applied sciences

MDPI

Article

DM: Dehghani Method for Modifying Optimization Algorithms

Mohammad Dehghani [1], Zeinab Montazeri [1], Ali Dehghani [2], Haidar Samet [3], Carlos Sotelo [4], David Sotelo [4], Ali Ehsanifar [3], Om Parkash Malik [5], Josep M. Guerrero [6], Gaurav Dhiman [7] and Ricardo A. Ramirez-Mendoza [4,*]

1 Department of Electrical and Electronics Engineering, Shiraz University of Technology, Shiraz 71557-13876, Iran; m.dehghani@sutech.ac.ir (M.D.); Z.Montazeri@sutech.ac.ir (Z.M.)
2 Department of Civil Engineering Islamic Azad Universities of Estahban, Estahban Fars 74518-64747, Iran; adanbax2@gmail.com
3 Department of Power and Control Engineering, School of Electrical and Computer Engineering, Shiraz University, Shiraz 71557-13876, Iran; samet@shirazu.ac.ir (H.S.); a.ehsanifar@shirazu.ac.ir (A.E.)
4 School of Engineering and Sciences, Tecnologico de Monterrey, Monterrey 64849, NL, Mexico; carlos.sotelo@tec.mx (C.S.); david.sotelo@tec.mx (D.S.)
5 Department of Electrical and Computer Engineering, University of Calgary, Calgary, AB T2N 1N4, Canada; maliko@ucalgary.ca
6 CROM Center for Research on Microgrids, Dept. of Energy Technology, Aalborg University, 9220 Aalborg, Denmark; joz@et.aau.dk
7 Department of Computer Science, Government Bikram College of Commerce, Patiala 147004, Punjab, India; gaurav.dhiman@thapar.edu
* Correspondence: ricardo.ramirez@tec.mx; Tel.: +52-81-2001-5597

Received: 21 September 2020; Accepted: 15 October 2020; Published: 30 October 2020

check for updates

Abstract: In recent decades, many optimization algorithms have been proposed by researchers to solve optimization problems in various branches of science. Optimization algorithms are designed based on various phenomena in nature, the laws of physics, the rules of individual and group games, the behaviors of animals, plants and other living things. Implementation of optimization algorithms on some objective functions has been successful and in others has led to failure. Improving the optimization process and adding modification phases to the optimization algorithms can lead to more acceptable and appropriate solution. In this paper, a new method called Dehghani method (DM) is introduced to improve optimization algorithms. DM effects on the location of the best member of the population using information of population location. In fact, DM shows that all members of a population, even the worst one, can contribute to the development of the population. DM has been mathematically modeled and its effect has been investigated on several optimization algorithms including: genetic algorithm (GA), particle swarm optimization (PSO), gravitational search algorithm (GSA), teaching-learning-based optimization (TLBO), and grey wolf optimizer (GWO). In order to evaluate the ability of the proposed method to improve the performance of optimization algorithms, the mentioned algorithms have been implemented in both version of original and improved by DM on a set of twenty-three standard objective functions. The simulation results show that the modified optimization algorithms with DM provide more acceptable and competitive performance than the original versions in solving optimization problems.

Keywords: optimization; Dehghani method; modifying; optimization algorithm; population-based algorithm

1. Introduction

The purpose of optimization is to determine the best solution among the available solutions for an optimization problem according to the constraints of problem [1]. Each optimization problem is designed with three parts: constraints, objective functions, and decision variables [2]. There are many optimization problems in different sciences that should be optimized using the appropriate method. Stochastic search-based optimization algorithms have always been of interest to researchers in solving optimization problems [3]. Optimization algorithms are able to provide a quasi-optimal solution based on random scan of the search space instead of a full scan. The quasi-optimal solution is not the best solution, but it is close to the global optimal solution of the problem [1]. In this regard, optimization algorithms have been applied by scientists in various fields such as energy [4–6], protection [7], electrical engineering [8–13], topology optimization [14] and energy carriers [15–17] to achieve the quasi-optimal solution. Table 1 shows the optimization algorithms grouped according to the main design idea.

Table 1. Optimization algorithms.

