

Original Research Paper

# Natural Reforestation Optimization (NRO): A Novel Optimization Algorithm Inspired by the Reforestation Process

<sup>1</sup>Fernando L. Rodríguez-Gallegos, <sup>2</sup>César A. Rodríguez-Gallegos,  
<sup>3</sup>Andrés A. Rodríguez-Gallegos and <sup>4</sup>Carlos D. Rodríguez-Gallegos\*

<sup>1</sup>Department of Electrical Engineering and Information Technology, Paderborn University, Paderborn 33098, Germany

<sup>2</sup>Department of Mechanical and Industrial Engineering, Concordia University, Montreal H3G 1M8, Canada

<sup>3</sup>Faculty of Systems, Telecommunications and Electronics,

Universidad de Especialidades Espíritu Santo, Samborondón 092301, Ecuador

<sup>4</sup>Department of Electrical and Computer Engineering, National University of Singapore (NUS), Singapore 117583

## Article history

Received: 19-05-2020

Revised: 29-07-2020

Accepted: 01-09-2020

## Corresponding Author:

Carlos D. Rodríguez-Gallegos

Department of Electrical and

Computer Engineering,

National University of

Singapore (NUS), Singapore

117583

Email: carlos.rodriguez@nus.edu.sg

**Abstract:** This paper proposes a new meta-heuristic-based optimization algorithm for single-objective problems. The algorithm is called Natural Reforestation Optimization (NRO) and is inspired by the process in which natural reforestation takes place. The features of this algorithm (such as the distribution of the initial population, the exploration and exploitation mechanisms, the interactions between the particles, the stopping criteria, among others) are discussed and analyzed to show how they are applied to enhance the search of the global solution. The performance of this algorithm is tested with standard single-objective optimization problems (which contain from 2 to 20 optimization variables) and is compared with other optimization algorithms. The results reveal that in general, the NRO algorithm produces solutions close to the global optimal and is able to surpass the other optimization algorithms for many of the benchmark functions. The current study shows the qualities of the NRO algorithm and serves as the starting point for further investigation to take place to keep improving its capabilities.

**Keywords:** Meta-Heuristic Optimization Algorithm, Single-Objective Optimization, Global Optimization

## Introduction

Meta-heuristic optimization algorithms are generally employed for problems which cannot be solved directly. Many of these algorithms are inspired by different natural phenomena such as the social behavior of bird flocking (Particle Swarm Optimization (Clerc, 2010; Parsopoulos and Vrahatis, 2010)), the biological evolution of the organisms (Genetic Algorithm (Davis, 1991; Kantardzic, 2003)), among others (Suman and Kumar, 2006; Rashedi *et al.*, 2009; Dasgupta *et al.*, 2011; Krishnanand *et al.*, 2009; Yang, 2011; Beheshti and Shamsuddin, 2013; Dorigo and Blum, 2005; Alvarez-Alvarado and Jayaweera, 2018a; Kohli and Arora, 2018; Alvarez-Alvarado and Jayaweera, 2018b; Cui *et al.*, 2016). In this study we propose a new optimization approach based on the natural reforestation of trees named Natural Reforestation Optimization (NRO).

Similar to other meta-heuristic algorithms, NRO algorithm optimizes a defined problem by iteratively searching for the best solution. For this, it employs a population of particles (called trees) from which new

particles are produced (called seeds) which are distributed within the search space (called forest). The amount of seeds each tree produces is based on the fertility of the place it was planted i.e., how good is its current solution. To distribute the seeds, they can, on the one hand, fall from the tree without the influence of the wind (these will then land close to their parent tree) or, on the other hand, leave their parent tree due to the wind influence in which the wind direction (that is influenced by the location of the current best solution) and speed are taken into account. The reduction of the wind speed due to other trees that block its path is also considered. Once the seeds have landed, they will grow and become trees and their associated solution is evaluated. The reforestation process is then repeated.

Based on the reforestation principle, our algorithm has the following characteristics:

- It prioritizes the exploration of the search space in the beginning i.e., the number of seeds influenced by the wind (which can travel a long distance) is the

highest at the first iteration and reduces as the number of iteration increases. These seeds are also useful to escape from local optimal regions

- It prioritizes the exploitation of the search space towards the end i.e., the number of seeds not influenced by the wind (which travel only a short distance) increases as the number of iteration increases. The area range within these seeds can land is also reduced with increasing number of iterations
- For the case of the seeds influenced by the wind, the search of the new solutions is affected by the location of the current best solution (which influences the wind direction) as well as the location of other trees (if close to the trajectory of the seeds) since they can reduce the wind speed based on how good is their associated solution. Therefore, the new population of particles (seeds) have the opportunity to interact with the current population (trees) in order to search for the best solution
- The seeds that are not influenced by the wind are especially useful when the solutions are close to the global solution as they can perform a more intensive search within this area (exploitation). Particles located close to the global solution are expected to be obtained as the number of iterations increases and hence the number of these seeds increases (while the ones influenced by the wind decreases) as the number of iterations rises
- The better the solution obtained by a tree, the more seeds it will be assigned so that more exploration and exploitation can take place in this area

The rest of the paper is arranged as follows. In Section II, the principles of the natural reforestation process, from which our algorithm is inspired, are explained. Subsequently, the algorithm formulation is detailed in section III while the method to set the values of the variables are given in section IV. In section V, the performance of the NRO algorithm is compared with other optimization algorithms to solve different optimization problems. Section VI then concludes the paper. An Appendix section is presented at the end of this document describing the benchmark functions and providing the nomenclature table (Table 3).

## Natural Reforestation Process

The NRO algorithm takes into account the following elements involved in the natural reforestation process: The forest, the trees, the seeds and the wind. Here, the forest represents the range of values in which the optimization variables are bounded. The trees represent the population of solutions whose location within the forest defines the value of their optimization variables. Their height is calculated from a fitness function based on the optimization problem. In case of a minimization

(maximization) problem, the lower (higher) the objective function value for a particular tree, the greater its height. The seeds represent the new solutions that are utilized to search for better solutions within the forest. They are generated from the trees (the tree that generates a seed is referred as its parent tree). The place where they are planted can depend on several factors such as the location and height of the trees as well as the wind direction and speed.

The natural reforestation process is as follows:

- 1) The population of trees is planted among the forest
- 2) The taller the tree, the more seeds it can generate
- 3) The seeds produced by the trees can be dispersed with two options:
  - a) No wind influence: Some seeds can fall without the wind intervention. These will land close to their parent tree
  - b) Wind influence: The wind can separate the seeds from their parent tree and the place where they are planted depends on their parent tree's location, the wind direction and speed. The wind speed may be reduced due to other trees blocking its path
- 4) Based on the soil quality of the place where the seed is planted, it will grow into a tree with a certain height (the more fertile the land, the greater its height)
- 5) As the forest has a limited space, the trees will compete with each other for nutrients and sunlight and the ones with the greatest heights will remain
- 6) The process is then repeated

## Algorithm Formulation

Based on the process described in the previous section, the steps of the NRO algorithm for an optimization problem are formulated below:

- 1) Initialization. The following parameters are defined:
  - a) Objective function  $f^{obj}$  which is in terms of the  $n^{var}$  optimization variables  $x_{(k)}$ ,  $k = 1, 2, \dots, n^{var}$
  - b) Range of values the optimization variables can have  $x_{\min,(k)} < x_{(k)} < x_{\max,(k)}$ ,  $k = 1, 2, \dots, n^{var}$ . This will constitute the size of the forest
  - c) Population size  $n^{pop}$
  - d) Total number of seeds that will be generated at each iteration  $N^{seed}$
  - e) Percentage of seeds that are planted without the wind influence, called internal seeds, at the initial iteration  $per^{seed,ini}$  and at the final iteration  $per^{seed,fin}$
  - f) Multidimensional sphere size within which the internal seeds are planted (the parent tree is the center of the sphere) at the initial iteration  $Sph^{seed,ini}$  and at the final iteration  $Sph^{seed,fin}$ . The

sphere size is measured as a percentage of the maximum distance within the forest

