revista de Matemática: Teoría y Aplicaciones 2013 **20**(2) : 183–202
cimpa – ucr issn: 1409-2433

AN OPTIMIZATION ALGORITHM INSPIRED BY MUSICAL COMPOSITION IN CONSTRAINED OPTIMIZATION PROBLEMS

UN ALGORITMO DE OPTIMIZACIÓN INSPIRADO EN COMPOSICIÓN MUSICAL PARA EL PROBLEMA DE OPTIMIZACIÓN CON RESTRICCIONES

Roman Anselmo Mora-Gutiérrez[*]

Javier Ramírez-Rodríguez[†]

Eric Alfredo Rincón-García[‡]

Antonin Ponsich[§]

Oscar Herrera-Alcántara[¶]

Pedro Lara-Velázquez[||]

Received: 14/Mar/2012; Revised: 22/May/2013; Accepted: 28/May/2013

Abstract

Many real-world problems can be expressed as an instance of the constrained nonlinear optimization problem (CNOP). This problem has a set of constraints specifies the feasible solution space. In the last years several algorithms have been proposed and developed for tackling CNOP. In this paper, we present a cultural algorithm for constrained optimization, which is an adaptation of "Musical Composition Method" or *MCM*, which was proposed in [33] by Mora et al. We evaluated and analyzed the performance of *MCM* on five test cases benchmark of the CNOP. Numerical results were compared to evolutionary algorithm based on homomorphous mapping [23], Artificial Immune System [9] and anti-culture population algorithm [39]. The experimental results demonstrate that *MCM* significantly improves the global performances of the other tested metaheuristics on same of benchmark functions.

Keywords: Constrained nonlinear optimization, metaheuristics, cultural algorithms, system socio-cultural of creativity, musical composition.

Resumen

Muchos de los problemas reales se pueden expresar como una instancia del problema de optimización no lineal con restricciones (CNOP). Este problema tiene un conjunto de restricciones, el cual especifica el espacio de soluciones factibles. En los últimos años se han propuesto y desarrollado varios algoritmos para resolver el CNOP. En este trabajo, se presenta un algoritmo cultural para optimización con restricciones, el cual es una adaptación del " Método de Composición Musical" o *MCM*, propuesto en [33] por Mora et al., para resolver instancias del CNOP. La adaptación propuesta del *MCM* se aplicó a cinco instancias de prueba del CNOP a fin de evaluar y analizar su comportamiento Los resultados experimentales del *MCM* se compararon con los resultados obtenidos por algoritmo evolutivo basado en homomorfismo [23] , Sistema Inmune Artificial [9] y el algoritmo de anti-cultural [39]. Los resultados experimentales

[*]Departamento de Sistemas, Universidad Autónoma Metropolitana - Azcapotzalco. Avenida San Pablo 180, Colonia Reynosa Tamaulipas, 02200 México D. F. E-Mail: `ing.romanmora@gmail.com`

[†]Departamento de Sistemas, Universidad Autónoma Metropolitana and LIA Université d'Avignon et des Pays de Vaucluse, same address that Mora, `jararo@azc.uam.mx`

[‡]Misma dirección que/*Same address as:* R.A. Mora, E-Mail: `rigaeral@azc.uam.mx`

[§]Misma dirección que/*Same address as:* R.A. Mora, E-Mail: `aspo@azc.uam.mx`

[¶]Misma dirección que/*Same address as:* R.A. Mora, E-Mail: `oha@azc.uam.mx`

[‖]Misma dirección que/*Same address as:* R.A. Mora, E-Mail: `pedro_lara@azc.uam.mx`

muestran que el *MCM* genera resultados significativamente mejores que los obtenidos por las otras metaheurísticas probadas en algunos de los problemas de referencia.

Palabras clave: optimización no lineal con restricciones, metaheurísticas, algoritmos culturales, sistema socio-cultural de la creatividad, composición musical.

Mathematics Subject Classification: 97M40, 90C59, 68T20.

1 Introduction

Many real-world problems can be formulated as a nonlinear optimization problem (NLP). Every NLP includes constrains called “Constrained optimization problem”(CNOP). A CNOP is made up of three basic components: a set of variables, an objective function to be optimized and a set of constraints that specify the feasible spaces of the variables [18]. Given the functions $f : \mathbb{R}^n \to \mathbb{R}, g : \mathbb{R}^n \to \mathbb{R}^p$ and $h : \mathbb{R}^n \to \mathbb{R}^m$, without loss of generality, the general constrained optimization problem can be formulated as:

$$\text{Optimize }_{x \in \mathcal{F} \subseteq \mathcal{S}} f(x) \tag{1}$$

subject to

$$\begin{aligned} g_j(x) &\leq 0, \forall j = 1, 2, \ldots, p \\ h_j(x) &= 0, \forall j = p+1, \ldots, m; x \in \mathbb{R}^n. \end{aligned}$$

