Dear author,

In the stage of preparing your full work for evaluation, after receiving approval of the corresponding abstract, we kindly ask you to follow the next steps:

1. Update your approved abstract (you might want to introduce modifications) and submit it now in a written version according to the abstract template (many abstracts were introduced directly into the websystem). This version of the abstract will compose the abstracts booklet to be distributed to the participants when registering;

2. Fill the form in the next page, indicating to which section you would like your work to be directed. This will help organizing the sessions according to the mini-symposia advertised rather than in the areas and subareas present in the websystem. We apologize for this possible confusion. The reason is that we are using the websystem provided by ABCM, which does not allow us to introduce thematic sessions freely.

If any doubts arise, please contact us. Your presence in MecSol 2017 will help making this a very successful event. We look forward to seeing you in Joinville!

With kind regards,

The Organizing Committee
MecSol 2017

6 th International Symposium on Solid Mechanics

Select the box to select the Mini-Symposia:

[ ] Multiaxial and Fretting Fatigue (Edgar Mamiya and Lucival Malcher);

[ ] Composite Materials (Ricardo de Medeiros);

[ ] Constitutive Models I - Plasticity and Damage (Lucival Malcher, Miguel Vaz Jr. and Edgar Mamiya);

[ ] Constitutive Models II - Viscoelastic and Viscoplastic Solids: Models and Applications (Heraldo Mattos);

[ ] Impact Engineering (Marcílio Alves);

[ ] Structural Reliability Methods and Reliability-Based Design Optimization (André Beck);

[ ] Topology Optimization of Multifunctional Materials, Fluids and Structures (Eduardo Lenz Cardoso e Emilio Carlos Nelli Silva);

[ ] Topological Derivatives: Theory and Applications (André Novotny);

[ ] X-FEM, G-FEM and Meshfree Methods (Roberto Dalledone Machado and Rodrigo Rossi);

[ ] High Order Finite Elements (Marco Lúcio Bittencourt);

[ ] Nonlinear Analyses (Eduardo Campello);

[X] Other section;

MECSOL Joinville 2017
April 26-28, 2017
Joinville, Santa Catarina, Brazil

Investigation of the diversity index trend of the Search Group Algorithm in comparison with others meta-heuristic algorithms

Rafael Rodrigues de Souza
CORE - Center Optimization and Reliability in Engineering - Department of Civil Engineering
Federal University of Santa Catarina
souza.rafael.rodrigues@gmail.com

Leandro Fleck Fadel Miguel
CORE - Center Optimization and Reliability in Engineering - Department of Civil Engineering
Federal University of Santa Catarina
leandro.miguel@ufsc.br

Matheu Silva Gonçalves
CORE - Center Optimization and Reliability in Engineering - Department of Civil Engineering
Federal University of Santa Catarina
matheusgoncalves.contato@gmail.com

Rafael Holdorf Lopez
CORE - Center Optimization and Reliability in Engineering - Department of Civil Engineering
Federal University of Santa Catarina
rafael.holdorf@ufsc.br

ABSTRACT

Several meta-heuristic optimization algorithms have been developed and tested over the last few decades. The development of these algorithms continues to be an active field of research, because of their ability to solve real life complex problems. However, one of the issues that meta-heuristic algorithms present is that their efficiency is problem dependent, i.e. the performance of a given algorithm solving a certain class of problems does not guarantee its good performance for other problems. Thus, the performance of a meta-heuristic algorithm may only be assessed *a-posteriori*, e.g. analyzing the best result, mean value, standard deviation and convergence curve of several independent runs. In the context of meta-heuristic algorithms, it is well-recognized that they should have two capabilities, exploration and exploitation, in order to be able to find reasonable solutions. It is important for meta-heuristic algorithms to maintain an adequate balance between the exploration and exploitation tendencies in order to be competitive in terms of robustness and performance. Kaveh and Zolghadr (2014) present the so called diversity index, which is a practical approach to analyze these features. One of the conclusions of the mentioned study is that the diversity index plays a major role in understanding the behavior of meta-heuristic algorithms. Indeed, it is indicated that satisfactory results are related to a specific behavior of the diversity index. Thus, the main goal of this paper is to show that using the Search Group Algorithm (SGA) (Gonçalves et. al (2015)), the designer is able to choose *a-priori* the diversity index trend and consequently a good balance between exploration and exploitation. In contrast to other meta-heuristic algorithms, where the correlation between the parameters and the diversity index trend is not clear. To achieve this goal, several benchmark optimization problems are solved, employing the SGA, the Firefly Algorithm and the Backtrack Search Algorithm.

Keywords: Optimization algorithm, Search Group Algorithm, Diversity index, meta-heuristic algorithms.

1. INTRODUCTION

In the last decades, several meta-heuristic optimization algorithms have been developed and tested: Simulated annealing [1], [2]; Particle swarm optimization [3]; Harmony search [4]; Ant colony [5]; Imperialist competitive algorithm [6]; Mine Blast algorithm [7]; Big Bang-Big Crunch [8]; Bat-inspired algorithm [9]; Dolphin echolocation algorithm [10]; Teaching-learning-based algorithm [11]; Chaotic swarming of particles [12]; Colliding bodies optimization [13]; Firefly Algorithm [14],; Charged system search [15]; Cuckoo Search [16], [17]; Enhanced ray optimization [18]; to name just a few.

The development of these algorithms continues to be an active field of research because of their ability of solve real life complex problems. For example, in the case of structural design, usually non-linear, non-convex and mixed variables, i.e. the presence of both discrete and continuous design variables, problems. However, one of the issues that meta-heuristic algorithms present is that their efficiency is problem dependent, i.e. the performance of a given algorithm solving a certain class of problems does not guarantee its good performance for others problems.

Due to the previously mentioned aspect, the performance of a meta-heuristic algorithm may only be assessed *a-posteriori*, e.g. analyzing the best result, mean value, standard deviation and convergence curve of several independent runs. In other words, it is not possible to pursue a formal mathematical convergence analysis, such as in gradient based algorithms for convex problems. Indeed, it is the price to pay for the ability of meta-heuristic algorithms to provide solutions for problems that do not posses the required assumptions for such mathematical analysis.

In the context of meta-heuristic algorithms, it is well-recognized that they should have two capabilities, exploration and exploitation, in order to be able to find reasonable solutions. Exploration may be described as the ability of the algorithm to find promising regions on the design domain, i.e. regions in which the optimal solution may be located. Exploitation is the ability of the algorithm to refine the solution on these promising regions, i.e. to pursue a local search on them. It is important for a meta-heuristic algorithm maintaining an adequate balance between the exploration and exploitation tendencies in order to be competitive in terms of robustness and performance. An practical approach to analyze these features is through the diversity in the population, which may be evaluated by diversity indexes, e.g. the one presented by [19] and modified in [20].

In [20], the authors solved truss optimization problems with frequency constraints, comparing nine meta-heuristic algorithms. In their analysis, the diversity index was employed to inspect the exploration/exploitation capabilities of each algorithm. One of their conclusions was that the diversity index plays a major role in understanding the behavior of meta-heuristic algorithms. Indeed, this study suggested that for a good balance between exploration and exploitation the most desirable trend of variation of the diversity of the population is the one in which it has high values in the first iterations of the search to explore the design domain, and then, the diversity should gradually decrease as the iterations pass by to exploit promising regions where the global solution may be located (see Section 2for further details). This conclusion was also supported by the results of the Search Group algorithm (SGA) [21], which is a recently developed meta-heuristic for the optimization of truss structures.