- g) Minimum distance between the traveling external seeds and the trees in order for the latter (called blocking trees) to reduce the speed of the wind that is transporting the external seeds  $dist_{min}^{wind,tree}$ . This distance is measured as a percentage of the maximum distance within the forest
- h) Maximum distance that an external seed will reach if it is generated from the highest tree of the current iteration and with the highest wind speed (without the influence of blocking trees)  $dist_{h_{max},v_{max}}^{seed}$ . This distance is measured as a percentage of the maximum distance within the forest
- i) Percentage of the wind speed reduction when it is blocked by the highest tree of the current iteration  $speed_{h_{max}}^{wind,red}$
- j) Height of the tree with the best (closest to the optimal solution) and worst (furthest from the optimal solution) objective value at the current iteration:  $h_{max}$  and  $h_{min}$ , respectively
- k) Neighborhood parameter to generate the initial population  $\rho^{neigh}$
- l) Radius of the individual neighborhood of the initial population  $r^{neigh} = \sqrt{n^{var}} / \rho^{neigh}$
- m) Maximum accepted distance between the trees to stop the algorithm  $dist_{max}^{stop}$
- n) Maximum number of consecutive iterations to stop the algorithm in which no improved solution has been obtained  $n1_{max}^{it}$
- o) Maximum number of iterations to stop the algorithm  $n2_{max}^{it}$
- p) The number of the consecutive iterations without an improvement of the solution  $n3^{it}$  is set to zero
- q) The number of the current iteration  $n^{it}$  is set to one

## 2) Generation of the initial population

- a) Each of the  $n^{pop}$  initial trees are randomly distributed in the search space and are assigned with optimization variables values  $x_{(k,p)}^{tree}$ ,  $k = 1, 2, \dots, n^{var}$ ,  $p = 1, 2, \dots, n^{pop}$  with the condition that no particle should be within the neighborhood of another particle (to assure diversity, similar to (Chelouah and Siarry, 2000)). Therefore, for any two particles with values of the optimization variables of  $x_{(k,a)}^{tree}$  and  $x_{(k,b)}^{tree}$ ,  $a \neq b$ , respectively, the following condition must hold (this is represented in Fig. 1a for a problem with two optimization variables)<sup>1</sup>:

$$\sqrt{\sum_{k=1}^{n^{var}} \left( \frac{x_{(k,a)}^{tree} - x_{(k,b)}^{tree}}{x_{max(k)} - x_{min(k)}} \right)^2} < r^{neigh} \quad (1)$$

- b) Calculation of the objective value for each of the initial trees  $Obj_{(p)}$ ,  $p = 1, 2, \dots, n^{pop}$
- c) Calculation of the height for each of the initial trees. The tree height  $h_{(p)}^{tree}$ ,  $p = 1, 2, \dots, n^{pop}$  is calculated based on a linear relation with its objective value. Here, the trees with the best and worst  $Obj_{(p)}$  are assigned a height of  $h_{max}^{tree}$  and  $h_{min}^{tree}$ , respectively, as shown<sup>2</sup>:

$$h_{(p)}^{tree} = \begin{cases} \frac{h_{max}^{tree} - h_{min}^{tree}}{\min(Obj) - \max(Obj)} \cdot (Obj_{(p)} - \min(Obj)) \\ + h_{max}^{tree}, \text{ if minimization problem} \\ \frac{h_{max}^{tree} - h_{min}^{tree}}{\max(Obj) - \min(Obj)} \cdot (Obj_{(p)} - \min(Obj)) \\ + h_{min}^{tree}, \text{ if maximization problem} \end{cases} \quad (2)$$

- d) Among the current population, the tree with the best solution is selected (the tree closest to the optimal result). The value of the optimization variables and the objective value of this tree are stored in  $B_{x(k)}$  and  $B_{Obj}$ , respectively

## 3) Seeds production

- a) The number of seeds that each tree will produce  $n_{(p)}^{seed}$ ,  $p = 1, 2, \dots, n^{pop}$  is based on their height (the greater their height, the more seeds they will produce)<sup>3</sup>:

$$n_{(p)}^{seed} = \text{round} \left( \frac{N^{seed} \cdot h_{(p)}^{tree}}{\sum_{k=1}^{n^{pop}} h_{(k)}^{tree}} \right) \quad (3)$$

- b) When the algorithm starts, the percentage of internal seeds is low with a value of  $per^{seed,ini}$  while the one of external seeds is high. As the iterations go on, the percentage of internal seeds increases reaching a maximum of  $per^{seed,fin}$  when  $n^{it} = n2_{max}^{it}$ , while the one of external seeds decreases. This takes place to give emphasis to the exploration at the beginning and then to the exploitation towards

<sup>1</sup> Because the optimization variables can have different searching ranges, they should be normalized. This takes place in Equation 1, 7, 9 and 11.

<sup>2</sup> The functions  $\min(z)$  and  $\max(z)$  represent the minimum and maximum values within the  $z$  values, respectively.

<sup>3</sup> The function  $\text{round}(z)$  rounds the value of  $z$  to its closest integer.

the end of the algorithm. The number of internal seeds  $n_{(p)}^{seed,int}$ ,  $p = 1, 2, \dots, n^{pop}$  from each tree is then calculated as:

$$n_{(p)}^{seed,int} = \text{round} \left( n_{(p)}^{tree} \cdot \left( \left( \frac{per^{seed,fin} - per^{seed,ini}}{n2_{max}^{it} - 1} \right) \cdot \frac{(n^{it} - 1)}{100\%} + \frac{per^{seed,ini}}{100\%} \right) \right) \quad (4)$$

c) The number of external seeds for each tree  $n_{(p)}^{seed,ext}$ ,  $p = 1, 2, \dots, n^{pop}$  is:

$$n_{(p)}^{seed,ext} = n_{(p)}^{seed} - n_{(p)}^{seed,int} \quad (5)$$

4) Estimation of the place where the internal seeds are planted

a) The internal seeds are the ones that fall from the trees without the wind influence. Consequently, they are expected to land close to their parent tree ( $x_{(k,p)}^{tree}$ ) who acts as the sphere center where these seeds can be placed. At the initial iteration, the radius of this sphere is  $Sph^{seed,ini}$ , this is then linearly reduced after each iteration reaching a value of  $Sph^{seed,fin}$  when  $n^{it} = n2_{max}^{it}$ . The reduction of the sphere size takes place to intensify the search of the current area. The location (values of the optimization variables) for each of the  $n_{(p)}^{seed,int}$  internal seeds associated to each of the  $n^{pop}$  trees  $x_{(k,p,m)}^{seed,int}$ ,  $k = 1, 2, \dots, n^{var}$ ,  $p = 1, 2, \dots, n^{pop}$ ,  $m = 1, 2, \dots, n_{(p)}^{seed,int}$  is estimated as<sup>4</sup>:

$$x_{(k,p,m)}^{seed,int} = x_{(k,p)}^{tree} \pm \left( x_{max(k)} - x_{min(k)} \right) \cdot \text{rand} \cdot \left( \left( \frac{Sph^{seed,ini} - Sph^{seed,fin}}{1 - n2_{max}^{it}} \right) \cdot \frac{(n^{it} - 1)}{100\%} + \frac{Sph^{seed,ini}}{100\%} \right) \quad (6)$$