where: n is the number of variables, the objective function f is defined on the search space $\mathcal{S} \subseteq \mathbb{R}^n$ and the feasible region is $\mathcal{F} \subseteq \mathcal{S}$. Usually $\mathcal{S}$ is defined as a $n-$dimensional rectangle in $\mathbb{R}^n$ domains of variables defined by their lower and upper bounds ($\mathcal{S} = \{x \in \mathbb{R}^n : x_l \in [x_l^L, x_l^U]\}$). Whereas the feasible set $\mathcal{F}$ is described by inequality (g) and / or equality (h) constraints on the decision variables ($\mathcal{F} = \{x \in \mathbb{R}^n : g_j(x) \leq 0 \ \forall j = 1, 2, \ldots, p; h_j(x) = 0 \ \forall j = p+1, \ldots, m; m \leq n\}$) [23, 29, 3].

Due to the complexity and unpredictability of CNOP, it is very difficult to develop a deterministic method of solution, since the presence of constraints significantly affects the performance of solution strategies (optimization algorithm, heuristics and metaheuristics). A set of techniques non-linear mathematical programming and software are shown in [26, 16], focusing on the contrasting strategies of local optimization and global optimization [2]. In the last years, several metaheuristic methods have been proposed for approaching CNOP problems e.g: an evolutionary algorithm based on homomorphous mappings proposed in [23], in [18] is put forward an adaptation of particle swarm optimization, a cultural algorithm is proposed in [24] and in [39] is presented the anti-culture population algorithm, among others.

In this paper, we propose an extension of the metaheuristic "Musical Composition Method"(*MCM*), which is a cultural algorithm, (see section 2) which was proposed in [33], to solve CNOP. The design of *MCM* was based on the following ideas: a) musical composition can be viewed naturally as an algorithm, since composition process uses rules, principles, and a finite number of steps to create a new track [7], b) musical composition is developed within a creative system and c) composers learn from their experience.

In order to evaluate the performance of *MCM*, numerical experiments were carried out on five instances benchmark CNOP proposed in [29, 30, 31, 32], which are G1, G2, G3, G4 and G5. Numerical results show that *MCM* can find better solutions compared with evolutionary algorithms based on homomorphous mapping [23], Artificial Immune System [9] and the anti-culture population algorithm [39] on a set of cases benchmark of the CNOP.

This paper is organized as follows. In the following section, we examine some basic concepts and previous works related with cultural algorithms constraint handling methods, process of musical composition and creative system. In section 3, we describe the MMC algorithm. In Section 4, we describe the experimental methodology used to analyze the performances of our algorithm, also the obtained results are shown. Finally, in Section 6, we give our conclusions.

2 Cultural algorithms

The term culture does not have a standard definition. For the purposes of this paper, culture is defined as the shared patterns of behaviors and interactions, cognitive constructs, and affective understanding that are learned through a process of socialization. In the human society, the cultural changes are faster than genetic changes so a cultural change takes place in minutes, days, moths, year and decades in contrast a genetic change takes at least a decade [15].

In evolutionary computational, the methods based on social-behavioral models have been recently developed. These algorithms are based on the idea that every individual in a society is adapted and evolves faster through cultural evolution than by genetic inheritance e.g: a) Cultural algorithms [38], b) ant colony optimization algorithm [11], c) particle swarm optimization [22] d) society and civilization [37], etc.

Cultural algorithms (CA) are techniques that add domain knowledge to evolutionary computation methods. CA, which were introduced by Reynolds and Chuang in [38], were developed as a vehicle for modeling social evolution and learning. CA involve a model of dual-inheritance since the evolution takes place in two levels, the macro-evolution that takes places in a cultural level (the belief space) while micro-evolution occurs in a population level (for each individual).

2.1 Previous work

Reynolds and Chung in [38] introduced cultural algorithms, which were used both in [28] and [4] together with strategies of evolutionary programming to solve constrained nonlinear optimization problems.

Jin and Reynolds in [21] developed an evolutionary programming-based cultural algorithm which uses an $n-$dimensional regional based schema, called belief. The belief-cells provides an explicit mechanism to support the acquisition, storage and integration of knowledge within the constraints.

Ray et. al. in [36] presented an evolutionary algorithm for constrained optimization, which incorporates intelligent partner selection for cooperative mating. Ray and Liew in [37] proposed the society and civilization algorithm (SCA), which was based on social behavior. SCA was tested with four instances of constrains optimization problems in engineering, the numerical results showed advantages compared with others methods. Coello and Landa in [5], proposed the cultural algorithm evolutionary programming (CAEP), which uses CA with evolutionary programming, to solve constrains optimization. These authors in [6] proposed to use a cultural algorithm to solve multiobjective optimization problems. Coello and Landa in [24] proposed a cultured differential evolution (CDE) approach to handle the constrained optimization problems.