From the papers and results cited in the previous paragraph, one could conclude that if one were able to set *a-priori* the diversity index trend of a given algorithm, one would increase the chance of obtaining good results. However, as the performance of meta-heuristic algorithms, their diversity index trend is only know *a-posteriori*. Indeed, meta-heuristic algorithms often have several parameters, and usually, different set of parameters may lead to different behaviors of the diversity index, and also, different performances. In this context, the main goals of this paper are:

- to show that, by using the SGA, the designer is able to choose *a-priori* the diversity index trend, i.e., such a trend is problem independent for the SGA.
- to show that the diversity index trend is problem dependent, even for the same set of parameters of a given algorithm;

The advantage of the first goal is that by using the SGA, the designer may set *a-priori* the diversity index trend of the search, and consequently, a good balance between exploration and exploitation. In order to accomplish these goals, three benchmark problems are solved, employing the SGA, the FA (Firefly algorithm [14]) and the BSA (Backtrack Search Algorithm [22])

The rest of the paper is organized as follows: section 2.presents an explanation and a discussion regarding the diversity index, in Section 3.the optimization algorithms studied in this paper are briefly described, in section 4.the numerical examples are presented. Finally, the main conclusions drawn from this work are summarized in Section 5.

2. Diversity index

The diversity index studied in this paper was first introduced by [19]. According to these authors this index can be viewed as the ratio of the search space which is covered by the algorithm at each iteration, thus representing with a significant clarity the exploration/exploitation features of the algorithm. The Diversity index formulation employed is the one presented in [20], shown in equation 1.

$$DI = \frac{1}{n_{pop}} \sum_{i=1}^{n_{pop}} \sqrt{\sum_{j=1}^{dim} \left(\frac{P_{1j} - P_{ij}}{x_j^{max} - x_j^{min}}\right)^2} \tag{1}$$

where P_{1j} is the jth component of the best design in the current population, x_j^{max} and x_j^{min} are the upper and lower bounds of the jth design variable, dim is the number of variables on each design vector and n_{pop} is the number of design vectors on each iteration (i.e the size of the population).

In [20], the authors demonstrate a significant relation between the Diversity Index trend and the efficiency of the algorithm, as already mentioned. Thus, an optimization algorithm demands a satisfactory balance among exploration and exploitation, i.e. the ability of seek the most promising areas of the search space, avoiding local minimum traps. The Diversity index allows a better understanding of this process, clearly showing if the algorithm is on a exploration or exploitation phase.

Based on the results presented on the mentioned study, the diversity index trends can be roughly divided into three groups, shown on Figure 1. It is possible to observe that the Trend 1 maintain a relatively high distance from the horizontal axis at the final iterations. Which means that an algorithm with this diversity index trend, will not perform the local phase (i.e. exploitation) properly. On the other hand, an algorithm with diversity index similar to Trend 3, will reach the local search phase to early. Consequently, the exploration phase will be neglected, rushing the algorithm to be trapped into a local minimum. Additionally, when the diversity trend achieve the horizontal axis means that the search has stopped, i.e. on this stage the algorithm generates identical individuals. The results presented by [20] showed that the tested algorithms with diversity index trend similar to Trends 1 and 3 (on Figure 1) figured among the worst results for the studied problems. Still on Figure 1, the Trend 2 represents the region where theres a desirable balance between the exploration and exploitation feature. The diversity index trends, on the algorithms tested by [20], similar to Trend 2 figured among the best results for the studied problems.Thus, It is possible to infer Trend 2 as a desirable region for the diversity index trend.

[21], which tested the SGA performance through several benchmark problems, also demonstrated the correlation between the diversity index trend and the quality of the results (minimum value, mean value and standard deviation). Since, the SGA was able to achieve better results for all the studied problems and presented a diversity index trend similar to the Trend 2, on Figure 1.

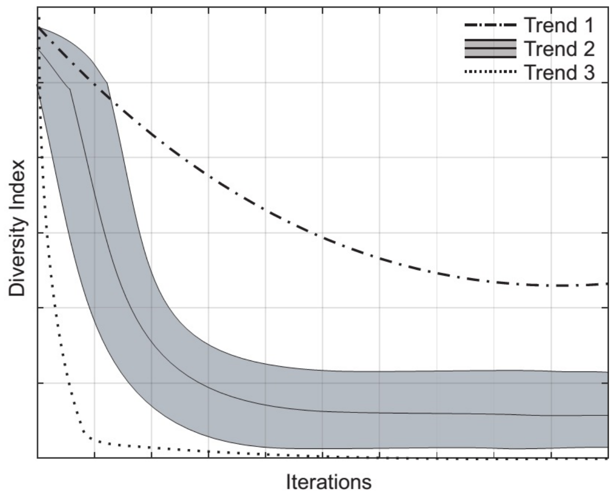

Figure 1. Diversity index trends

3. Meta-heuristic algorithms

3.1 Firefly algorithm - FA

Based on the natural behavior of fireflies, the FA [14] has shown promising results for structural optimization problems [23] and [24]. In this algorithm each possible solution is consider as a firefly and the group of fireflies at each iteration composes a population. The basic idea is that during the procedure each individual of the population (i.e. each firefly) move in the domain, seeking better objective function values. The natural behavior of the fireflies is adapted in a numerical methodology through the following rules, [14]:

- all fireflies are unisex, so the attractiveness between them does not depend on their sex;
- the attractiveness is proportional to its brightness, in other words, between two fireflies the less bright will move toward the brighter one. The attractiveness and brightness decrease with increasing distance between the fireflies. Also, if there is no brighter firefly, it will move randomly;
- the brightness of a firefly is affected or determined by the shape of the objective function.

Firstly, the initial population of fireflies $\mathbf{P}$ is randomly generated. The dimension of the population matrix $\mathbf{P}$ is $n_{pop} \times dim$. Where n_{pop} is the number of fireflies, thus the size of the population and dim is the number of design variables. Therefor, each firefly P_i is a set of design variables which represents a possible solution.

The most important properties of FA are the variation of the light intensity and the formulation of the attractiveness. The brightness I of a firefly is proportional and governed by the objective function, $I(P_i)\ \alpha\ f(P_i)$, although the brightness of one firefly P_a seen by another one P_b is inversely proportional to the distance between fireflies a and b, considering γ. Thus, it can be written as:

$$I(r) = I_0\, e^{-\gamma r_{ab}^2} \quad (2)$$

where γ is the absorption coefficient of the light intensity and r_{ab} is the relative distance between two fireflies. Since the attractiveness of a firefly is proportional to its light intensity seen by an adjacent firefly, the attractiveness β can be written as:

$$\beta(r) = \beta_0\, e^{-\gamma r_{ab}^2} \quad (3)$$

where, β_0 is the attractiveness at the $r = 0$ position, thus it is possible to define the movement of a firefly as:

$$P_a^{k+1} = P_a^k + \beta_0\, e^{-\gamma r_{ab}^2} (P_a^k - P_b^k) + \alpha\varepsilon \quad (4)$$

The main operation of the FA is the movement of each individual, which is defined by equation 4. Thus, a firefly a is compared against another firefly b. If $I_a < I_b$ then a will move toward b, otherwise a stays in the same position. Considering $a = 1,2,\ldots,n_{pop}$ and $b = 1,2,\ldots,n_{pop}$, each firefly is compared with all the other in the population, defining their new position at iteration $K+1$.