b) In case any of the particles goes beyond the search space, they will be brought back to the searching boundary:

if  $x_{(k,p,m)}^{seed,int} > x_{max(k)}$ , then

$$x_{(k,p,m)}^{seed,int} = x_{max(k)}$$

if  $x_{(k,p,m)}^{seed,int} < x_{min(k)}$ , then

$$x_{(k,p,m)}^{seed,int} = x_{min(k)}$$

5) Estimation of the place where the external seeds are planted

a) The external seeds are the ones dispersed from their parent tree with the wind influence. The location where they are planted depends on the wind direction and speed, the height of their parent tree and the blocking trees. The falling seeds are assumed to experience a two dimensional - consisting of  $x$  and  $y$  axis - parabolic movement which follows the kinematic equations. Here, the gravity and height of the parent tree control the movement of the seeds in the  $y$  axis (height of the seed) while their movement in the  $x$  axis (travelled distance of the seed) is affected by the wind as well as the height and location of the parent and blocking trees (the direction of the  $x$  axis is the same as the wind direction). For simplicity, the rest of this section deals with the travelled trajectory of an individual seed. This procedure can then be repeated for the other external seeds

b) Once the external seed is traveling, its travelled distance  $dist^{seed}$ , when reaching a final position of  $x^{seed,fin}$ ,  $k = 1, 2, \dots, n^{var}$  within the forest, is calculated based on its initial location  $x^{seed,ini}$ ,  $k = 1, 2, \dots, n^{var}$  (location of its parent tree), as:

$$dist^{seed} = \sqrt{\sum_{k=1}^{n^{var}} \left( \frac{x_{(k)}^{seed,fin} - x_{(k)}^{seed,ini}}{x_{max(k)} - x_{min(k)}} \right)^2} \quad (7)$$

c) To estimate the value of the gravity  $g$ , it is assumed that, when the external seed is generated from the tree with the highest height in the current iteration and the wind is at maximum speed, it will be able to travel a total distance of  $dist_{h_{max},v_{max}}^{seed}$  within the forest (assuming no blocking tree is present to reduce the wind speed). The gravity is then defined as<sup>5</sup>:

$$g = \frac{2 \cdot \max \left( h_{(p)}^{tree} \right)}{\left( \sqrt{n^{var}} \cdot dist_{h_{max},v_{max}}^{seed} / 100\% \right)^2} \quad (8)$$

<sup>4</sup> The function rand generates a random number between zero and one drawn from the standard uniform distribution. The  $\pm$  symbol indicates that it can be an addition or a subtraction (the operation is randomly selected).

<sup>5</sup> Equation 8 and 13 are obtained based on the kinematic equations.

- d) The wind direction  $dir_{(k)}^{wind}$ ,  $k = 1, 2, \dots, n^{var}$  is a  $n^{var}$ -dimensional vector due to the  $n^{var}$  optimization variables. It is calculated taking into account the position where the tree with the best result is located, as follows:

$$dir_{(k)}^{wind} = \text{rand} \cdot \frac{(B_{x(i)} - x_{(k,p)}^{tree})}{x_{\max(k)} - x_{\min(k)}} \quad (9)$$

- e) The initial wind speed is randomly selected between zero (no wind) and one (maximum wind speed):

$$speed_{(k)}^{wind,ini} = \text{rand} \quad (10)$$

- f) Based on the wind direction, the position of the trees that are close to its traveling path (the blocking trees) are stored. Among the total population, the blocking trees are the ones whose minimum distance between them and the traveling seed is lower than  $dist_{\min}^{wind,tree}$ . This is represented in Fig. 1b for the case of a problem with two optimization variables
- g) Once all the blocking trees are found, the traveling seeds will reach a particular blocking tree (which is located at a position of  $x_{(k)}^{tree,block}$   $k = 1, 2, \dots, n^{var}$ ) when it has travelled a distance of:

$$dist^{seed,tree} = \sqrt{\sum_{k=1}^{n^{var}} \left( \frac{x_{(k)}^{tree,block} - x_{(k)}^{seed,ini}}{x_{\max(k)} - x_{\min(k)}} \right)^2} \quad (11)$$

- h) To estimate the location at which the external seed is planted, it is considered that it starts at an initial height equal to the height of its parent tree  $h_{(p)}^{tree,parent}$  and moves with an initial speed and direction equal to the one of the wind ( $speed_{(k)}^{wind,ini}$  and  $dir_{(k)}^{wind}$ , respectively). Once the seed reaches a blocking tree, its speed  $speed_{(k)}^{wind}$ ,  $k = 1, 2, \dots, n^{var}$  is linearly reduced based on the height of this tree  $h^{tree,block}$ . The greater the height of the blocking tree, the higher will be the speed reduction. If the blocking tree is the one with the maximum height at the current iteration, it will reduce the wind speed by a factor of  $100\% - speed_{h_{\max}}^{wind,red}$ . Based on this, when a blocking tree is reached, the new value of  $speed_{(k)}^{wind}$  is calculated as:

$$speed_{(k)}^{wind} = speed_{(k)}^{wind} \cdot \left( 1 - \frac{h^{tree,block}}{\max(h_{(p)})} \cdot \frac{speed_{h_{\max}}^{wind,red}}{100\%} \right) \quad (12)$$

- i) Because the seed is in a state of free fall in the y axis, its height  $h_{(k,p,n)}^{seed}$  is first equal to its tree parent height ( $h_{(p)}^{tree}$ ) but it is then reduced in time  $t$  in the following way:

$$h_{(k,p,n)}^{seed} = h_{(p)}^{tree} - \frac{g \cdot t^2}{2} \quad (13)$$

- j) When  $h_{(k,p,n)}^{seed}$  reaches zero, the seed has landed. Figure 1c illustrates the parabolic trajectory of an external seed
- k) The landing position of all the  $n_{(p)}^{seed,ext}$  external seeds from the  $n^{pop}$  trees are stored in  $x_{(k,p,m)}^{seed,ext}$ ,  $k = 1, 2, \dots, n^{var}$ ,  $p = 1, 2, \dots, n^{pop}$ ,  $n = 1, 2, \dots, n_{(p)}^{seed,ext}$
- l) The equations defined in this section are employed to calculate the landing positions of the external seeds by assuming that the units for the distance and height ( $dist^{seed}$ ,  $dist^{seed,tree}$ ,  $h_{(p)}^{tree}$ ,  $h_{(p)}^{tree,block}$  and  $h_{(k,p,n)}^{seed}$ ) are in  $m$ , for the gravity ( $g$ ) is in  $m/s^2$ , for the ( $speed_{(k)}^{wind,ini}$ ) is in  $m/s$  and for the time ( $t$ ) is in  $s$
- m) In case any of the particles goes beyond the search space, they will be brought back to the searching boundary:

$$\begin{aligned} &\text{if } x_{(k,p,m)}^{seed,ext} > x_{\max(k)}, \text{ then} \\ &x_{(k,p,m)}^{seed,ext} = x_{\max(k)} \\ &\text{if } x_{(k,p,m)}^{seed,ext} < x_{\min(k)}, \text{ then} \\ &x_{(k,p,m)}^{seed,ext} = x_{\min(k)} \end{aligned}$$

- 6) Estimation of the new population of trees
- a) At this step it is assumed that the seeds have grown into trees. So it is necessary to select the  $n^{pop}$  trees that will stay for the next iteration. For this, the objective values from the particles stored in  $x_{(k,p,m)}^{int}$  and  $x_{(k,p,m)}^{seed,ext}$  are obtained and are stored together with the objective values from parent trees in the  $Obj^{seed,tree}$  list. From this list, the  $n^{pop}$  particles