Optimization Algorithms	Swarm-based	General description: Designed based on simulation of the living thing behavior processes of the plants, and other swarm-based phenomena.	
		• Particle Swarm Optimization (PSO) [18], Ant Colony Optimization (ACO) [19,20], Spotted Hyena Optimizer (SHO) [21], Group Optimization (GO) [22], Artificial Bee Colony (ABC) [23], Following Optimization Algorithm (FOA) [24], Rat Swarm Optimizer (RSO) [2], Multi Leader Optimizer (MLO) [1], Bat-inspired Algorithm (BA) [25], Emperor Penguin Optimizer (EPO) [26], Cuckoo Search (CS) [27], Donkey Theorem Optimization (DTO) [28], Teaching-Learning-Based Optimization (TLBO) Algorithm [29], Grasshopper Optimization Algorithm (GOA) [30], Doctor and Patient Optimization (DPO) [31], Gray Wolf Optimizer (GWO) [32]	Ref. [18] The most widely used algorithm in this group, which is designed based on modeling the movement of birds. Refs. [19,20] It is based on the modeling the discovery of the shortest path by ants. Ref. [29] It has gained wide acceptance among the optimization researchers. This algorithm is a teaching-learning process inspired algorithm and is based on the effect of influence of a teacher on the output of learners in a class. Ref. [31] It is designed by simulating the process of treating patients by a physician. The treatment process has three phases, including vaccination, drug administration, and surgery. Ref. [32] It mimics the leadership hierarchy and hunting mechanism of grey wolves in nature. Four types of grey wolves such as alpha, beta, delta, and omega are employed for simulating the leadership hierarchy.
	Game-based	General description: Designed based on simulation of different processes and rules of individual and group games.	
		• Football Game-Based Optimization (FGBO) [3], Hide Objects Game Optimization (HOGO) [33], Orientation Search Algorithm (OSA) [34,35], Dice Game Optimizer (DGO) [36], Shell Game Optimization (SGO) [37], Darts Game Optimizer (DGO) [38]	Ref. [3] It is designed based on mathematical modeling of football league rules and behaviors of football players and clubs. Ref. [33] It is based on the simulation of the players behaviors and their trying to find a hidden object in the hide object game.

Table 1. *Cont.*

Physics-based	General description: Designed based on the ideation of various laws of physics.	
	• Spring Search Algorithm (SSA) [39,40], Curved Space Optimization (CSO) [41], Black Hole (BH) [42], Ray Optimization (RO) [43] algorithm, Artificial Chemical Reaction Optimization Algorithm (ACROA) [44], Galaxy-based Search Algorithm (GbSA) [45], and Small World Optimization Algorithm (SWOA) [46]	Refs. [39,40] The simulation of the Hooke's law between a number of weights and springs is used.
Evolutionary-based	General description: They have involved evolution of a population in order to create new generations of genetically superior individuals [47].	
	• Biogeography-based Optimizer (BBO) [48], Differential Evolution (DE) [49], Genetic Algorithm (GA) [50], Evolution Strategy (ES) [51], and Genetic Programming (GP) [52].	Ref. [50] It has found wide acceptance in many disciplines, with application to environmental science problems. This algorithm is an optimization tool that mimics natural selection and genetics.

Each optimization problem has a definite solution called a global solution. Optimization algorithms provide a solution based on random search of the search space, which is not necessarily a universal solution, but because it is close to the optimal solution, it is an acceptable solution. The solution that is provided by optimization algorithms is called quasi-optimal solution. Therefore, an optimization algorithm that offers a better quasi-optimal solution than another algorithm is a better optimizer algorithm. In this regard, many optimization algorithms have been proposed by researchers to solve optimization problems and achieve to the better quasi-optimal solution.

Although optimization algorithms have been successful in solving many optimization problems, improving the equations of optimization algorithms and adding modification phases to optimization algorithms can lead to better quasi-optimal solutions. In fact, the purpose of improving an optimization algorithm is to increase the ability of that algorithm to more accurately scan the problem search space and thus provide a more appropriate quasi-optimal solution and closer to the global optimal solution.