The figure 1 presents the pseudo code of the Firefly algorithm, illustrating its steps.

Table 1. Pseudocode of FA

Objective function $f(P_i)$
Generate the initial population of fireflies $P_i (i = 1,2,\ldots,n_{pop})$
Light intensity I_i *at* P_i *is determined by* $f(P_i)$
Define light absorption coefficient γ
While (k <MaxGeneration)
  for $a = 1 : n_{pop}$
    for $b = 1 : n_{pop}$
      if $I_a < I_b$, *move firefly toward b in d-dimension*; ***end if***
      Attractiveness varies with distance r via $exp[-\gamma r]$
      Evaluate new solutions and update light intensity
    end for a
  end for b
Rank the fireflies and find the current best
end while
Post process results and visualization

3.2 Search group algorithm - SGA

Developed by [21], the basic idea of the SGA is to have a good balance between the exploration and exploitation. During the optimization the generation of new individuals is pursued only by a few members of the population, which form the search group. Additionally, a mutation operator is employed to generate new designs away from the ones of the current search group. The SGA is divided in five steps: the initial population; initial search group selection; mutation of the search group; generation of the families and selection of the new search group. A briefly description of each step is presented in the following subsections.

3.2.1 Initial population

The initial population **P** is randomly generated:

$$P_{ij} = x_j^{min} + (x_j^{max} - x_j^{min})\ U[0,1], \quad i = 1 \text{ to } n_{pop},\ \ and\ \ j = 1 \text{ to } dim \tag{5}$$

where $U[0,1]$ is a uniform random variable which ranges from 0 to 1, x_j^{min} and x_j^{max} are the lower and upper bounds of the jth design variable, respectively, n is the number of design variables and n_{pop} is the size of the population.

3.2.2 Initial search group selection

Once the initial population **P** is generated, the search group **R** is constructed by selecting n_g individuals from **P**. For this selection a standard tournament is applied on the ranked **P**. n_g is defined by the user as a percentage (DG %) of the population.

3.2.3 Mutation of the search group

For this step, the search group **R** is mutated at each iteration.This mutation strategy consist in replacing n_{mut} individuals from **R** by new individuals generated based on the statistics of the current search group. The mutated individuals are generated according to the equation 6.

$$x_j^{mut} = E[R_{:j}] + t\,\varepsilon\,\sigma\,[R_{:j}], \text{ for } j = 1,\ldots,n \tag{6}$$

where x_j^{mut} is the jth design variable of a given mutated individual, E and σ are the mean value and standard deviation operators, ε is a convenient random variable, t is a parameter that controls how far the new individual is generated, and $\mathbf{R}_{:j}$ is the jth column of the search group matrix. In order to select the individuals from **R** to be mutated, an inverse tournament is employed. Thus the worse individuals are more likely to suffer the mutation.

3.2.4 Generation of the families of each search group member

Once the search group is formed, each of its members generates a family (named $\mathbf{F}_i$) by the perturbation described in the equation 7:

$$x_j^{new} = R_{ij} + \alpha_j\,\varepsilon, \ \text{ for } j = 1,\ldots,n \tag{7}$$

where α controls the size of the perturbation for the global and local phases. At each iteration k, the parameter α is reduced, as described on equations 8 to 10, in order to achieve a good balance between the exploration and exploitation features.

$$\alpha_j = \alpha_{perturbation}(x_j^{max} - x_j^{min}) \tag{8}$$

$$\alpha_{perturbation}^{global} = (\alpha_0\beta^{global} + \alpha_{min}) \tag{9}$$

$$\alpha_{perturbation}^{local} = (\alpha_{min}\beta^{local} + r_{min}\alpha_{min}) \tag{10}$$

where β^{global},β^{local} and r_{min} are parameters of the SGA. α_{min} is the minimum value which α may assume, while α_0 is the initial value of α

The number of individuals that each member of the search group generates is given as an input vector $\boldsymbol{v} = [v_1, \ldots, v_{ng}]$. v must follow two rules:(i) be formed by decreasing values so that the members of the search group with better objective function values generate more individuals; (ii) the sum of v plus the number of members on the search group must result on the population size (n_{pop}).

3.2.5 Selection of the new search group

The SGA iterations are divided in two phases: global and local. The ratio of iterations employed on each phase is defined as a parameter. In the global phase, the new search group is selected, choosing the best individuals among each family. Nevertheless, in the local phase, the new search group is formed by the best individuals among all families, forcing the algorithm to pursue the exploitation feature.

The number of iterations employed in the global phase (it^{max}_{global}), is a percentage of the total iterations (it^{max}), defined by the parameter Global *it* %.

3.2.6 Step by step of SGA

The SGA step by step can br written as [21]:

1. Initialize the parameters;

2. Generate the initial population **P**;

3. Creates the initial search group R^k, selecting n_g individuals from the initial population using a tournament selection;

4. Replace n_{mut} individuals by new members, created as described in equation 6;

5. Generate the families F_i using equation 7;

6. Select new search group according to the rule:

   (a) if $k < it^{global}_{max}$: Search group R^{k+1} is formed by the best member of each family;

   (b) Else: Search group R^{k+1} is formed by the best n_g individuals of the population;

7. Update α^{k+1} through 8;

8. Make $k = k+1$, if $k \geq it^{max}$, go to step 9, otherwise return to step 4;

9. Solution found: $X^* = R_{1,:}$

3.3 Backtracking Search Algorithm - BSA

The BSA is a multi-agent based evolutionary algorithm, developed by [22]. A general over view of the BSA is illustrated in Table 1 and the details of each step are presented in the next subsections. The BSA description shown here is different from the original description given by [22]. The authors hope that the description given here is more direct and easy to understand.

Table 2. Pseudoconde of BSA

1. Initialization
Do
Generation of the Perturbed/Trial Population
2. Evaluation of "direction/length" of the population
3. Perturbation of the current population
end
4. Selection of the new population
Until *convergence criteria are met*

3.3.1 Initialization

The initial population of the BSA is generated as:

$$P_{ij} \sim U(x_j^{min}, x_j^{max}) \tag{11}$$

where U is a uniform random variable, x_j^{min} and x_j^{max} are the lower and upper bounds of the j^{th} design variable $i = 1, \ldots, n_{pop}$ and $j = 1, \ldots, dim$ where n_{pop} represents the size of the population and dim the dimension of the problem. Thus, each row of $\mathbf{P}$ represents an individual of the population and each column represents a design variable. After the construction of the initial population the iterative process of the algorithm is initiated by generation the trial population and updating them until some convergence criterion is achieved.

3.3.2 Contruction of the trial or perturbed population $\mathbf{P}_{pert}$

The first step in the construction of the perturbed population $\mathbf{P}_{pert}$ (or trial population) is the evaluation of the "direction" od the perturbation that will be applied to the current population. Such a direction is calculated with the aid of the historical population $\mathbf{P}_{old}$. There are two possible cases for the evaluation of $\mathbf{P}_{old}$, each with a 50% of chance of happening. In case 1, $\mathbf{P}_{old}$ is generated by a random permutation of the lines of the current population, while in case 2, it is randomly generated just as the initial population. This procedure is illustrated is Table 3, in which a and b are random constants following a uniform ditribuition between zero and one, and $:=$ is an update operator. The author that proposed the BSA claimed that it has a memory from past iterations. Actually, this memory is due the the construction of $\mathbf{P}_{old}$ using case 1.