- with the best objective values (the ones closest to the optimal solution) are selected to become the remaining trees within the forest
- b) The objective value of the remaining trees are stored in  $Obj_{(p)}$
- c) The height of the remaining trees are calculated and stored in  $h_{(p)}^{tree}$
- d) Among the current population, the tree with the best solution is selected (the tree associated with the best objective value). The value of the optimization variables and the objective value of this tree are stored in  $B_{x_{(k)}}$  and  $B_{Obj}$ , respectively
- e) If the  $B_{Obj}$  value from the current iteration is lower than the one from the previous iteration, then  $n3^{it} = 0$ . Otherwise  $n3^{it} = n3^{it} + 1$
- f) The iteration number is increased by one:

$$n^{it} = n^{it} + 1 \quad (14)$$

7) Analysis of the stopping criteria

- a) If any of the following conditions are fulfilled, then the algorithm is stopped and the final results ( $B_{x_{(k)}}$  and  $B_{Obj}$ ) are provided:
  - i) if  $\max(Obj_{(p)}) - B_{Obj} < dist_{max}^{tree,stop}$
  - ii) if  $n3^{it} = n1_{max}^{it}$

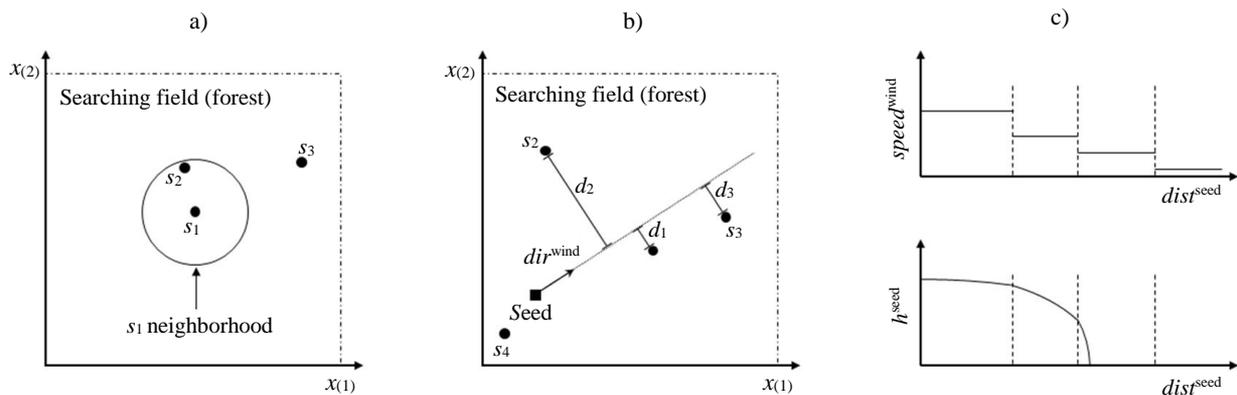
iii) if  $n^{it} = n2_{max}^{it}$

- b) If neither of the conditions from the previous step were fulfilled, the algorithm goes back to the numeral 3. Seeds production

The algorithm structure previously described is summarized in Fig. 2.

**Table 1:** Value assignment of the constants required for the NRO algorithm

Parameters	Values
$n^{pop}$	max (30, $4 \cdot n^{var}$ )
$N^{seed}$	$n^{pop}$
$per^{seed,ini}$	5%
$per^{seed,fin}$	95%
$Sph^{seed,ini}$	5%
$Sph^{seed,fin}$	0.001%
$dist_{min}^{wind,tree}$	5%
$dist_{h_{max},v_{max}}^{seed}$	50%
$speed_{h_{max}}^{wind,red}$	50%
$h_{max}$	100
$h_{min}$	10
$p^{neigh}$	$n^{pop} \cdot n^{var}$
$dist_{max}^{tree,stop}$	0.01
$n1_{max}^{it}$	max (10, $n^{pop} \cdot n^{var} / 10$ )
$n2_{max}^{it}$	$10 \cdot n^{pop} \cdot n^{var}$



**Fig. 1:** (a) Neighborhood constraint for the definition of the initial population for a problem with two optimization variables  $x_{(1)}$  and  $x_{(2)}$ . The solution  $s_2$  cannot be taken into account as it is within the neighborhood of  $s_1$  while  $s_3$  can be considered as it is not within this neighborhood. (b) Selection of the blocking trees based on their shortest distance to the seed traveling path. In this example, the minimum distance from the solutions  $s_1$  and  $s_3$  ( $d_1$  and  $d_3$ , respectively) are assumed to be lower than  $dist_{min}^{wind,tree}$  so they are considered as blocking trees while the distance  $d_2$  from  $s_2$  is assumed to be greater and therefore it is not considered as a blocking tree. Because the analyzed seed will not travel towards  $s_4$  (due to the wind direction),  $s_4$  is not considered as a blocking tree. (c) Representation of the trajectory of the external seeds. The dashed lines show the locations of the blocking trees. At the beginning the seed has a height equal to the one of its parent tree and moves with a speed equal to the one of the wind speed. As it travels, its height decreases following a parabolic trajectory. Once it reaches the location of a blocking tree, the wind speed is reduced based on its height, changing the parabolic trajectory of the seed. The seed lands in the location where it reaches a height equal to zero

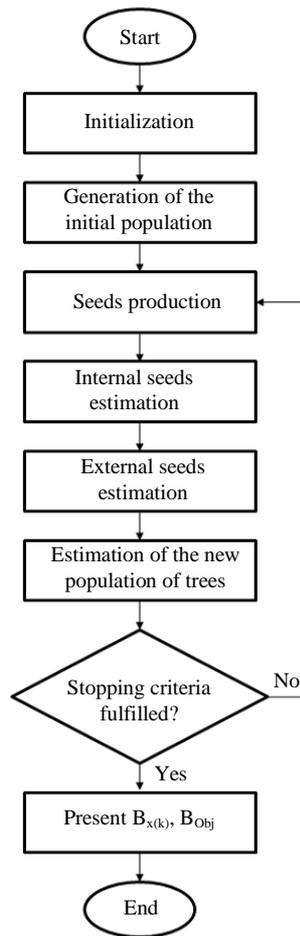


Fig. 2: Flow chart of the NRO algorithm

### Parameter Specifications

Based on an empirical analysis in which the algorithm was tested under several objective functions, the assigned values to the constants defined in section III Algorithm Formulation are presented in Table 1. Despite the proposed values provided in this table, they can be further improved when dealing with a specific optimization problem.

### Results and Discussion

The performance of the NRO algorithm is evaluated by employing a series of benchmark functions typically employed to test optimization algorithms. These are described at the Appendix section. Furthermore, four optimization algorithms are also tested to compare their results to the ones from NRO: Particle Swarm Optimization (PSO) (Eberhart and Shi, 2000)<sup>6</sup>, Continuous Genetic Algorithm (CGA) (Chelouah and

Siarry, 2000), Enhanced Simulated Annealing (ESA) (Siarry *et al.*, 1997) and Enhanced Continuous Tabu Search (ECTS) (Chelouah and Siarry, 1999). These are selected due to the high number of optimization problems found in the literature which have been solved with them. The authors programmed the previous algorithms by following the descriptions of the cited papers so that a deeper analysis can be made. All the simulations were run on the super computer from the National University of Singapore which is composed by clusters with RAM of 48 GB and Intel Xeon X5650 of 2.67 GHz. Their results are then summarized in Table 2 and Fig. 3.