Tang and Li in [39] proposes a triple spaces cultural algorithm in which a new framework called anti-culture population. The numerical results obtained in theirs tests showed advantages compared with others methods.

The techniques to adapt Evolutionary algorithms for solving constrained optimization problems are divided into five basic categories[31]: a) methods based on preserving the feasibility of solutions, b) methods based on penalty function, c) methods that makes a clear distinction between feasible and infeasible solutions, d) methods based on decoders and e) Other hybrid methods.

2.2 Creativity and algorithms

Musical composition is the artistic process of creating and innovating an artwork through a recursive procedure that involves a phenomena of creativity. Creativity can be understood by distinguishing two different levels: personal and social-cultural [27]. The personal composer's creativity results from connection making between disjoint ideas [10] and it may be produced through a moment of genius or for a recursive reasoning process about a thought, called "hard work" [19].

Besides, the musical composition process is not an isolated phenomenon but can rather be situated within a creative framework, meaning that some events in the composer's socio-cultural environment might have an impact on the creative process.

These ideas can be used to model, simulate or replicate creativity using a computer. For instance, artificial creativity, which is a branch of artificial intelligence, studies the creative behaviour of individuals and artificial societies [15]. The model of artificial creativity has been used in music from the beginning of the music composition process [19]. Some examples are: a) Genetic Algorithms (GAs) and Computer Assisted Music Composition [17], which is an application of genetic algorithms to the problem of composing music; b) GenJam [1], which is a GA based model of a novice jazz musician learning to improvise; c) Composing with Genetic Algorithms [20], in which GAs were used to produce a set of data filters that were identified as acceptable material from the output of a stochastic music generator; d) Composition Algorithm as a creativity model [19], in which both GAs and the idea of creativity (generated for "hard work") were used to into compositional process ; e) Algorithmic Integrated Composition Environment (ALICE), which is based in analysis, and pattern matching of musical pieces [7] and f)Simple Analytic Recombinancy Algorithm (SARA), which is based in analysis and recombination of information [8], etc.

3 New heuristic based on the process of musical composition

Designing a metaheuristic method is a process that involves researchers's creativity and knowledge. Usually analogies are used as a starting point in these process, e.g: physical phenomena (simulate annealing SA), biological system and processes (ant colony optimization ACO, genetic algorithms GA), human memory (neuronal networks), social systems (Cultural algorithms) etc. In this work we used an analogy between music composition and optimization. It should be mentioned that Z. W. Geem was the first in propose a metaheuristic based on system of creativity, which is called Harmonic Search (HS). For specific information about HS see [12, 13, 14, 25].

In MCM algorithm each solution is called "tune" ($tune_{\star,\star,\star}$), which is an $n-$dimensional vector. Every element of $tune_{\star,\star,\star}$ is called motif ($y_{\star,\star,l}$) which has values in the range from -1 to 1,so a $tune_{\star,\star,\star} \in [-1,1]^n$ (see equations 2 and 3). This idea is based on decoding of the search space proposed by Kosiel and Michalewicz in [23], which is a technique for handling to restrictions.

$$tune_{i,q,\star} = [y_{i,q,1} y_{i,q,2} \cdots y_{i,q,n}] \tag{2}$$

where: $tune_{i,q,\star}$ is the tune q of the composer i and $y_{i,q,l} \in [-1,1]$ is $l-th$ motif in the $q-th$ tune.

$$y_{i,q,l} = \frac{2 * x_{i,q,l} - (x_l^U + x_l^L)}{x_l^U - x_l^L} \;\forall l = 1,\ldots,n \tag{3}$$

where: $x_{i,q,l}$ is the value assigned to $l-th$ decision variable associated with the tune $y_{i,q,l}$ (see equation 4) and x_l^L and x_l^U are lower and upper bounds, respectively.

$$x_{i,q,l} = y_{i,q,l} * \left(\frac{x_l^U - x_l^L}{2}\right) + \frac{x_l^U + x_l^L}{2} \;\forall l = 1,\ldots,n\ ,\ [23]. \tag{4}$$

In this paper, we used the relation of isomorphism between $[-1,1]^n$ and $\mathcal{S}$, which was proposed in [23] . Isomorphism can be defined as an one-to-one mapping ($\mathcal{H}$) between the hypercube $[-1,1]^n$ and $\mathcal{S}$, which is denoted as $\mathcal{H} : [-1,1]^n \to \mathcal{S}$, which is formally defined in equation 5

$$\mathcal{H}(tune_{i,q,\star}) = x_{i,q,\star}\ ,\ [23]. \tag{5}$$

Without loss of generality the equation 5 can be formulated as:

$$\mathcal{H}([y_{i,q,1}, y_{i,q,2}, \ldots, y_{i,q,n}]) = [x_{i,q,1}, x_{i,q,2}, \ldots, x_{i,q,n}]. \tag{6}$$