In this paper, a new modification method called Dehghani method (DM) is proposed to improve the performance of optimization algorithms. DM is designed based on the use of the algorithm population members information. In the proposed DM, the information of each population member can improve the situation of the new generation. The main idea of DM is to amplify the best population member of an optimization algorithm using population member information. The proposed method is fully described in the next section.

The continuation of the present article is organized in such a way that in Section 2, the DM is fully explained and modeled. Following this, Section 3 explains how to implement the proposed method on several algorithms. The simulation of the proposed method for solving optimization problems is presented in Section 4. Finally, conclusions and several suggestions for future studies are presented in Section 5.

2. Dehghani Method (DM)

In this section, first DM is explained and then its mathematical modeling is presented. DM shows that all population members of the optimization algorithm, even the worst one, can contribute to the development of the population of algorithm.

Each population-based optimization algorithm has a matrix called the population matrix, which each row of this matrix represents a population member. Each member of the population is actually a vector which represents the values of the problem variables. Given that each member of the population is a random vector in the problem search space, it is a suggested solution (SS) to the problem. After the formation of the population matrix, the values proposed by each population member for the problem variables are evaluated in the objective function (OF). The population matrix and values of the objective functions are defined in Equation (1).

$$\mathrm{SS} = X = \begin{bmatrix} SS_1 = X_1 & x_1^1 & \cdots & x_1^d & \cdots & x_1^m & OF_1 \\ \vdots & \vdots & \ddots & \vdots & & \vdots & \vdots \\ SS_i = X_i & x_i^1 & \cdots & x_i^d & \cdots & x_i^m & OF_i \\ \vdots & \vdots & & \vdots & \ddots & \vdots & \vdots \\ SS_N = X_N & x_N^1 & \cdots & x_N^d & \cdots & x_N^m & OF_N \end{bmatrix}, \tag{1}$$

where, SS is the suggested solutions matrix, X is the population matrix, $\mathrm{SS_i}$ is the i'th suggested solution, X_i is the i'th population member, x_i^d is the value of d'th variables of optimization problem suggested by i'th population member, N is the number of population members or suggested solutions, m is the number of variables, and OF_i is the value of objective function for the i'th suggested solution.

Different values for the objective function are obtained based on the values proposed for the variables by the population members. The member that offers the best-suggested solution (BSS) to the optimization problem plays an important role in improving the algorithm population. The row number of this member in the population matrix is determined using Equation (2).

$$best = \begin{cases} the\ row\ number\ of\ X\ matrix\ with \min OF, & for\ minimization\ problems \\ the\ row\ number\ of\ X\ matrix\ with \max OF, & for\ maximization\ problems \end{cases}, \tag{2}$$

where, *best* is the row number of the member with BSS. The BSS and it's *OF* are specified by Equations (3) and (4).

$$X_{best}\ : variables\ value\ of \min objective\ function, \tag{3}$$

$$F_{best}\ : value\ of \min objective\ function, \tag{4}$$

where, X_{best} is the BSS or best population member and F_{best} is the value of *OF* for BSS.

As mentioned, the best member of the population plays an important role in improving the population of the algorithm and thus the performance of the optimization algorithm. Optimization algorithms update the status of its population members according to its own process to achieve a quasi-optimal solution. Accordingly, with the improvement of the best member of the population, is expected that the population be updated more effectively and the performance of the algorithm in solving the optimization problem is improved.

DM is designed with the idea of modifying the best population member with the aim of improving the performance of an optimization algorithm.

In DM, just as the best member is influential in updating the population members, the population members even the worst member can influence to modification the best member. The measurement criterion for suggested solutions is the value of the objective function. However, a suggested solution that is not the best solution may provide appropriate values for some problem variables. The proposed DM modifies the best member considering this issue and using the values suggested by other of the population members. This concept is mathematically simulated in Equations (5) and (6).

$$X_{DM} = X_{DM}^{i,d} = \left[x_{best}^1 \ \cdots \ x_i^d \ \cdots \ x_{best}^m \right], \tag{5}$$

$$X_{best}^{new} = \begin{cases} X_{DM}, & OF(X_{DM}^{i,d}) < F_{best} \\ X_{best}, & else \end{cases}, \tag{6}$$

where, X_{DM} is the modified best member by DM, $X_{DM}^{i,d}$ is the modified best member based on the suggested value for d'th variable by i'th SS, X_{best}^{new} is the new status for best member based on DM, and $OF(X_{DM}^{i,d})$ is the objective function value for modified best member by DM.