Table 3. Two cases employed for the construction of $\mathbf{P}_{old}$

if $a < b \| a, b \sim U(0,1)$	
then	| **Case 1:** $\mathbf{P}_{old}$ is the random permutation of
$\mathbf{P}_{old} := \mathbf{P}$	| the lines of the current $\mathbf{P}$
$\mathbf{P}_{old} := randperm(\mathbf{P}_{old})$	
else	| **Case 2:** $\mathbf{P}_{old}$ is randomly restarted
$(\mathbf{P}_{old}) \sim U(x_j^{min}, x_j^{max})$	
end	

With $\mathbf{P}_{old}$ defined, the perturbed population is evaluated as:

$$\mathbf{P}_{pert} = \mathbf{P} + \mathbf{M}. * [\alpha(\mathbf{P}_{old} - \mathbf{P})] \tag{12}$$

in which the operator $.*$ holds for multiplication term by term and α is a random parameter (called scale factor) that controls the amplitude of the search-direction matrix ($\mathbf{P}_{old} - \mathbf{P}$). The purpose of the matrix $\mathbf{M}$ in Equation 12 is to define which terms of P are perturbed by $\alpha(\mathbf{P}_{old} - \mathbf{P})$ to generate the perturbed or trial population. Thus, the matrix $\mathbf{M}$ is composed by zeros and ones, where each term M_{ij} of $\mathbf{M}$ equal to one, means that the corresponding term P_{ij} of $\mathbf{P}$ will be perturbed for the construction of the perturbed population $\mathbf{P}_{pert}$.

Initially $\mathbf{M}$ is set as $t_{pop}\ x\ n_v$ zero matrix, and for the rest of its construction two cases may occur at each iteration of the algorithm, each with a 50% chance of happening. In the first case, the parameter *mixrate* (m_r) chooses randomly the elements of each line of $\mathbf{M}$ to assume the unit value. In the second case, only one term of each line is randomly chosen to be equal to 1. The process just described for the construction of $\mathbf{M}$ is also illustrated in Table 4, in which $randi(n_v)$ is a discrete uniform random value between one and n_v. As a result of the perturbation process, some individual of the perturbed population may extrapolate the boundaries of the design domain. Thus, at the end of this step, the individuals beyond the search-limits are randomly generated in the admissible design domain.

Table 4. Two cases employed for the construction of M

$\mathbf{M} = zeros(t_{pop,n_v})$	
if $a < b \vert a,b \sim U(0,1)$ **then**	
for$i = 1 : t_{pop}$	| **Case 1:** up to m_rUn_v are perturbed
$\mathbf{M}_{i,u_{(1:[m_rUn_v])}} = 1 \vert u = randperm(1,\ldots,n_v)$	
end	
else	
for$i = 1 : t_{pop}$	
$\mathbf{M}_{i,u_{randi_{(nv)}}} = 1$	| **Case 2:** only one term is perturbed
end	
end	

3.3.3 Selection of the new population

In this step, the fitness value (J) of each individual of the perturbed population $\mathbf{P}_{pert}$ is evaluated. Then, the algorithm compares the fitness value of the i^{th} individual $(P_{pert})_i$ of the perturbed population to the i^{th} individual $(P)_i$ of $\mathbf{P}$. If the fitness value of $(P_{pert})_i$ is better than the one of $(P)_i$, the latter is replaced by the former in the new population of the algorithm.

3.3.4 Further coments

In *t*h original description of the BSA, the author claimed that this algorithm had only one parameter to be adjusted, the *mixrate* (m_r). It is only possible to have this one parameter due to the following aspects: (i) fixing the probability of occurrence of cases 1 and 2 in the construction of bo*th* $\mathbf{P}_{old}$ and $\mathbf{M}$, and (ii) fixing the length of the perturbation, i.e. the value of α. Of course, these parameters may also be adjusted accordingly to the designer needs.

From sections 3.3.2and 3.3.3one may see that there is no direct interaction among the individuals in the BSA. This fact is the reason we described the generation of the trial population as a perturbation of the current one, not using the mutation and cross over operators as described in the original paper of the BSA.

4. Numerical examples

In order to evaluate the variation of the diversity index, the algorithms were tested with different sets of parameters. For the SGA, the parameter varied was the decreasing of $\alpha_{perturbation}$, which controls the size of perturbation during the generation of new individuals. Thus, has a significant impact on the diversity index.

Four scenarios were tested for the SGA:

1. Set 1: standard configuration of the algorithm. $\alpha_{perturbation}$ is described by equations 9 and 10. β^{global} is defined by equations 13 and 14.

$$\beta^{global} = \max[1 - \frac{4k}{it_{max}^{global}} \quad | \quad 0.25 - \frac{0.25k}{it_{max}^{global}}] \tag{13}$$

$$\beta^{local} = \frac{(it_{max} - it_{max}^{global}) - k}{(it_{max} - it_{max}^{global})} \tag{14}$$

2. Set 2: for this configuration the β parameter was define in order to simulate an exponential behavior. For global and local phase, β (defined by equation 15) and $\alpha_{perturbation}$ (defined by equation 9) feature the same formulation.

$$\beta^{global} = \beta^{local} = 2.4e^{-0.9k} \tag{15}$$

3. Set 3: on this set the main goal is to achieve a linear decreasing. Thus, the parameter formulation is written as equation 16:

$$\beta^{global} = \beta^{local} = 1 - \frac{0.9k}{it_{max}^{global}} \tag{16}$$

4. Set 4: the final set is defined by equation 17:

$$\beta^{global} = \beta^{local} = 1 - 8.16 \cdot 10^{-9} x^3 \tag{17}$$

The other parameters o the SGA were kept unchanged, with values shown on table 5.

Note that for sets 2 to 4 the $\alpha_{perturbation}$ parameter follows the same formulation (defined in equation 9) for the global and local phases. Additionally, for all sets it^{max} and n_{pop} are defined as 500 and 100, respectively.

For the FA, in a rough analysis of equation 4 it is noticed that variations on the three parameters (α,β_0,γ) defines the size of the movement of a firefly from its initial position. Thus, controls the perturbation of the generation of new individuals on a population and consequently the diversity index. Thus, these parameters are varied in four sets, presented in table 6. The other parameters it^{max} and n_{pop} were set as 500 and 100, respectively.

For the Backtrack Search Algorithm the parameter α on Equation 12 and the *mixrate* parameter (on Section 3.3.2) were chosen to be varied. α controls the amplitude of the perturbation, while, *mixrate* controls how many j*th* variables, of each individuals P_i of a population $\mathbf{P}$, will be perturbed. Since this two parameters control the perturbation, they have a significant influence on the diversity index. It is important to highlight, that the *mixrate* parameter is applied under a 50% chance.