A summarized description of *MCM* is as follows. Initially, the *MMC* generates a social network with Nc composers and a set of links, among composers, randomly created. After, for $i-th$ composer ($\forall\ i = 1 \ldots Nc$) is randomly created a set of artworks (an artwork is a solution), which is used as previous knowledge by $i-th$ composer, this previous knowledge is called score matrix ($P_{\star,\star,i}$). While the stopping criterion is not met, the following steps are performed: a) social networks' links are updated, b) $i-th$ composer exchanges information with $j-th$ composer ($\forall\ i = 1 \ldots Nc$ and $i \neq j$), if there is a link among themselves, c) $i-th$ composer decides to select information of the $j-th$ composer, if the worst solution in $P_{\star,\star,j}$ ($tune_{j,worst}$) is better that worst solution in $P_{\star,\star,i}$ ($tune_{i,worst}$), d) $i-th$ composer builds his acquired knowledge as matrix ($ISC_{\star,\star,i}$) with based on information previously selected, e) $i-th$ composer builds his knowledge background ($KM_{\star,\star,i}$) through he concatenates $P_{\star,\star,i}$ and $ISC_{\star,\star,i}$, f) $i-th$ composer creates a new tune ($x_{i,new}$) based on his $KM_{\star,\star,i}$, and g) finally, $i-th$ composers decides on replace or not $tune_{j,worst}$ by $x_{i,new}$ into $P_{\star,\star,i}$, he takes this decision based on the quality of $x_{i,new}$. Generally, the stopping criterion is the maximum number of arrangements ($max_arrangement$), or iterations. The basic structure of the initial version of *MMC* is presented in Algorithm 1.

The *MCM* method consists of six steps: 1)initialize the process of optimization (from input to line 9), 2)exchange of information among agents (from line 11 to line 12), 3)generate for every agent a new tune (from line 17 to line 20), 4)update the artwork of each agent (from line 21 to line 23), 5)build the set of solutions (line 25), and 6) repeat while the stoping criterion is not fulfilled (from line 10 to line 26). For more information about *MCM* see [33, 35, 34].

For each composer in the *MCM* algorithm, should generated a $P_{i,\star,\star}$, which works as $i-th$ composer's memory (experience and knowledge). Before building $P_{i,\star,\star}$ should generated both the matrix of tunes (MT) and the matrix of constraints (MC). MT and MC are structured as shown in equations 7 and 9, respectively.

Algorithm 1: The basic *MCM* algorithm.

```
Input: MCM method parameters and data about the instance to
       solve, which should be expressed through the model shown
       in equation (1)
Output: The best tunes generated by composers
1  Creating an artificial society with rules of interaction among agents.
2  for every individual in society do
3      repeat
4          Randomly generate the tune $q$ by composer $i$ ($tune_{i,q}$).
5          Determine how many and which constraints are violated by
           the tune $q$.
6          Evaluate tune $q$ ($\phi(tune_{i,q})$ .
7      until Creating $Ns$ tunes;
8      Writing the score ($P_i$) with the artworks of the composer $i$
9  end
10 repeat
11     Update the artificial society.
12     Exchange information among agents.
13     for each individual in society do
14         Updating the matrix of knowledge.
15         Evaluate fitness of every tune in matrix of knowledge.
16         $tune_{i,worst} \leftarrow$ is the tune in matrix of knowledge with both
           worst value of function objective and most number of
           violated constraints.
17         Generate a new tune ($tune_{i,new}$)
18         Determine how many and which constraints are violated by
           the new tune.
19         Evaluate new tune ($\phi(tune_{i,new})$).
20         if $tune_{i,new}$ is better than $tune_{i,worst}$ (based on the set of
           criteria for update of artwork) then
21             Replacing $tune_{i,worst}$ with $tune_{i,new}$ in $P_i$.
22         end
23     end
24     Build the set of solutions
25 until Termination criterion is satisfied;
```

$$MT_i = \left(\begin{array}{cccc|c} y_{i,1,1} & y_{i,1,2} & \cdots & y_{i,1,n} & \phi(tune_{i,1,\star}) \\ y_{i,2,1} & y_{i,2,2} & \cdots & y_{i,2,n} & \phi(tune_{i,2,\star}) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ y_{i,Ns,1} & y_{i,Ns,2} & \cdots & y_{i,Ns,n} & \phi tune_{i,Ns,\star}) \end{array} \right) \tag{7}$$

where: MT_i is the matrix of tunes of the composer i; $y_{i,q,j}$ is the motif j in the tune q, which is randomly initialized as a uniformly distributed value between the -1 and 1 and $\phi(tune_{i,q,\star})$ is defined as:

$$\phi(tune_{i,q,\star}) = \begin{cases} f((x_{i,q,\star})) & \text{if all constraints are satisfied by } x_{i,q,\star} \\ f((x_{i,worst})) & \text{if at least 1 constraint is violated by } x_{i,q,\star} \end{cases} \tag{8}$$

where: $f(x_{i,q,\star})$ is the result of evaluating the function objective on $(x_{i,q,\star})$, for which the $tune_{i,q,\star}$ is converted in its corresponding $x_{i,q,\star}$ and $f((x_{i,worst}))$ is the worst value of objective function on feasible solutions of the composer i.