The pseudo code of DM is presented in Algorithm 1. In addition, the different stages of the proposed method with the aim of improving the best member are shown as a flowchart in Figure 1.

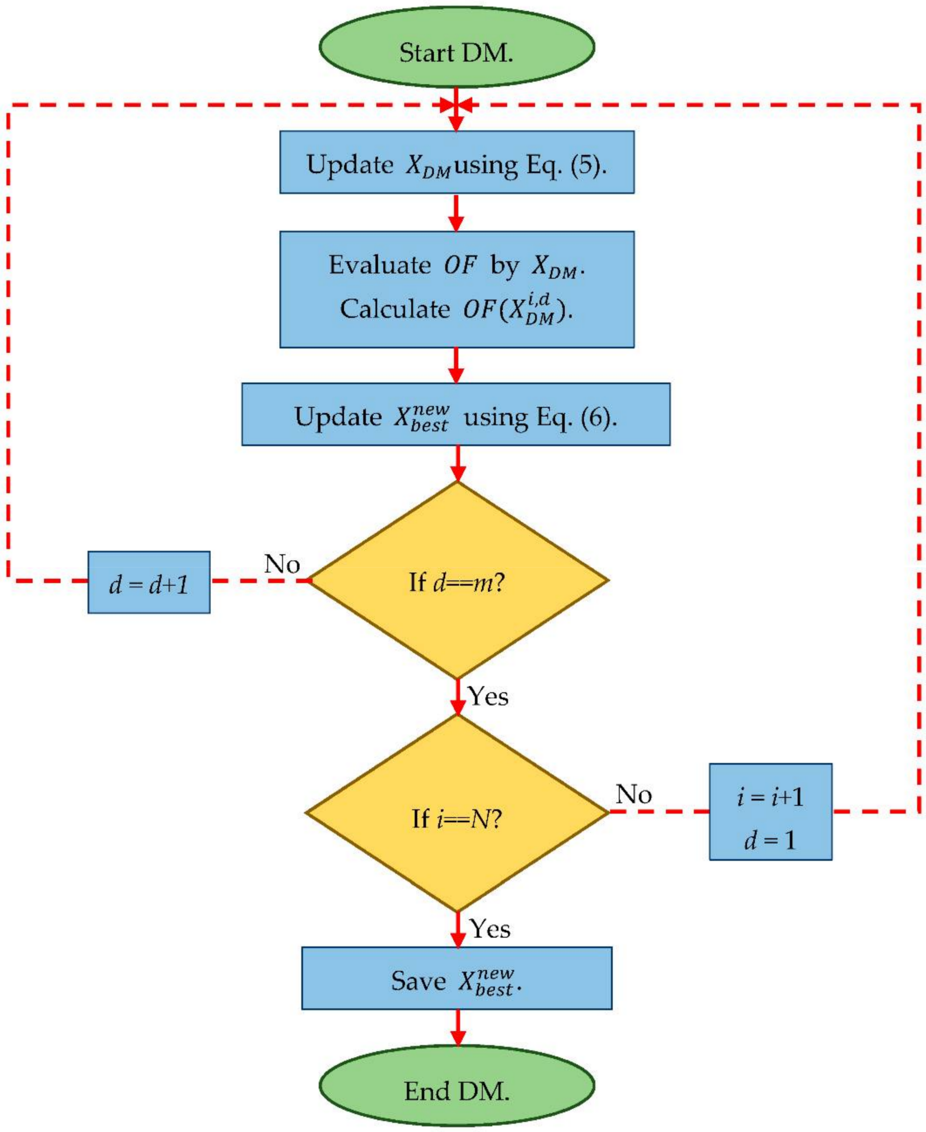

Figure 1. Flowchart of Dehghani method (DM).