In order to analyze the diversity index behavior on the BSA, four sets of parameters are tested:

Table 5. SGA Parameters

Parameter	Value
α_{min}	0.01
α_0	2
Global *it* %	90%
SG %	10%
n_{mut}	5
$\boldsymbol{v}$	[16 15 14 12 10 8 6 5 3 1]
it^{max}	500
n_{pop}	100
r_{min}	0.0025

Table 6. Four sets of parameters for the FA

Parameter	set 1	set 2	set 3	set 4
α	0.5	1.0	1.0	0.8
β_0	0.2	0.8	0.5	1.0
γ	1.0	0.5	0.9	0.1
it^{max}	500	500	500	500
n_{pop}	100	100	100	100

1. Set 1: α is defined by equation 18. Where *randn* is a random number with normal distribution, mean value 0 and standard deviation 1. *mixrate* is defined as 1.

$$\alpha = 3 * randn \tag{18}$$

2. Set 2: on this set, α is also defined by equation 18. While *mixrate* is defined as 0.5.

3. Set 3: for this case α is defined by equation 19. Where $lognrnd(\mu,\sigma)$ is a random value generated from a lognormal distribution. Considering, $\mu = rand$ and $\sigma = 5 * rand$, where rand is random number chosen from uniform distribution between 0 and 1. *mixrate* is equal to 1.

$$\alpha = lognrnd(\mu,\sigma) \tag{19}$$

4. Set 4: on this set α is also defined by equation 19. *mixrate* is 0.5.

In order to analyze the diversity index, three optimization problems are tested:

4.1 Function 1: Shekel's Foxholes function

This benchmark function was introduced by Shekel in 1971 and it is presented in figure 2. The Shekels Foxholes is a two-dimensional function, with 25 peaks and minimum located at (-32,-32), defined by equation 20. Where $a(i) = 16[(imod5) - 2]$ and $b(i) = 16[(\frac{1}{5} - 2)]$

$$Z(x,y) = 500 - \frac{1}{0.002 + \sum_{i=0}^{24} \frac{1}{1+i+(x-a(i))^6+(y-b(i))^6}} \tag{20}$$
$$-65.536 \leq x,y \geq 65.536$$

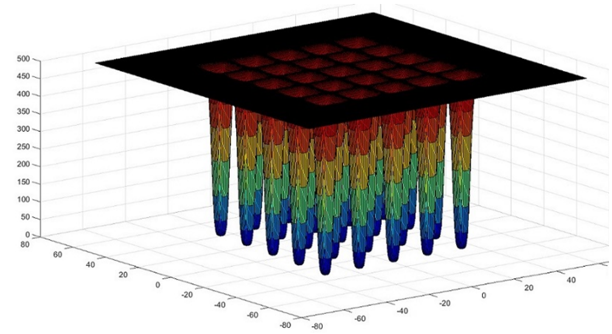

Figure 2. Shekel's Foxholes function

The results for the algorithms are illustrated in figures 3, 4 and 5. In order to achieve an graphic easier to understand, the diversity index presented are the mean values for 50 runs of each algorithm.

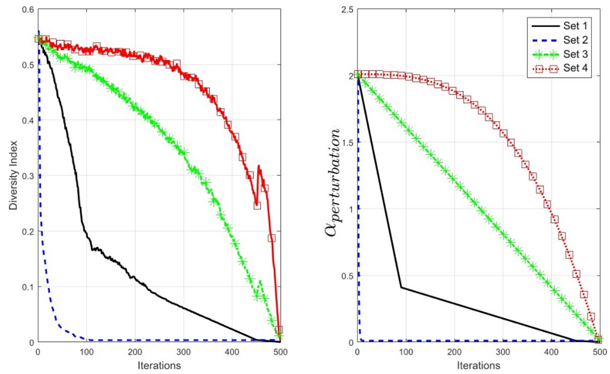

Figure 3. Diversity index and Alfa function for SGA on Shekel's Foxholes function

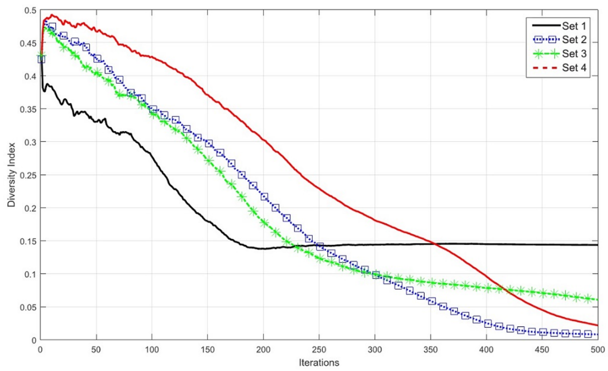

Figure 4. Diversity index for FA on Shekel's Foxholes function

The results show the SGA ability of control de diversity index variation, which follows closely the $\alpha_{perturbation}$ parameter. For the FA, the parameter γ is the most important one, since it is taken exponentially, equation 4. Sets 1 and 3 have the lower values of Diversity Index, since γ possess the higher values on these sets. The results for BSA, on Figure 5, show relatively similar trends for the four tested sets. This mostly related to low dimension of this problem, i.e. only two design variables. Additionally, all the sets are in the Trend 2 region of Figure 1.

4.2 Function 2: Shekel's Family m=7

This is a four-dimensional function, with formulation shown in equation 21 and table 7 The global minimum is located at (4,4,4,4), since the m parameter is set as 7, the global minimum value is

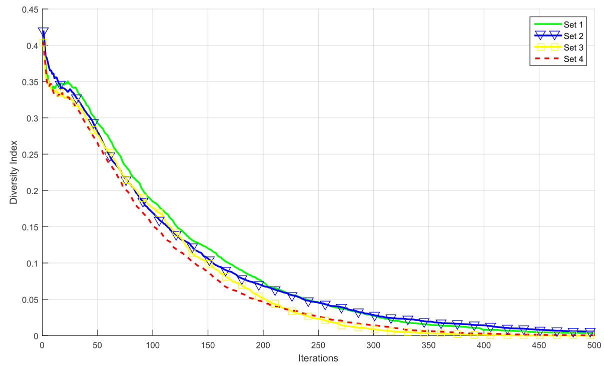

Figure 5. Diversity index for BSA on Shekel's Foxholes function

-10.4029.

$$Z(x,y) = -\sum_{i=1}^{7}\sum_{1}^{4}[(x_j - a_{ij})(x_j - a_{ij}) + c_j]^{-1} \tag{21}$$

$$-10 \le x_j \ge 10$$

Table 7. Shekel's Family m=7 Parameters

i	a_{ij},j=1,...,4				c_i
1	4	4	4	4	0.1
2	1	1	1	1	0.2
3	8	8	8	8	0.2
4	6	6	6	6	0.4
5	3	7	3	7	0.4
6	2	9	2	9	0.6
7	5	5	3	3	0.3
8	8	1	8	1	0.7
9	6	2	6	2	0.5
10	7	3.6	7	3.6	0.5

The results for the algorithms are presented in figures 6 and 7, 8. As well as in the previous function, the values of diversity index are the mean values of 50 runs.