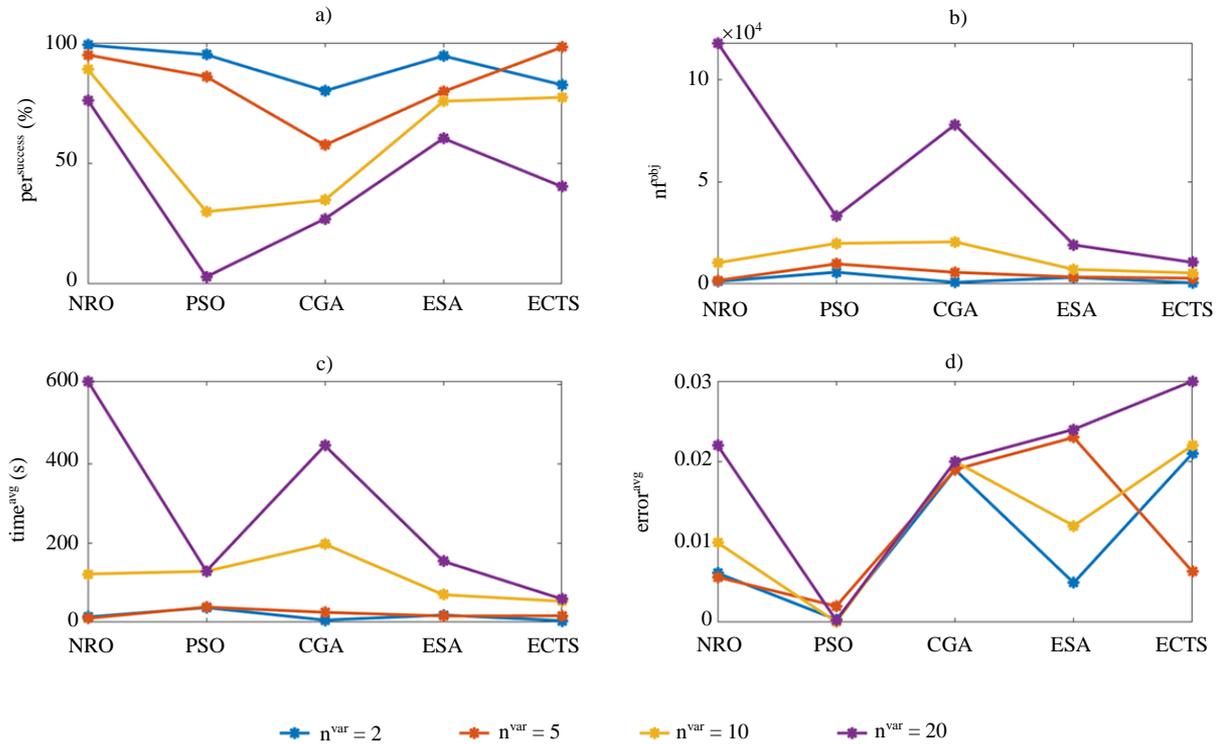
Because of the semi-randomness presented in the optimization algorithms (e.g., the location of the initial population is variable), each of them were evaluated 100 times so that a fair analysis could be performed. Table 2 presents the obtained outcomes of these algorithms with respect to the benchmark functions, each composed by a number of  $n^{var}$  optimization variables. Only the successful results were taken into account and their averaged values are provided. A simulated result  $r^{sim}$  is considered successful if it is close to the known/real analytical solution of the optimization problem  $r^{real}$ , by fulfilling the following condition:

$$|r^{sim} - r^{real}| \leq \alpha_1 \cdot |r^{real}| + \alpha_2 \tag{15}$$

where  $|r^{sim} - r^{real}|$  is the error of the simulation while  $\alpha_1$  and  $\alpha_2$  are the coefficients to determine the success condition, both were set to 0.1.

Table 2 presents the percentage of successful results ( $per^{success}$ ), the average number of times the objective function was evaluated ( $n_f^{obj}$ ), the average simulation time ( $time^{avg}$ ) and the average absolute error ( $error^{avg}$ ). As previously indicated,  $n_f^{obj}$  and  $error^{avg}$  were calculated by considering only the successful results. Here, it can be appreciated that the NRO algorithm achieves the highest percentage of success for most of the benchmark functions. In comparison to the other optimization algorithms, the NRO shows a noticeable advantage when dealing with the Bea, Pow ( $n^{var} = 10, 20$ ) and Per ( $n^{var} = 10, 20$ ) functions. In general, it can be seen that, as the number of optimization variables increases, the performance of the algorithms tend to decrease (in particular for the PSO). Nevertheless, among them, the ESA is the one who achieves 100% of success rate for the SS, SDP and Sph regardless of the employed optimization variables. For a few functions (Pow with  $n^{var} = 20$  and Per with  $n^{var} = 5, 10, 20$ ) the NRO algorithm achieves values of less than 80% success rate. However, with respect to the Pow ( $n^{var} = 20$ ) and Per ( $n^{var} = 10, 20$ ) functions, it still reaches a better result than the ones obtained from the benchmark optimization algorithms being overcome only by the PSO and ECTS algorithms when evaluating the Per ( $n^{var} = 5$ ) function.

<sup>6</sup> The value of the PSO parameters are the employed in this study when the Clerc's constriction method was applied. As no stopping criteria was formulated, the ones from [16] were considered.



**Fig. 3:** Results based on the number of optimization variables the benchmark functions contain

**Table 2:** Successful results from the optimization algorithms. The numbers in violet and blue are the best and worst result from each category respectively