Algorithm 1. Pseudo code of DM

For i = 1:$N_{populaion}$
  For d = 1:m
    Update X_{DM} using Equation (5).
    Calculate $OF(X_{DM}^{i,d})$.
    Update X_{best}^{new} using Equation (6):
    If $OF(X_{DM}^{i,d}) < F_{best}$
      $X_{best}^{new} = X_{DM}$
    End if
  End for d
End: for i

$N_{populaion}$: number of population members.
m: number of variables.

3. DM Implantation on Optimization Algorithms

This section describes how to implement the proposed DM on optimization algorithms. The proposed DM is applicable to modify population-based optimization algorithms. Although the idea of designing optimization algorithms is different, the procedure is the same. These algorithms provide a quasi-optimal solution starting from a random initial population and following a process based on repetition and population updates in each iteration.

The pseudo code of implantation of the DM for modifying optimization algorithms is presented in Algorithm 2. The steps of the modified version of the optimization algorithms using DM are shown in Figure 2.

Algorithm 2. Pseudo code of implantation of the DM for modifying optimization algorithms

Start.
Set parameters.
Input: m, OF, constraints.
Create initial population.
Create another matrix (if there are).
For t = 1: $iteration_{max}$
  Calculate OF.
  Find X_{best}.
  DM toolbox:
    Update X_{best}^{new} based on DM.
  Continue the processes of optimization algorithm.
  Update population.
End for t
Output: BSS.
End.

$iteration_{max}$: maximum number of iterations.

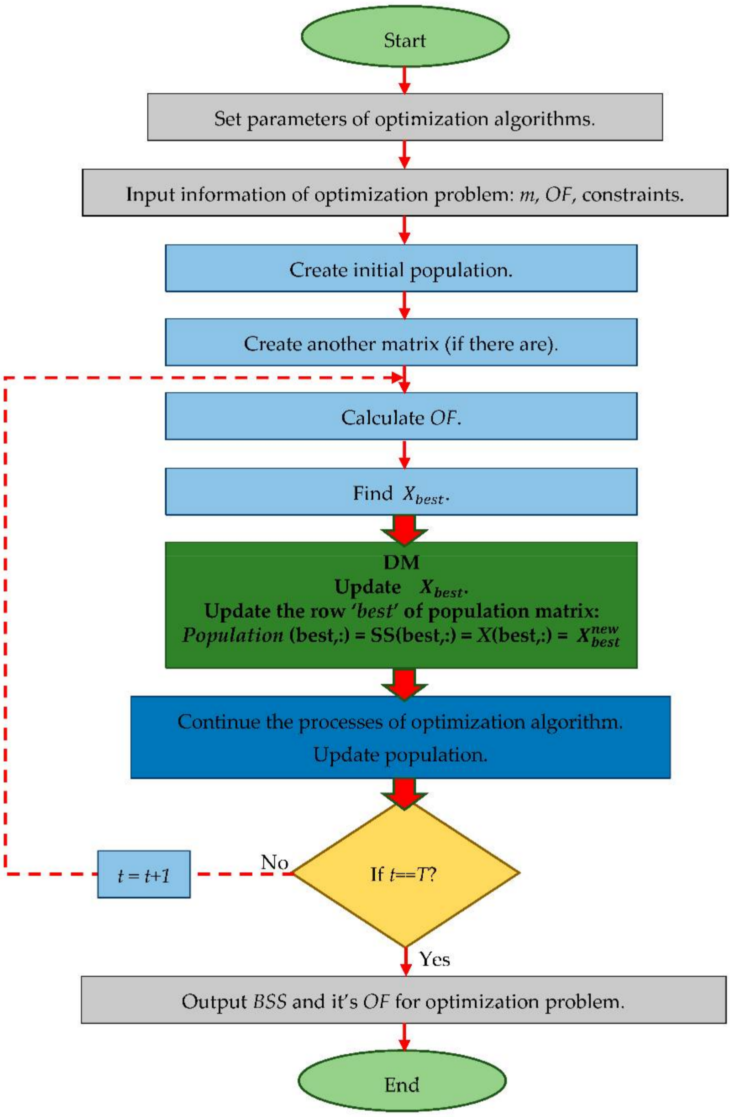

Figure 2. Flowchart of implantation of the DM for modifying optimization algorithms.