$$MC_i = \left(\begin{array}{cccc|c|c} c_{i,1,1} & c_{i,1,2} & \cdots & c_{i,1,m} & \mathcal{C}_{i,1,\star} & dif_{i,1} \\ c_{i,2,1} & c_{i,2,2} & \cdots & c_{i,2,m} & \mathcal{C}_{i,2,\star} & dif_{i,2} \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ c_{i,Ns,1} & c_{i,Ns,2} & \cdots & c_{i,Ns,m} & \mathcal{C}_{i,Ns,\star} & dif_{i,Ns} \end{array} \right) \tag{9}$$

where: MC_i is the matrix of constraints of the composer i; $c_{i,q,j}$ indicates whether the $j-th$ constraint is either feasible or unfeasible by $q-th$ tune (see equation 10); $\mathcal{C}_{i,q,\star}$ is total of constraints violated by tune q (see equation 11); $dif_{i,q}$ is the sum of differences among right side of every constraint j and the value obtained by tune q in this constraint (see equation 12).

$$c_{i,q,j} = \begin{cases} 1 & \text{If the } j-th \text{ constrain is violated by tune } q \\ 0 & \text{If the } j-th \text{ constrain is satisfied by tune } q \end{cases} \tag{10}$$

.

$$\mathcal{C}_{i,q,\star} = \sum_{j=1}^{m} c_{i,q,j} \tag{11}$$

.

$$dif_{i,q} = \sum_{j=1}^{p} |h_j(tune_{i,q,\star})| + \sum_{j=p+1}^{m} |g_j(tune_{i,q,\star})| \tag{12}$$

Function	Number of variables	Type of decision variables	ρ	LI	NE	NI	α
G1	13	Quadratic	0.0111%	9	0	0	6
G2	k	Nonlinear	99.8474%	0	0	2	1
G3	k	Polynomial	0.0000%	0	1	0	1
G4	5	Quadratic	51.1230%	0	0	6	2
G5	4	Cubic	0.0000%	2	3	0	3

Table 1: Summary of 5 test cases [31].

where: $k = 50$, $\rho = \frac{|\mathcal{F}|}{|\mathcal{S}|}$, LI is the number of linear inequalities,NE is the number of nonlinear equations, and NI is the number of nonlinear inequalities.

.

$P_{i,\star,\star}$ is built as shown the equation 13

$$P_{i,\star,\star} = \left(\begin{array}{c|c} MT_i & MC_i \end{array} \right) \tag{13}$$

It should be noted, the MCM is not a variant of HS. Since the MCM is a cultural algorithm because it has a model of dual-inheritance as it allows for a two-way system of learning and adaptation to take place, in contrast the HS is not a a cultural algorithm because it has only one-inheritance mechanism. Also, the MCM algorithm generates a new solution using a process most similar to that is used by PSO than is occupied by HS.

4 Experimental methodology and test problems

4.1 Test functions

This section presents the computational experiments and associated results obtained by the MCM algorithm on a set of CNOP. We used the MCM algorithm to solved the problems G1, G2, G3 ,G4, and G5 of test set G-suite [29, 31], with the aim of analyzing the performance of our algorithm. A summary with characteristic of test problems is shown in the Table 1.

Considering the stochastic nature of the MCM algorithm, 20 independent replications were performed for each instance (this number of replications has been used in other studies see [9, 39]). For each replication, we registered both the computational execution time and the best target function value. Next we obtain the best mean and worst value and standard deviation of the results. Next, the results obtained are then compared with those of other heuristic algorithms. The techniques selected for comparison are:

- Evolutionary algorithm with homomorphous mappings for constrained optimization (KM) with 1,400,000 fitness function evaluations [23]

- Artificial Immune System (AIS) with 350,000 fitness function evaluations [9]

- Anti-culture population algorithm (TSCGA) with 100,000 fitness function evaluations [39]

Furthermore, a non-parametric Wilcoxon rank sum test is applied to the results obtained by both *MCM* variants and for the other tested heuristic algorithms. The null hypothesis is that data in two solution sets are independent: a test value returned equal to $h = 1$ 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. In the statistical test are computed parameters p (standing for the symmetry and mean of the distribution) and h (which is test results hypothesis)

4.1.1 Parameter setting for the *MCM* algorithm

Because heterogeneity in the set test, we took decision of setting of parameters for each test instance. For tuning of parameters, we used a brute-force approach. In Table 6 are shown the parameter settings used by *MCM* algorithm in every test-case.