Observing this example the SGA advantage is highlighted. Since, the Diversity Index variation follows the same trend (obeying the $\alpha_{perturbation}$ parameter) from the previously example. On the other hand, the Diversity Index variation for the FA is significantly different. On the FA, the understanding of the parameters allows the designer to enforce a diversity index. For example, a low value of γ (Sets 2 and 4) represents a high diversity index on both examples 1 and 2, however, the diversity index trend through out the iterations is still unpredictable and is different on each example. For the BSA, the behavior of the five sets are not longer similar to each other. Yet, it is possible to observe that sets 2 and 4 present a high and stagnated diversity index, that is mostly because the low value of *mixrate*, prevents the algorithm from properly perturb the populations. Thus, the diversity index remains the same, and the algorithm is not able to pursue the exploitation feature. Additionally, the diversity index trends for the same sets are significant different, in comparison with the previous example.

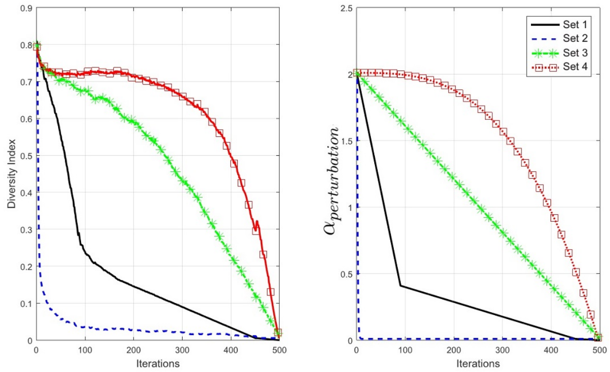

Figure 6. Diversity index and Alfa function for SGA on Shekel's Family m=7 function

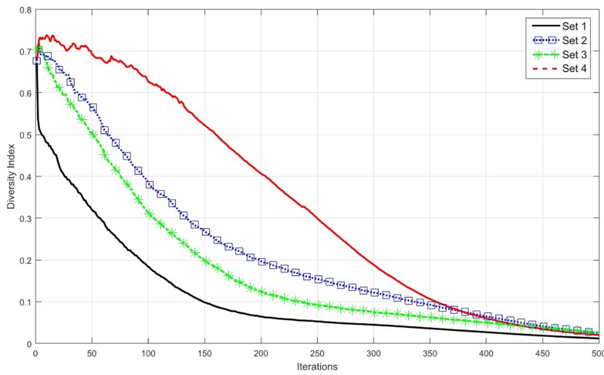

Figure 7. Diversity index for FA on Shekel's Family m=7 function

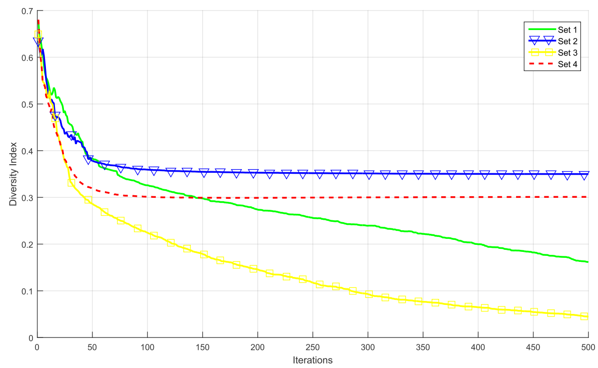

Figure 8. Diversity index for BSA on Shekel's Family m=7 function

4.3 Function 3: Eleven-bar truss example

This truss has been widely used as a benchmark problem by several researches. The ground structure and the applied load is presented in Figure 9. The design parameters are given in Table 8. In to order to analyze the diversity index, It is performed on this example the size and topology optimization.

During the optimization process, each of the 11 bars, can be chosen from the discrete set ω =(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 10.45159, 11.61288, 15.35481, 16.90319, 18.58061, 19.93544, 20.19351, 1.80641, 23.41931, 24.77414, 24.96769, 26.96769, 28.96768, 30.96768, 32.06445, 33.03219, 37.03218, 46.58055, 51.41925, 74.19340, 87.09660, 89.67724, 91.61272, 99.99980, 103.22560, 121.29008, 128.38684, 141.93520, 147.74164, 170.96740, 193.54800, 216.12860) cm^2. Note that the first 10 cross-section areas from the set ω are null values. Where a member addressed to a null area is elimi-

Table 8. Design parameters for the eleven-bar truss problem

Design Parameter	**Value**
Modulus of elasticity	68947.591 MPa
Weight density	2767.990 kg/m^3
Allowable stress in tension	172.369 MPa
Allowable stress in compression	172.369 MPa
Allowable y-displacement	50.8 mm

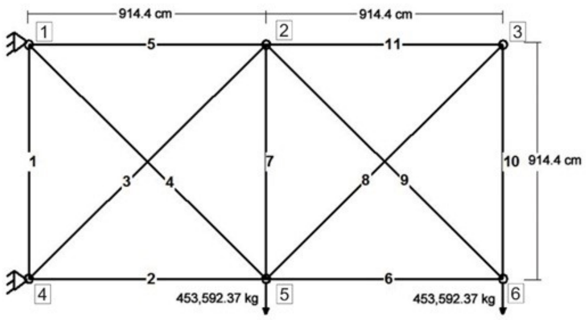

Figure 9. Eleven-bar truss ground structure

nated from the ground structure.

The results for the algorithms are presented in figures 10 and 11, 12. As well as in the previous function, the values of diversity index are the mean values of 50 runs.

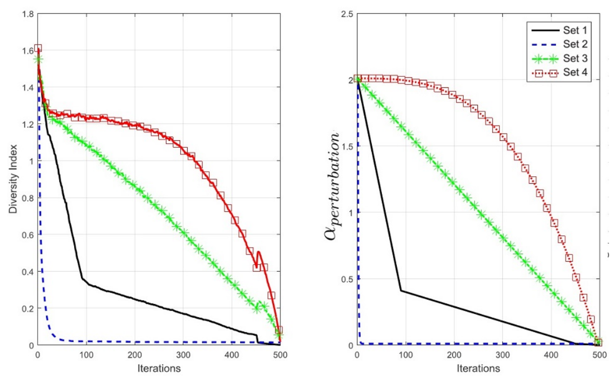

Figure 10. Diversity index and Alfa function for SGA on the eleven-bar problem

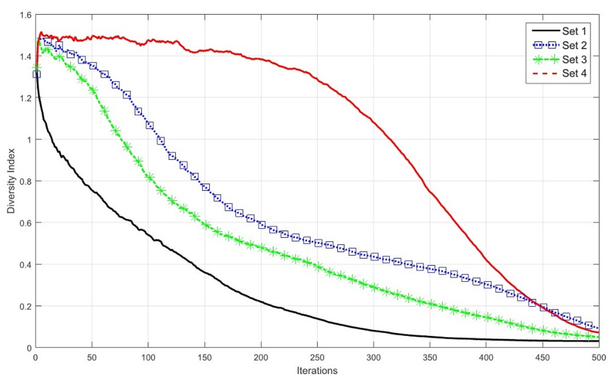

Figure 11. Diversity index for FA on the eleven-bar problem

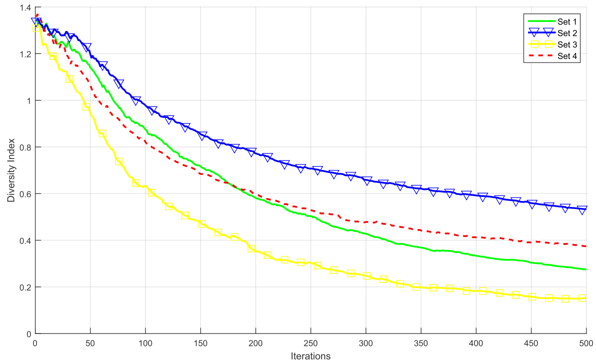

Figure 12. Diversity index for BSA on the eleven-bar problem

The results show a similar pattern, in comparison to the previous examples. The FA, continue to follow mainly the β parameter. However, the diversity index trend is different from the other examples. The diversity index trends for the SGA continue to follow the $\alpha_{perturbation}$ parameter. The results for the BSA show again similar trends for the tested sets,however, the actual behavior of the diversity index remain relatively unpredictable. Demonstrating its problem dependency. One can notice that, for numerical examples 2 and 3, the set 3 presented a diversity index trend close to the Trend 2 region on Figure 1.