Functions	$n^{var}$	$per^{success}(\%)$					$n^{obj}$					$time^{avg}(s)$					$error^{avg}$				
		NRO	PSO	CGA	ESA	ECTS	NRO	PSO	CGA	ESA	ECTS	NRO	PSO	CGA	ESA	ECTS	NRO	PSO	CGA	ESA	ECTS
Bea	2	90	78	93	25	65	322	4588	784	1170	507	5	47	8	13	2	2E-4	9E-12	2E-2	4E-2	3E-3
Boo	2	100	99	92	99	100	554	4207	857	1401	519	5	29	6	10	3	6E-4	6E-11	2E-2	5E-3	5E-4
Mat	2	100	94	100	100	100	251	3725	749	1468	518	2	23	5	10	2	2E-4	7E-12	3E-3	0	2E-5
CT	2	100	96	100	100	100	239	7518	685	10100	451	2	43	4	64	8	4E-04	2E-6	1E-3	3E-6	2E-2
Sch_N2	2	100	90	99	100	65	782	4821	677	1686	409	5	21	3	8	2	2E-4	2E-14	2E-2	0	4E-2
Sch_N4	2	100	76	100	100	80	842	9000	693	10025	412	4	29	2	32	3	7E-4	4E-7	1E-2	4E-4	4E-2
DW	2	100	94	98	100	87	599	8611	713	10100	382	4	40	3	51	3	5E-2	1E-3	6E-2	0	7E-2
Gri	2	100	99	35	100	68	3966	8991	942	1399	477	102	182	21	30	13	1E-2	4E-3	6E-2	0	6E-2
Boh_f1	2	97	100	12	100	38	3907	5372	1109	1395	493	23	25	5	7	3	1E-2	4E-12	2E-2	0	5E-2
Boh_f2	2	97	100	23	100	51	3639	5326	1014	1398	495	28	31	6	9	4	2E-2	5E-13	3E-2	0	4E-2
Boh_f3	2	100	99	26	100	64	4116	5455	984	1399	493	33	34	7	9	4	5E-3	2E-12	3E-2	0	3E-2
SHC	2	100	100	100	100	88	333	5783	695	6508	524	2	34	4	40	4	7E-4	3E-5	6E-3	3E-5	3E-5
DP	2	100	94	90	85	99	450	7735	793	1247	517	2	26	3	4	3	5E-4	4E-14	2E-2	3E-2	3E-3
Pow	5	100	68	49	98	100	1173	20153	4593	3265	2863	4	54	13	9	10	3E-3	1E-7	2E-2	2E-2	7E-3
Pow	10	98	13	23	79	85	4569	45000	22922	6459	5419	43	377	189	60	54	2E-2	6E-7	4E-2	5E-2	4E-2
Pow	20	45	0	9	2	0	69708	-	81880	16969	-	618	-	655	218	-	5E-2	-	4E-2	10E-2	-
SS	2	100	100	94	100	100	480	3855	836	1393	489	3	19	4	7	3	5E-4	2E-11	2E-2	0	2E-4
SS	5	100	89	48	100	100	3345	4632	8463	3586	2768	25	27	50	22	21	4E-3	2E-9	1E-2	0	4E-3
SS	10	95	19	26	100	72	19122	6159	29412	7383	5358	457	124	607	153	126	1E-2	3E-7	2E-2	0	4E-2
SS	20	85	0	16	100	2	121726	-	80945	22003	10581	449	-	227	85	35	2E-2	-	3E-2	0	8E-2
SDP	2	100	100	100	100	100	124	2538	660	1387	492	1	10	3	6	3	6E-5	2E-11	10E-5	0	7E-7
SDP	5	100	92	100	100	100	201	2591	1714	3464	2768	2	15	10	21	20	5E-4	2E-10	4E-4	0	4E-6
SDP	10	100	44	100	100	100	333	2960	8109	6982	5375	2	11	31	29	24	1E-3	3E-9	4E-4	0	10E-6
SDP	20	100	6	100	100	100	663	4325	77525	14405	10569	7	33	653	188	108	6E-4	2E-8	6E-5	0	1E-5
Sph	2	100	99	100	100	100	271	3489	795	1400	487	2	15	4	6	2	4E-4	8E-12	4E-3	0	4E-5
Sph	5	100	94	63	100	100	1447	4011	5871	3645	2771	4	9	14	9	15	3E-3	4E-9	3E-2	0	2E-4
Sph	10	100	38	23	100	100	5680	5004	24624	7789	5359	31	21	107	36	43	1E-2	2E-7	3E-2	0	2E-3
Sph	20	94	1	10	100	100	27114	9210	71699	23103	10570	128	34	246	124	33	2E-2	3E-5	1E-2	0	2E-2
Per	2	100	99	100	100	100	221	7073	660	1413	481	1	22	2	5	2	5E-4	2E-12	2E-3	2E-3	3E-4
Per	5	75	87	28	1	91	2447	17732	7880	3078	2873	15	85	39	17	17	2E-2	1E-2	4E-2	10E-2	2E-2
Per	10	52	36	2	0	30	22110	40238	18085	-5431	75	109	50	-	18	6E-3	1E-4	1E-2	-	4E-2	
Per	20	57	8	0	0	0	370997	85871	-	-	-	1835	320	-	-	-	1E-2	7E-4	-	-	-

**Table 3:** Nomenclature table

$\rho^{neigh}$	Neighborhood parameter to generate the initial population
$dir^{wind}$	Wind direction
$dist^{seed}$	Travelled distance of a seed
$dist_{h_{max},v_{max}}^{seed}$	Maximum distance that a seed can reach if it is generated from the highest tree with the highest wind speed
$dist^{seed,tree}$	Travelled distance by a seed at which will encounter a blocking tree
$dist_{max}^{tree,stop}$	Maximum accepted distance between the trees to stop the algorithm
$dist_{min}^{wind,tree}$	Minimum distance between the traveling external seeds and the trees in order for the latter to reduce the speed of the wind that is transporting the external seeds
$error^{avg}$	Average simulation error
$f^{obj}$	Objective function
$h^{seed}$	Height of the seed
$h^{tree}$	Height of the tree
$h_{max}^{tree} (h_{min}^{tree})$	Height of the tree with the best (and worst) objective value
$h^{tree,block}$	Height of the blocking tree
$n^{it}$	Number of the current iteration
$n^{pop}$	Population size
$n^{seed}$	Number of produced seeds from a tree
$n^{seed,ext}$	Number of external seeds
$n^{seed,int}$	Number of internal seeds
$n^{var}$	Number of optimization variables
$n1_{max}^{it}$	Maximum number of consecutive iterations to stop the algorithm in which no improved solution has been obtained
$n2_{max}^{it}$	Maximum number of iterations to stop the algorithm
$n3^{it}$	Number of the consecutive iterations without an improvement of the solution
$nf^{obj}$	Average number of times the objective function was evaluated
$N^{seed}$	Total number of seeds generated per iteration
$Obj$	Objective value
$Obj^{seed,tree}$	Objective values from all the seeds and trees
$per^{seed,fin}$	Percentage of seeds at the final iteration
$per^{seed,ini}$	Percentage of seeds at the initial iteration
$per^{success}$	Percentage of simulation success
$r^{neigh}$	Radius of the individual neighborhood of the initial population
$r^{real}$	Real analytical result
$r^{sim}$	Simulated result
$speed^{wind}$	Wind speed
$speed^{wind,ini}$	Initial wind speed
$speed_{h_{max}}^{wind,red}$	Percentage of the wind speed reduction when it is blocked by the highest tree
$Sph^{seed,fin}$	Multidimensional sphere size at the final iteration
$Sph^{seed,ini}$	Multidimensional sphere size at the initial iteration
$time^{avg}$	Average simulation time
$x^{(k)}$	$k^{th}$ optimization variable
$x_{max}(k) (x_{min}(k))$	Maximum (and minimum) allowed value of the $k^{th}$ optimization variable
$x^{seed,ext}$	Landing position from the external seeds
$x^{seed,fin}$	Location of seed after travelling
$x^{seed,ini}$	Original location of travelling seed
$x^{seed,int}$	Landing position from the internal seeds
$x^{tree}$	Tree location
$x^{tree,block}$	Blocking tree location

The analysis of the algorithms based on their value of  $nf^{obj}$  is required as, for complex problems, the evaluation of the objective function tends to represent a high (if not the highest) percentage of the simulation time. So, the lower this value is, the less computational time is required. All of the evaluated optimization algorithms reveal that as the number of its optimization

variable increases, so does the  $nf^{obj}$  value. This is an expected result because by the increase of  $n^{var}$ , the problem becomes more complex and more evaluations are needed. Among them all, the NRO algorithm achieves the lowest  $nf^{obj}$  values for most of the analyzed functions. For the Bea, Mat, CT, SHC, DP, Pow ( $n^{var} = 5, 10$ ), SS ( $n^{var} = 2$ ), SDP ( $n^{var} = 2, 5, 10$ ,

20), Sph ( $n^{var} = 2, 5$ ) and Per ( $n^{var} = 2, 20$ ) functions, the NRO algorithm reaches  $n_f^{obj}$  values lower than the ones from the benchmark algorithms with a higher (or similar) success rate. This illustrates the high computational performance of NRO. Nevertheless, for particular cases such as the SS ( $n^{var} = 20$ ) and Per ( $n^{var} = 20$ ) functions, the NRO algorithm produces the highest value of  $n_f^{obj}$ . This, however, is required in order to achieve the highest success rate among the optimization algorithm (except for the ESA algorithm which achieved a greater success rate for the case of SS with  $n^{var} = 20$ ). As expected, the average simulation time has a similar trend as  $n_f^{obj}$ .

At this point it can be noted that, when the number optimization variables increase considerably, the NRO algorithm still achieves the highest success rate by sacrificing its computational time. This is different to the benchmark algorithms that will produce a lower simulation time with a lower success rate.