Therefore 20 runs were executed for each instance. A run consisted performing 240,000 evaluations of objective function. This number of evaluations is closed of the average of 350,000; and 100,000 evaluations which are the number of evaluations used by AIS and TSCGA, which are included in the comparison.

The way to create a new tune was determined through previous experiments.

Parameter	G1	G2	G3	G4	G5
$\max\_arrangement$	48000	48000	12000	48000	48000
ifg	0.005	0.005	0.001	0.005	0.005
cfg	0.02	0.02	0.0009	0.01	0.01
Nc	5	5	20	5	5
Ns	10	10	5	10	3
Way to create a new tune	First	First	Second	First	First

Table 2: Parameter settings of *MCM* algorithm.

where: $\max_a$ is the maximum number of arrangements to be performed by each composer (stopping criterion); ifg is the factor of genius over innovation; cfg is the factor of genius over change; Nc is the number of composers and Ns is the number of tuner for each composer.

4.1.2 Experimental results and discussions

The *MCM* algorithm was implemented in Matlab R2007a. Results obtained for *MCM* algorithm on set test are shown in the Table 3 . This Table synthesizes, for each case, the best, worst, mean, variance (s^2), standard deviation (s) and $\rho = \frac{\text{number of solutions } \in \mathcal{F}}{\text{number of solutions } \in \mathcal{S}}$, computed over 20 runs. In the Table 4 are shown confidence interval at 95% determined for bootstrap method.

We compared our algorithm against three state-of-the-art approaches: KM, AIS, TSCGA. The best results obtained by each approach are shown in Table 5. The mean values provided are compared in Table 6 and the worst results are presented in Table 7. The results provided by these approaches were taken from the original references for each method.

Based on previous Tables, we can say that our algorithm achieves the best of top results in 60% of cases, the mean of top results in 20% of cases and the worst of top results in 40% of cases.

Based on information obtained by non-parametric Wilcoxon rank sum test (see Table 8), we can say that the *MCM* algorithm obtains performance levels similar to those of other metaheuristics considered in this experiment

In the Table 4.1.2 we show the running time of our algorithm for each test-problem.

Test-problem	ρ	Optimum	Best	Mean
G1	1	-15	-15	-14.9
G2	1	0.802964	0.8036	0.7864
G3	0	1	0.9999	0.9839
G4	1	-30005.7	-30664.9924	-30664.9913
G5	0.8	5126.498‡	4232.57159	4896.5759

Test-problem	Worst	s^2	s
G1	-13	2.00E-01	4.47E-01
G2	0.7395	3.18E-04	1.78E-02
G3	0.7768	2.47E-03	4.97E-02
G4	-30664.9705	2.38E-05	4.88E-03
G5	5612.51901	1.48E+05	3.85E+02

‡ Do not know the optimal, the best known is 5126.4981.

Table 3: Summary of results obtained by *MCM* algorithm. The test G2 was run with $k = 20$ variables, whereas G3 with $k = 10$.

Test-problem	Lower limit	Upper limit
G1	-15	-14.7
G2	0.77871	0.79413
G3	0.95913	0.99755
G4	-30664.99236	-30664.98909
G5	4741.83877	5056.98263

Table 4: Confidence Interval 95% obtained by method bootstrap.

5 Conclusions

In this paper, we have presented a novel cultural algorithm for CNOP, which mimics creativity within music composition process. The creativity in *MCM* is made up two different levels: personal and social-cultural. The *MCM* uses an artificial society, where interactions among agents cause changes over characteristics of artworks in following levels: A) Personal so every composer proposes a change in his artwork considering both his experience and his knowledge (own and selected), B) Local so at time "t_v", the $i-th$ composer can be influenced by tunes of other composer, with which he has a link, and C) Social: in this level, a metapatters (common characteristics in artworks generated into society) emerge, which is results

Test-problem	KM	AIS	TSCGA	MCM
G1	-14.7864	**-15.0000**	**-15.0000**	**-15.0000**
G2	0.7995	0.7703	0.7908	**0.8036**
G3	0.9997	**1.0000**	0.9999	0.9999
G4	-30664.5000	-30665.0815	-30665.3743	-30664.9924
G5	-	5126.6861	-	**4232.5716**

Should be noted that *MCM* has found the best results for test-cases in three times.

Table 5: Comparison of the best results obtained for algorithms.

Test-problem	KM	AIS	TSCGA	MCM
G1	-14.7082	**-15.0000**	**-15.0000**	-14.9000
G2	**0.7967**	0.7040	0.7554	0.7864
G3	0.9989	0.9970	**0.9995**	0.9839
G4	-30655.3000	-30648.1750	-30664.8926	-30664.9913
G5	-	5307.2016	-	**4896.5759**

Should be noted that *MCM* has found the best results for test-cases in one time.