Analyzing the results for the SGA (Figures 3, 8 and 12) It is possible to observe a small jump on the diversity index trends at the final iterations, specially on sets 3 and 4. This phenomena is due to the transition from the global to the local phase. On this transition, besides a possible variation on the $\alpha_{perturbation}$, the strategy of selection of the family leaders is changed. As explained on Section 3.2.5

This new strategy causes at the short-term a small increase of diversity. The main reason for this behavior is because, during the final iteration of the global phase, the families leaders are relatively stables. So, when the selection strategy changes, the family leaders also change. Since the new individuals on each population are generated from the family leaders, the following iterations will generate a new population that is slightly more diverse. Nevertheless, after few iteration on the local phase, the new selection strategy will choose family leaders which are very close to each other (consequently, generating similar individuals), allowing the algorithm to perform is exploitation feature.

The problem dependency of the diversity index trend can be further analyzed through Figures 13, 14 and 15. These figures show the trends for the three tested function for the same set of parameters. It possible to observe that, regarding using the same parameters, the algorithms perform different for each tested function. Additionally, one can notice that for the SGA (Figure 14) the trends are significant similar for the three function, especially compared with the FA (Figure 13) and BSA (Figure 15).

5. CONCLUSIONS

In this paper, it was presented a study of the diversity index of three meta-heuristic algorithms, the Search Group Algorithm, the Firefly algorithm and the Backtrack Search Algorithm. Three benchmark function were solved, in order to evaluate the diversity index variation on different problems.

Through the results it is possible to observe that, for the SGA, the $\alpha_{perturbation}$ has a major role on the diversity index variation. For each of the four sets, the index variation closely followed the decreasing of the $\alpha_{perturbation}$, for all numerical examples.

The results for the FA shown the tendencies defined by the parameters. For example, an increasing of β_0 results on an increasing of the diversity index, and a decreasing of α results on a increasing of the diversity index, in figures 4 and 7 the curve of Set 4 represents this tendency. Thus, although

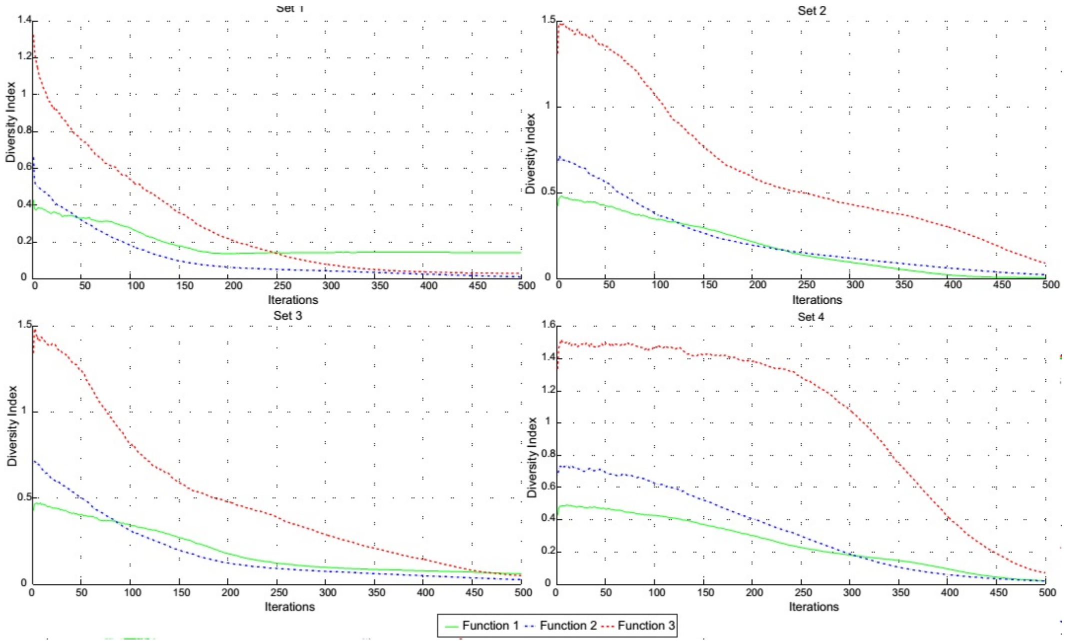

Figure 13. Diversity index employing the FA on the four sets considering each tested function

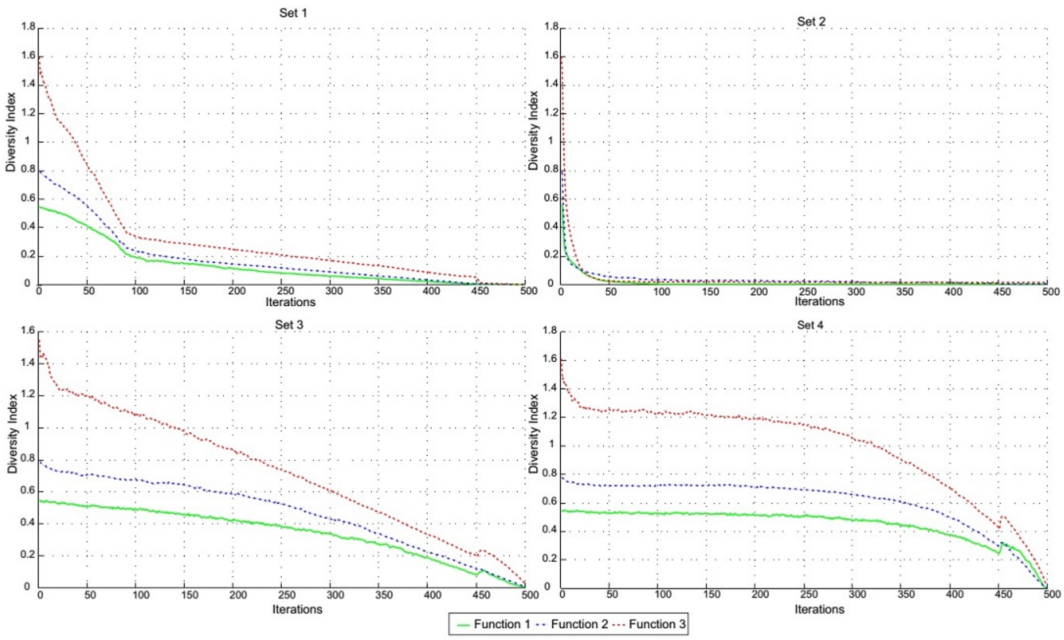

Figure 14. Diversity index employing the SGA on the four sets considering each tested function

there is a correlation between the parameters and the diversity index, it is not possible to predict the variation of the diversity index.