4. Simulation and Discussion

In this section, the performance of the proposed DM in improving optimization algorithms is evaluated. Thus, the present work and the optimization algorithms described in [18,29,32,50,53] are developed using the same computational platform: Matlab R2014a (8.3.0.532) version in the environment of Microsoft Windows 10 with 64 bits on Core i-7 processor with 2.40 GHz and 6 GB memory. To generate and report the results, for each objective function, optimization algorithms utilize 20 independent runs where each run employs 1000 times of iterations.

4.1. Algorithms Used for Comparisons and Benchmark Test Functions

To evaluate the performance of the proposed DM, the following methodology is applied:

(1) Find in the literature five well-known optimization algorithms, such as: genetic algorithm (GA) [50], particle swarm optimization (PSO) [18], gravitational search algorithm (GSA) [53], teaching learning based optimization (TLBO) [29] and grey wolf optimizer (GWO) [32].
(2) Modify the optimization algorithms implementing the proposed DM.

(3) Define the set of twenty-three objective functions and divide it into three main categories: unimodal [53,54], multimodal [31,54], and fixed-dimension multimodal [54] functions (see Appendix A).
(4) Implement the present work and the optimization algorithms in the same computational platform.
(5) Compare the performance of the modified and the original optimization algorithms using the following metrics: the average and the standard deviation of the best obtained optimal solution till the last iteration is computed.

4.2. Results

Optimization algorithms in the original version and the modified version, using the proposed DM, are implemented on the objective functions. The simulation results are presented from Tables 2–6 for three different categories: unimodal, multimodal, and fixed-dimension multimodal functions. The first category consists of seven objective functions, F_1 to F_7, the second category consists of six objective functions, F_8 to F_{13}, and the third category consists of ten objective functions, F_{14} to F_{23}.

To further analyze the simulation results, the convergence curves of the optimization algorithms for the twenty-three objective functions are shown from Figures 3–7. In these figures, the convergence curves for the original and the modified versions are plotted simultaneously.

Computational time analysis in accessing the optimal solution is presented in Tables 7–9. This analysis shows computational time for per iteration, per complete run, and the overall time required for the original and modified algorithm to achieve similar objective function value. In these tables, P.I. means per iteration, P.C. means per complete run, and the O.T.S. means overall time required for the original and modified algorithm to achieve similar objective function value.

Table 2. Optimization results of genetic algorithm (GA) for original and "modified by DM" versions.

		Unimodal				Multimodal				Fixed-Dimension Multimodal	
		Original	**DM**			**Original**	**DM**			**Original**	**DM**
F_1	Avg	13.2405	**1.6151×10^{-11}**	F_8	Avg	−8184.4142	**−12569.4850**	F_{14}	Avg	0.9986	**0.9980**
	std	4.7664×10^{-15}	**1.1560×10^{-26}**		std	8.3381×10^{-12}	**1.2202×10^{-21}**		std	1.5640×10^{-15}	**3.4755×10^{-16}**
F_2	Avg	2.4794	**4.5161×10^{-35}**	F_9	Avg	62.4114	**0.0019**	F_{15}	Avg	5.3952×10^{-2}	**3.3882×10^{-03}**
	std	2.2342×10^{-15}	**7.1717×10^{-51}**		std	2.5421×10^{-14}	**3.8789×10^{-19}**		std	7.0791×10^{-18}	**2.0364×10^{-18}**
F_3	Avg	1536.8963	**2.0620×10^{-16}**	F_{10}	Avg	3.2218	**0.0127**	F_{16}	Avg	−1.0316	**−1.0316**
	std	6.6095×10^{-13}	**6.6148×10^{-32}**		std	5.1636×10^{-15}	**0**		std	7.9441×10^{-16}	**5.9580×10^{-16}**
F_4	Avg	2.0942	**6.6058×10^{-15}**	F_{11}	Avg	1.2302	**0.0272**	F_{17}	Avg	0.4369	**0.3984**
	std	2.2342×10^{-15}	**8.1141×10^{-30}**		std	8.4406×10^{-16}	**3.1031×10^{-18}**		std	4.9650×10^{-17}	**4.9650×10^{-17}**
F_5	Avg	310.4273	**5.9802**	F_{12}	Avg	0.0470	**8.2396×10^{-7}**	F_{18}	Avg	4.3592	**3.0000**
	std	2.0972×10^{-13}	**6.3552×10^{-25}**		std	4.6547×10^{-18}	**3.3145×10^{-22}**		std	5.9580×10^{-16}	**2.0853×10^{-15}**
F_6	Avg	14.55	**0**	F_{13}	Avg	1.2085	**2.8065×10^{-5}**	F_{19}	Avg	-3.85434	**−3.8627**
	std	3.1776×10^{-15}	**0**		std	3.2272×10^{-16}	**2.1213×10^{-20}**		std	9.9301×10^{-17}	**9.9301×10^{-16}**
F_7	Avg	5.6799×10^{-03}	**2.6009×10^{-5}**					F_{20}	Avg	−2.8239	**−3.2387**
	std	7.7579×10^{-19}	**1.6364×10^{-29}**						std	3.97205×10^{-16}	**1.7874×10^{-15}**
								F_{21}	Avg	−4.3040	**−10.1522**
									std	1.5888×10^{-15}	**1.5888×10^{-15}**
								F_{22}	Avg	−5.1174	**−10.4016**
									std	1.2909×10^{-15}	**1.6682×10^{-14}**
								F_{23}	Avg	−6.5621	**−10.5362**
									std	3.8727×10^{-15}	**7.5469×10^{-15}**