Table 6: Comparison of the mean results obtained for algorithms.

Test-problem	KM	AIS	TSCGA	MCM
G1	-14.6154	**-15.0000**	**-15.0000**	-13.0000
G2	**0.7912**	0.5956	0.7094	0.7395
G3	**0.9978**	0.9810	0.9987	0.7768
G4	-30645.9000	-30613.4426	-30663.9483	**-30664.9705**
G5	-	5927.3671	-	**5612.5190**

Should be noted that *MCM* has found the best results for test-cases in two time.

Table 7: Comparison of the worst results obtained for algorithms.

Algorithm	Best		Mean		Worst	
	p	h	p	h	p	h
KM	0.97380076	0	1	0	0.5993607	0
AIS	0.97380076	0	0.94764453	0	1	0
TSCGA	1	0	1	0	1	0
MCM	1	0	1	0	1	0

Table 8: Results of non-parametric Wilcoxon rank sum test.

Test problem	Mean	Maximum	Minimum	s^2	s
G1	300.57	311.83	294.64	21.44	4.63
G2	328.63	340.82	321.22	20.51	4.53
G3	423.89	434.61	412.17	35.87	5.99
G4	254.83	259.24	249.23	7.18	2.68
G5	256.15	336.23	219.87	949.45	30.81

Table 9: Running time of *MMC* algorithm in seconds.

of interactions among agents and it is built with characteristics accepted and shared by the composers in the society.

The numerical results proved that our algorithm achieves the best of top results in 60% of cases, the mean of top results in 20% of cases and the worst of top results in 40% of cases. Also, the numerical results of non-parametric Wilcoxon rank sum test proved that our algorithm has a similar performance that other metaheuristics used in this experiment.

The results also illustrate that the *MCM* has a higher exploration capability of the solution space throughout the whole iteration due to the use of interaction among agents. Among our future work will be the study strategies of self-adapting of parameters involved in *MCM*. Also we will apply the MCM algorithm on combinatorial and discrete optimization problems.

References

[1] Biles, J.A. (1994) "GenJam: A genetic algorithm for generating jazz solos", *International Computer Music Conference. Aarhus, Denmark: International. Computer Music Association*: 131–137.

[2] Bekrar, A.; Chaabane, S.; Trentesaux, S.; Bornschlegell, A.; Pellé, J.; Harmand, S. (2011) "Hybrid PSO-tabu search for constrained nonlinear optimization problems ", *ICSI 2011: International Conference on Swarm Intelligence*.

[3] Cai, Z.; Wang, Y.A. (2006) "Multiobjective optimization based evolutionary algorithm for constrained optimization". *IEEE Transactions on evolutionary computation* **10**: 658–675.

[4] Chung, C.J.; Reynolds, R.G. (1996) "A testbed for solving optimization problems using cultural algorithms", *Proceedings of the Fifth Annual Conference on Evolutionary Programming*: 225–236.

[5] Coello, C.A.C.; Becerra, R.L. (2002) "Constrained optimization using an evolutionary programming-based cultural algorithm". *Adaptive Computing in Design and Manufacture* V, Springer, London: 317–328.

[6] Coello, C.A.C.; Becerra, R.L. (2003) "Evolutionary multiobjective optimization using a cultural algorithm", *Swarm Intelligence Symposium*, 2003. SIS-03. Proceedings of the 2003 IEEE: 6–13.

[7] Cope, D. (2000) *The Algorithmic Composer.* A-R Editions Inc. Wisconsin USA.

[8] Cope, D. (2005) *Computer Model of Musical Creativity.* MIT Press, Cambridge MA.

[9] Cruz Cortés, N. (2004) *Sistema Inmune Artificial para Solucionar Problemas de Optimización.* Doctoral Thesis, CINVESTAV-Instituto Politécnico Nacional. Ciudad de México.

[10] de Bono, E. (1991) *El Pensamiento Lateral: Manual de Creatividad.* Espsa libros S.L.U.

[11] Dorigo, M.; Maniezzo, V.; Colorni, A. (1996) "Ant system: optimization by a colony of cooperating agents", *IEEE Transactions on Systems, Man, and Cybernetics* **26**: 29–41.

[12] Geem, Z.W. (2009) *Recent Advances in Harmony Search Algorithm.* Springer, New York.

[13] Geem, Z.W. (2010) *Music-Inspired Harmony Search Algorithm.* Springer, New York.

[14] Geem, Z.W., Kim J.-H., Loganathan G.V. (2001) "A new heuristic optimization algorithm: harmony search", *Simulation* **76**(2): 60–68.

[15] Gessler, N. (2003) "Evolving cultural things-that-think", *Computational Synthesis: From Basic Building Blocks to High Level Functionality*, Papers from the 2003 AAAI Spring Symposium, Technical Report SS–03–02. Menlo Park, AAAI Press: 75–81.