Based on the average error values, it can be appreciated that in general, the algorithms achieve very low values. This means that their successful results are close to the global minimum. It is worth mentioning that the ESA algorithm produces the lowest simulation errors for most of the benchmark functions reaching even values of zero.

Figure 3 presents the performance of each algorithm based on the number of evaluated optimization variables. These graphs are obtained by averaging the results from Table 2 among the benchmark functions which contain the same number of optimization variables. Here, Fig. 3a reveals that, in general, the NRO algorithm outperforms the other algorithms with respect to the success rate regardless of the number of optimization variables (except for the case of  $n^{var} = 5$  in which the ECTS reaches a slightly higher value). It can also be seen that the PSO algorithm experiences the highest decrease in performance with the increase of  $n^{var}$ . Furthermore, with respect to the number of times the objective function is evaluated, Fig. 3b shows that for  $n^{var} = 2, 5, 10$ , the NRO achieves low  $n_f^{obj}$  values, similar to the other algorithms. Nevertheless, for the case of  $n^{var} = 20$ , a high increase of  $n_f^{obj}$  is observed for NRO. This, as previously explained, is because the optimization problem is more complex with  $n^{var} = 20$  and more evaluations on  $n_f^{obj}$  are required to achieve satisfactory results (this can be seen in Fig. 3a). The average simulation time shown in Fig. 3c presents a similar trend as the one from Fig. 3b due to the high influence that the number of evaluations of the objective function has on the simulation time for the analyzed benchmark functions. With respect to the simulation errors, Fig. 3d reveals that all the successful results achieve low errors. Among them, the PSO algorithm reaches the lowest errors regardless of the number of optimization variables.

## Conclusion

This work proposed a new meta-heuristic optimization technique for single objective optimization problems named Natural Reforestation Optimization (NRO). The NRO, as suggested by its name, is inspired by the natural reforestation process. The characteristics of this algorithm (which enhance the performance of the searching technique) were discussed in this study, among them: The initialization procedure, the exploration and exploitation process, the interaction between the particles, the stopping criteria, among others. The performance of this algorithm was tested with many benchmark functions against four other optimization algorithms. The results revealed that in general, the NRO is capable to generate a high success rate, reaching values close to the global optimum. It was also noted that when the number of optimization variables increased considerably (20 in this study), the success rate of the NRO algorithm would still be superior to the one from the benchmark requiring, however, a higher simulation time. Therefore, the potential of this algorithm was confirmed as a strong optimization tool opening the path for more work to be done to further enhance its performance. As a future work we will aim to further develop this algorithm so that it could also be applied for multi-objective optimization problems.

## Appendix

The benchmark functions used in this study, shown in Table 2, are described in this section<sup>7</sup>.

*Beale Function (Bea):*

$$Bea(x) = (1.5 - x_1 + x_1 \cdot x_2)^2 + (2.25 - x_1 + x_1 \cdot x_2^2)^2 + (2.625 - x_1 + x_1 \cdot x_2^3)^2 \quad (16)$$

The *Bea* function contains two variables and is evaluated in the range of  $x_i \in [-4.5, 4.5]$ , for all  $i = 1, 2$ . Its global minimum is 0 and is located in  $x^* = (3, 0.5)$ .

*Booth Function (Boo):*

$$Boo(x) = (x_1 + 2 \cdot x_2 - 7)^2 + (2 \cdot x_1 + x_2 - 5)^2 \quad (17)$$

The *Boo* function contains two variables and is evaluated in the range of  $x_i \in [-10, 10]$ , for all  $i = 1, 2$ . Its global minimum is 0 and is located in  $x^* = (1, 3)$ .

*Matyas Function (Mat):*

$$Mat(x) = 0.26 \cdot (x_1^2 + x_2^2)^2 - 0.48 \cdot x_1 \cdot x_2 \quad (18)$$

<sup>7</sup> The searching ranges of the optimization variables are obtained from the literature.

The *Mat* function contains two variables and is evaluated in the range of  $x_i \in [-10, 10]$ , for all  $i = 1, 2$ . Its global minimum is 0 and is located in  $x^* = (0, 0)$ .

**Cross-in-Tray Function (CT)**

$$CT(x) = -0.0001 \cdot \left( \left| \sin(x_1) \cdot \sin(x_2) \right| \cdot \exp \left( \left| 100 - \frac{\sqrt{x_1^2 + x_2^2}}{\pi} \right| \right) + 1 \right)^{0.1} \quad (19)$$

The *CT* function contains two variables and is evaluated in the range of  $x_i \in [-10, 10]$ , for all  $i = 1, 2$ . Its global minimum is -2.06261 and is located in  $x^* = (1.3491, -1.3491), (1.3491, 1.3491), (-1.3491, 1.3491), (-1.3491, -1.3491)$ .

**Schaffer Function N.2 (Sch\_N2):**

$$Sch\_N2(x) = 0.5 + \frac{\sin^2(x_1^2 - x_2^2) - 0.5}{(1 + 0.001 \cdot (x_1^2 + x_2^2))^2} \quad (20)$$

The *Sch\_N2* function contains two variables and is evaluated in the range of  $x_i \in [-100, 100]$ , for all  $i = 1, 2$ . Its global minimum is 0 and is located in  $x^* = (0, 0)$ .

**Schaffer Function N.4 (Sch\_N4):**

$$Sch\_N4(x) = 0.5 + \frac{\cos(\sin(|x_1^2 - x_2^2|)) - 0.5}{(1 + 0.001 \cdot (x_1^2 + x_2^2))^2} \quad (21)$$

The *Sch\_N4* function contains two variables and is evaluated in the range of  $x_i \in [-100, 100]$ , for all  $i = 1, 2$ . Its global minimum is 0.292579 and is located in  $x^* = (0, 1.25313)$ .

**Drop-Wave Function (DW):**

$$DW(x) = -\frac{1 + \cos(12 \cdot \sqrt{|x_1^2 - x_2^2|})}{0.5 \cdot (x_1^2 + x_2^2) + 2} \quad (22)$$

The *DW* function contains two variables and is evaluated in the range of  $x_i \in [-5.12, 5.12]$ , for all  $i = 1, 2$ . Its global minimum is -1 and is located in  $x^* = (0, 0)$ .

**Griewank Function (Gri):**

$$Gri(x) = \frac{x_1^2 + x_2^2}{4000} - \cos(x_1) \cdot \cos\left(\frac{x_2}{\sqrt{2}}\right) + 1 \quad (23)$$

The *Gri* function contains two variables and is evaluated in the range of  $x_i \in [-600, 600]$ , for all  $i = 1, 2$ . Its global minimum is 0 and is located in  $x^* = (0, 0)$ .

**Bohachevsky Function 1 (Boh\_f1):**

$$Boh\_f1(x) = x_1^2 + 2 \cdot x_2^2 - 0.3 \cdot \cos(3 \cdot \pi \cdot x_1) - 0.4 \cdot \cos(4 \cdot \pi \cdot x_2) + 0.7 \quad (24)$$

The *Boh\_f1* function contains two variables and is evaluated in the range of  $x_i \in [-100, 100]$ , for all  $i = 1, 2$ . Its global minimum is 0 and is located in  $x^* = (0, 0)$ .

**Bohachevsky Function 2 (Boh\_f2):**

$$Boh\_f2(x) = x_1^2 + 2 \cdot x_2^2 - 0.3 \cdot \cos(3 \cdot \pi \cdot x_1) \cdot \cos(4 \cdot \pi \cdot x_2) + 0.3 \quad (25)$$

The *Boh\_f2* function contains two variables and is evaluated in the range of  $x_i \in [-100, 100]$ , for all  $i = 1, 2$ . Its global minimum is 0 and is located in  $x^* = (0, 0)$ .