Table 3. Optimization results of particle swarm optimization (PSO) for original and "modified by DM" versions.

		Unimodal				**Multimodal**				**Fixed-Dimension Multimodal**	
		Original	**DM**			**Original**	**DM**			**Original**	**DM**
F_1	Avg	1.7740×10^{-5}	$\mathbf{6.746 \times 10^{-218}}$	F_8	Avg	-6908.6558	$\mathbf{-12516.1893}$	F_{14}	Avg	2.1735	**0.9980**
	std	6.4396×10^{-21}	**0**		std	1.0168×10^{-12}	$\mathbf{9.3549 \times 10^{-19}}$		std	7.9441×10^{-16}	$\mathbf{5.4615 \times 10^{-19}}$
F_2	Avg	0.3411	$\mathbf{2.9565 \times 10^{-111}}$	F_9	Avg	57.0613	$\mathbf{1.5631 \times 10^{-13}}$	F_{15}	Avg	0.0535	**0.0034**
	std	7.4476×10^{-17}	$\mathbf{1.0736 \times 10^{-126}}$		std	6.3552×10^{-15}	**0**		std	3.8789×10^{-19}	$\mathbf{2.0849 \times 10^{-18}}$
F_3	Avg	589.4920	$\mathbf{1.9627 \times 10^{-70}}$	F_{10}	Avg	2.1546	$\mathbf{4.7784 \times 10^{-14}}$	F_{16}	Avg	-1.0316	$\mathbf{-1.0316}$
	std	7.1179×10^{-13}	$\mathbf{5.0357 \times 10^{-86}}$		std	7.9441×10^{-16}	$\mathbf{1.1289 \times 10^{-29}}$		std	3.4755×10^{-16}	$\mathbf{4.4685 \times 10^{-21}}$
F_4	Avg	3.9634	$\mathbf{1.5239 \times 10^{-97}}$	F_{11}	Avg	0.0462	**0.0214**	F_{17}	Avg	0.7854	**0.4076**
	std	1.9860×10^{-16}	$\mathbf{5.8112 \times 10^{-113}}$		std	3.1031×10^{-18}	$\mathbf{3.1031 \times 10^{-23}}$		std	4.9650×10^{-17}	**0**
F_5	Avg	50.26245	$\mathbf{2.3706 \times 10^{-13}}$	F_{12}	Avg	0.4806	$\mathbf{1.5705 \times 10^{-32}}$	F_{18}	Avg	3	**3**
	std	1.5888×10^{-14}	$\mathbf{1.9191 \times 10^{-28}}$		std	1.8619×10^{-16}	$\mathbf{1.2239 \times 10^{-47}}$		std	3.6741×10^{-15}	$\mathbf{2.5818 \times 10^{-19}}$
F_6	Avg	20.25	**0**	F_{13}	Avg	0.5084	$\mathbf{1.3497 \times 10^{-32}}$	F_{19}	Avg	-3.8627	$\mathbf{-3.8627}$
	std	0	**0**		std	4.9650×10^{-17}	$\mathbf{1.2239 \times 10^{-47}}$		std	8.9371×10^{-15}	$\mathbf{9.0364 \times 10^{-21}}$
F_7	Avg	0.1134	**0.0110**					F_{20}	Avg	-3.2619	$\mathbf{-3.2744}$
	std	4.3444×10^{-17}	$\mathbf{1.2412 \times 10^{-27}}$						std	2.9790×10^{-16}	$\mathbf{3.9720 \times 10^{-21}}$
								F_{21}	Avg	-5.3891	$\mathbf{-10.1532}$
									std	1.4895×10^{-15}	$\mathbf{3.4755 \times 10^{-15}}$
								F_{22}	Avg	-7.6323	$\mathbf{-10.4029}$
									std	1.5888×10^{-15}	$\mathbf{1.2909 \times 10^{-15}}$
								F_{23}	Avg	-6.1648	$\mathbf{-10.5364}$
									std	2.7804×10^{-15}	$\mathbf{7.6462 \times 10^{-15}}$