[16] Gould N., Orban D., Toint P.L. (2001) " CUTTEr a Constrained and Unconstrained Testing Environment, revisited", in: http://www.cuter.rl.ac.uk/problems.html, consulted on 20/05/13.

[17] Horner, A.; Goldberg, D.E. (1991) "Genetic algorithms and computer assisted music composition", *Music Composition. ICMC9́1 Proceedings*, International Computer Music Association, San Francisco: 479–482.

[18] Hu, X.; Eberhart, R.(2002) "Solving constrained nonlinear optimization problems with particle swarm optimization", in: *6th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2002)*: 203–206.

[19] Jacob B.L. (1996) "Algorithmic composition as a model of creativity", *Organised Sound*, Cambridge University Press: 157–165.

[20] Jacob B.L. (1995) "Composing with genetic algorithms", *International Computer Music Association*: 452–455.

[21] Jin, X.; Reynolds, R.G. (1999) "Using knowledge based evolutionary computation to solve nonlinear constraint optimization problems: a cultural algorithm approach", *Proceedings of the 1999 Congress on Evolutionary Computation* **3**: 1678.

[22] Kennedy, J., Eberhart, R.C. (1995) "Particle swarm optimization", *International Conference Neuronal Networks*: 1942–1948.

[23] Koziel, S.; Michalewicz, Z. (1999) “Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization”, *Evolutionary Computation* **7**: 19–44.

[24] Landa Becerra, R.; Coello Coello, C.A. (2005) “Optimization with constraints using a cultured differential evolution approach”, in: *Proceedings of the GECCO Conference*: 34–35.

[25] Lee, K.S.; Geem, Z.W. (2004) “A new structural optimization method based on the harmony search algorithm”, *Computers & Structures* **82**: 781–798.

[26] Leyffer, S.; Mahajan, A.; Cochran, J.J.; Cox, L.A.; Keskinocak, P.; Kharoufeh, J.P.; Smith J.C. (2010) *Software for Nonlinearly Constrained Optimization.* John Wiley & Sons, New York.

[27] Liu Y.T. (2000) “Creativity or novelty?: Cognitive-computational versus social-cultural”, *Design Studies*: 261–276.

[28] McDonnell, J.; Reynolds, R.; Fogel, D.(1995) “Using cultural algorithms for constraint handling in GENOCOP”, *Evolutionary Programming IV: Proceedings of the Fourth Annual Conference on Evolutionary Programming*, MIT Press: 289–305.

[29] Michalewicz, Z. (1995) “Genetic algorithms, numerical optimization, and constraints”, *Proceedings of the 6th International Conference on Genetic Algorithms*: 151–158.

[30] Michalewicz, Z.; Janikow, C.Z. (1996) “Genocop: a genetic algorithm for numerical optimization problems with linear constraints”. *Communications of the ACM - Electronic supplement to the December issue* **39**, article 175.

[31] Michalewicz, Z.; Fogel, D.B. (1998) *How to Solve it: Modern Heuristics.* Springer, New York.

[32] Michalewicz, Z.; Deb, K.; Schmidtz, M.; Stidsenx, T. (2000) “Test-case generator for nonlinear continuous parameter optimization techniques”, *IEEE Transactions on Evolutionary Computation* **4**: 197–215.

[33] Mora-Gutiérrez, R.; Ramírez-Rodríguez, J.; Rincón-García, E.(2012) “ An optimization algorithm inspired by musical composition”, *Artificial Intelligence Review.* Springer Netherlands: 1–15 .

[34] Mora-Gutiérrez, R.; Ramírez-Rodríguez, J.; Rincón-García, E.; Ponsich, A.; Herrera O. (2012) "An optimization algorithm inspired by social creativity systems", *Computing* **94**(11): 887–914.

[35] Mora-Gutiérrez, R. (2013) *Diseño y desarrollo de un método heurístico basado en un sitema socio-cultural de creatividad para la resolución de problemas de optimización continuos no lineales y diseños de zonas electorales.* Thesis Ph. D. Universidad Nacional Autónoma de México. Ciudad de México.

[36] Ray, T.; Kang, T.; Chye, S.K. (2000) "An evolutionary algorithm for constrained optimization", *GECCO-00*: 771–777.

[37] Ray, T.; Liew, K.M. (2003) "Society and civilitation: an optimization algorithm based on the simulation of social behavior". *IEEE Transaction on evolutionary computation* **7**: 386–396.

[38] Reynolds, R.G. (1994) "An introduction to cultural algorithms". *Proceedings of the 3rd Annual Conference on Evolutionary Programming*, World Scientific Publishing: 131–139.

[39] Tang, W.; Li, Y. (2008) "Constrained optimization using triple spaces cultured genetic algorithm". *International Conference on Natural Computation* **6**: 589–593.