**Bohachevsky Function 3 (Boh\_f3):**

$$Boh\_f3(x) = x_1^2 + 2 \cdot x_2^2 - 0.3 \cdot \cos(3 \cdot \pi \cdot x_1 + 4 \cdot \pi \cdot x_2) + 0.3 \quad (26)$$

The *Boh\_f3* function contains two variables and is evaluated in the range of  $x_i \in [-100, 100]$ , for all  $i = 1, 2$ . Its global minimum is 0 and is located in  $x^* = (0, 0)$ .

**Six-Hump Camel Function (SHC):**

$$SHC(x) = \left( 4 - 2.1 \cdot x_1^2 + \frac{x_1^4}{3} \right) \cdot x_1^2 + x_1 \cdot x_2 + (-4 + 4 \cdot x_2^2) \cdot x_2^2 \quad (27)$$

The *SHC* function contains two variables and is evaluated in the range of  $x_1 \in [-3, 3], x_2 \in [-2, 2]$ . Its global minimum is -1.0316 and is located in  $x^* = (0.0898, -0.7126), (-0.0898, 0.7126)$ .

**Dixon-Price Function (DP):**

$$DP(x) = (x_1 - 1)^2 + 2 \cdot (2 \cdot x_2^2 - x_1^2)^2 \quad (28)$$

The *DP* function contains two variables and is evaluated in the range of  $x_i \in [-10, 10]$ , for all  $i = 1, 2$ . Its global minimum is 0 and is located in  $x^* = (1, \sqrt{0.5})$ .

**Powell Function (Pow):**

$$Pow(x) = \sum_{i=1}^{n/4} \left( (x_{4i-3} + 10 \cdot x_{4i-2})^2 + 5 \cdot (x_{4i-1} - x_{4i})^2 + (x_{4i-2} - 2 \cdot x_{4i-1})^4 + 10 \cdot (x_{4i-3} - x_{4i})^4 \right) \quad (29)$$

The *Pow* function contains  $n$  variables and is evaluated in the range of  $x_i \in [-4, 5]$ , for all  $i = 1, \dots, n$ . Its global minimum is 0 and is located in  $x^* = (0, \dots, 0)$ .

*Sum Squares Function (SS):*

$$SS(x) = \sum_{i=1}^n i \cdot x_i^2 \quad (30)$$

The *SS* function contains  $n$  variables and is evaluated in the range of  $x_i \in [-10, 10]$ , for all  $i = 1, \dots, n$ . Its global minimum is 0 and is located in  $x^* = (0, \dots, 0)$ .

*Sum of Different Powers Function (SDP):*

$$SDP(x) = \sum_{i=1}^n |x_i|^{i+1} \quad (31)$$

The *SDP* function contains  $n$  variables and is evaluated in the range of  $x_i \in [-1, 1]$ , for all  $i = 1, \dots, n$ . Its global minimum is 0 and is located in  $x^* = (0, \dots, 0)$ .

*Sphere Function (Sph):*

$$Sph(x) = \sum_{i=1}^n x_i^2 \quad (32)$$

The *Sph* function contains  $n$  variables and is evaluated in the range of  $x_i \in [-5.12, 5.12]$ , for all  $i = 1, \dots, n$ . Its global minimum is 0 and is located in  $x^* = (0, \dots, 0)$ .

*Perm Function 0, d,  $\beta$  (Per):*

$$Per(x) = \sum_{i=1}^n \left( \sum_{j=1}^n (j + \beta) \cdot \left( x_j - \frac{1}{j} \right) \right)^2 \quad (33)$$

The *Per* function contains  $n$  variables and is evaluated in the range of  $x_i \in [-n, n]$ , for all  $i = 1, \dots, n$  with  $\beta = 0$ . Its global minimum is 0 and is located in  $x^* = \left( 1, \frac{1}{2}, \dots, \frac{1}{n} \right)$ .

## Acknowledgment

The authors of this manuscript would like to express their appreciations and gratitude to their respective universities for supporting this research.

## Author's Contributions

All authors equally contributed in this work.

## Ethics

This article is original and contains unpublished material. The corresponding author confirms that all of

the other authors have read and approved the manuscript and there are no ethical issues involved.

## References

- Alvarez-Alvarado, M. S., & Jayaweera, D. (2018a, June). A New Approach for Reliability Assessment of a Static Var Compensator Integrated Smart Grid. In 2018 IEEE International Conference on Probabilistic Methods Applied to Power Systems (PMAPS) (pp. 1-7). IEEE.
- Alvarez-Alvarado, M. S., & Jayaweera, D. (2018b, October). A multi-stage accelerated quantum particle swarm optimization for planning and operation of static var compensators. In 2018 IEEE Third Ecuador Technical Chapters Meeting (ETCM) (pp. 1-6). IEEE.
- Beheshti, Z., & Shamsuddin, S. M. H. (2013). A review of population-based meta-heuristic algorithms. *Int. J. Adv. Soft Comput. Appl.*, 5(1), 1-35.
- Chelouah, R., & Siarry, P. (1999). Enhanced continuous tabu search: An algorithm for optimizing multimodality functions. In *Meta-Heuristics* (pp. 49-61). Springer, Boston, MA.
- Chelouah, R., & Siarry, P. (2000). A continuous genetic algorithm designed for the global optimization of multimodal functions. *Journal of Heuristics*, 6(2), 191-213.
- Clerc, M. (2010). *Particle swarm optimization* (Vol. 93). John Wiley & Sons.
- Cui, L., Li, G., Lin, Q., Chen, J., & Lu, N. (2016). Adaptive differential evolution algorithm with novel mutation strategies in multiple sub-populations. *Computers & Operations Research*, 67, 155-173.
- Dasgupta, D., Yu, S., & Nino, F. (2011). Recent advances in artificial immune systems: models and applications. *Applied Soft Computing*, 11(2), 1574-1587.
- Davis, L. (1991). *Handbook of genetic algorithms*.
- Dorigo, M., & Blum, C. (2005). Ant colony optimization theory: A survey. *Theoretical computer science*, 344(2-3), 243-278.
- Eberhart, R. C., & Shi, Y. (2000, July). Comparing inertia weights and constriction factors in particle swarm optimization. In *Proceedings of the 2000 congress on evolutionary computation. CEC00* (Cat. No. 00TH8512) (Vol. 1, pp. 84-88). IEEE.
- Kantardzic, M. (2003). Genetic algorithms. In: *Data Mining: Concepts, Models, Methods and Algorithms* (pp. 385-413).
- Kohli, M., & Arora, S. (2018). Chaotic grey wolf optimization algorithm for constrained optimization problems. *Journal of computational design and engineering*, 5(4), 458-472.

- Krishnanand, K. R., Nayak, S. K., Panigrahi, B. K., & Rout, P. K. (2009, December). Comparative study of five bio-inspired evolutionary optimization techniques. In 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC) (pp. 1231-1236). IEEE.
- Parsopoulos, K. E., & Vrahatis, M. N. (2010). Particle swarm optimization and intelligence: advances and applications.
- Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2009). GSA: a gravitational search algorithm. *Information sciences*, 179(13), 2232-2248.
- Siarry, P., Berthiau, G., Durdin, F., & Haussy, J. (1997). Enhanced simulated annealing for globally minimizing functions of many-continuous variables. *ACM Transactions on Mathematical Software (TOMS)*, 23(2), 209-228.
- Suman, B., & Kumar, P. (2006). A survey of simulated annealing as a tool for single and multiobjective optimization. *Journal of the operational research society*, 57(10), 1143-1160.
- Yang, X. S. (2011). Review of meta-heuristics and generalised evolutionary walk algorithm. *International Journal of Bio-Inspired Computation*, 3(2), 77-84.