Similar to the FA, the BSA allows a certain control over the diversity index trend, through the parameter settings. However, it is not possible to predict exactly the diversity index behavior.

Additionally, for the FA and BSA, the diversity index variation is different from one problem to the other, demonstrating its problem dependency. On the other hand, for the SGA, the diversity index remained correlated to the $\alpha_{perturbation}$, set *a-piori* and unchanged for every problem. This represents an important advantage, because, even if the algorithm parameters allow the user to force the diversity index trend to be at a desirable region, this process will demand a try and error procedure. Yet, when solving a different problem, this procedure will have to be repeated.

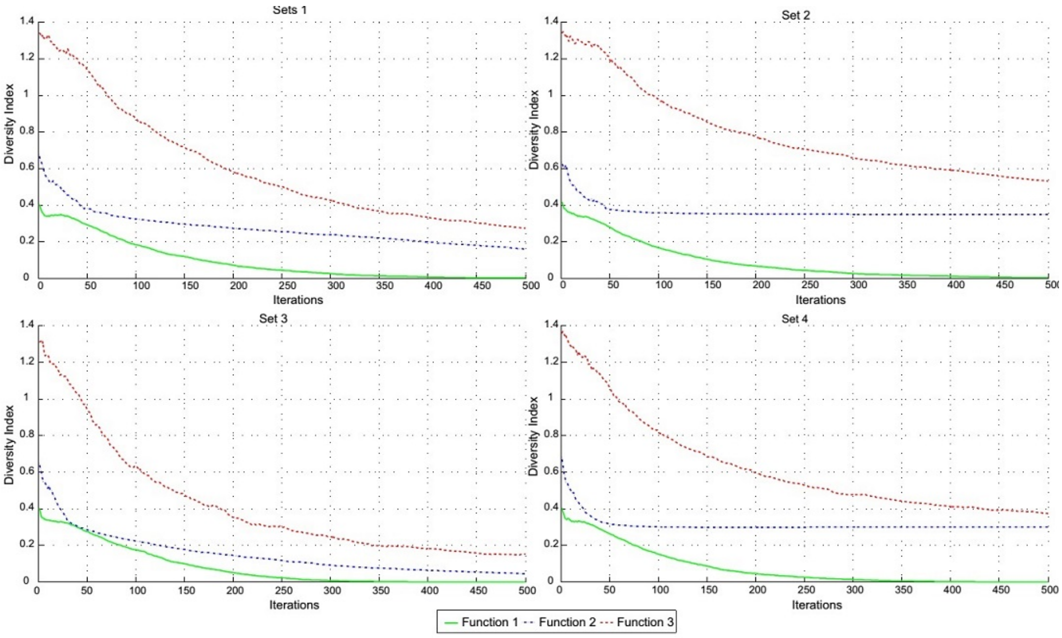

Figure 15. Diversity index employing the BSA on the four sets considering each tested function

REFERENCES

[1] Hasançebi, O. and Erbatur, F. (2002). "Layout optimisation of trusses using simulated annealing." Advances in Engineering Software, 33(7):681–696.

[2] Lamberti, L. (2008). "An efficient simulated annealing algorithm for design optimization of truss structures." Computers & Structures, 86(19):1936–1953.

[3] Kennedy, J. and Eberhart, R. C. (1995). "Particle swarm optimization." In Proceedings of IEEE international conference on neural networks, volume 4, pages 1942–8.

[4] Geem, Z., Kim, J., and Loganathan, G. (2001). "A new heuristic optimization algorithm: Harmony search." Simulation, 76(2):60–68.

[5] Bland, J. (2001). "Optimal structural design by ant colony optimization." Engineering optimization, 33(4):425–443.

[6] Kaveh, A. and Talatahari, S. (2010). "Optimum design of skeletal structures using imperialist competitive algorithm." Computers & Structures, 88(21):1220–1229.

[7] Sadollah, A., Bahreininejad, A., Eskandar, H., and Hamdi, M. (2012). "Mine blast algorithm for optimization of truss structures with discrete variables." Computers & Structures, 102:49–63.

[8] Erol, O. K. and Eksin, I. (2006). "A new optimization method: Big BangâĂŞBig Crunch." Advances in Engineering Software, 37(2):106 – 111.

[9] Hasançebi, O., Teke, T., and Pekcan, O. (2013). "A bat-inspired algorithm for structural optimization." Computers & Structures, 128:77–90.

[10] Kaveh, A. and Farhoudi, N. (2013). "A new optimization method: Dolphin echolocation." Advances in Engineering Software, 59:53–70.

[11] Degertekin, S. and Hayalioglu, M. (2013). "Sizing truss structures using teaching-learning-based optimization." Computers & Structures, 119:177–188.

[12] Kaveh, A., Sheikholeslami, R., Talatahari, S., and Keshvari-Ilkhichi, M. (2014). "Chaotic swarming of particles: A new method for size optimization of truss structures." Advances in Engineering Software, 67:136–147.

[13] Kaveh, A. and Mahdavi, V. (2014). "Colliding bodies optimization method for optimum design of truss structures with continuous variables." Advances in Engineering Software, 70:1–12.

[14] Yang, X.-S. (2009). "Firefly algorithms for multimodal optimization." In Stochastic algorithms: foundations and applications, pages 169–178. Springer.

[15] Kaveh, A. and Talatahari, S. (2010). "A novel heuristic optimization method: charged system search." Acta Mechanica, 213(3-4):267–289.

[16] Yang, X.-S. and Deb, S. (2009). "Cuckoo search via Lévy flights." In Nature Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on, pages 210–214. IEEE.

[17] Yang, X.-S. and Deb, S. (2010). "Engineering optimisation by cuckoo search." International Journal of Mathematical Modelling and Numerical Optimisation, 1(4):330–343.

[18] Kaveh, A. and Khayatazad, M. (2012). "A new meta-heuristic method: Ray Optimization." Structures, 112âĂŞ113:283 – 294.

[19] Kaveh, A. and Zolghadr, A. (2012). "Truss optimization with natural frequency constraints using a hybridized CSS-BBBC algorithm with trap recognition capability." Structures, 102:14–27.

[20] Kaveh, A. and Zolghadr, A. (2014). "Comparison of nine meta-heuristic algorithms for optimal design of truss structures with frequency constraints." Advances in Engineering Software, 76:9–30.

[21] Gonçalves, M. S., Lopez, R. H., and Miguel, L. F. F. (2015). "Search group algorithm: a new metaheuristic method for the optimization of truss structures." Computers & Structures, 153:165–184.

[22] Civicioglu, P. (2013). "Backtracking Search Optimization Algorithm for Numerical Optimization Problems." Appl. Math. Comput., 219(15):8121–8144. doi:10.1016/j.amc.2013.02.017.

[23] Miguel, L. F. F., Lopez, R. H., and Miguel, L. F. F. (2013). "Multimodal size, shape, and topology optimisation of truss structures using the Firefly algorithm." Advances in Engineering Software, 56:23–37.

[24] Gandomi, A. H., Yang, X.-S., and Alavi, A. H. (2011). "Mixed variable structural optimization using firefly algorithm." Structures, 89(23):2325–2336.

RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.