Table 4. Optimization results of gravitational search algorithm (GSA) for original and "modified by DM" versions.

		Unimodal				Multimodal				Fixed-Dimension Multimodal	
		Original	**DM**			**Original**	**DM**			**Original**	**DM**
F_1	Avg	2.0255×10^{-17}	$\mathbf{1.6060 \times 10^{-157}}$	F_8	Avg	-2849.0724	$\mathbf{-12532.65497}$	F_{14}	Avg	3.5913	**0.9980**
	std	1.1369×10^{-32}	**0**		std	1.0168 × 10-12	**2.84717 × 10-12**		std	7.9441×10^{-16}	$\mathbf{4.2203 \times 10^{-16}}$
F_2	Avg	2.3702×10^{-08}	$\mathbf{2.4203 \times 10^{-80}}$	F_9	Avg	16.2675	**0**	F_{15}	Avg	0.0024	$\mathbf{8.0939 \times 10^{-4}}$
	std	5.1789×10^{-24}	$\mathbf{8.3748 \times 10^{-96}}$		std	3.1776×10^{-15}	**0**		std	2.9092×10^{-19}	$\mathbf{3.5153 \times 10^{-28}}$
F_3	Avg	279.3439	$\mathbf{1.4902 \times 10^{-30}}$	F_{10}	Avg	3.5673×10^{-09}	$\mathbf{4.3396 \times 10^{-13}}$	F_{16}	Avg	-1.0316	$\mathbf{-1.0316}$
	std	1.2075×10^{-13}	$\mathbf{5.4834 \times 10^{-46}}$		std	3.6992×10^{-25}	$\mathbf{9.0314 \times 10^{-29}}$		std	5.9580×10^{-16}	$\mathbf{6.4545 \times 10^{-34}}$
F_4	Avg	3.2547×10^{-9}	$\mathbf{7.1301 \times 10^{-71}}$	F_{11}	Avg	3.7375	**0.0311**	F_{17}	Avg	0.3978	**0.3978**
	std	2.0346×10^{-24}	$\mathbf{3.5969 \times 10^{-87}}$		std	2.7804×10^{-15}	**0**		std	9.9301×10^{-17}	**0**
F_5	Avg	36.10695	**23.1212**	F_{12}	Avg	0.0362	$\mathbf{2.319 \times 10^{-27}}$	F_{18}	Avg	3	**3**
	std	3.0982×10^{-14}	$\mathbf{5.5608 \times 10^{-15}}$		std	6.2063×10^{-18}	$\mathbf{1.0829 \times 10^{-42}}$		std	6.9511×10^{-16}	$\mathbf{1.2909 \times 10^{-35}}$
F_6	Avg	0	**0**	F_{13}	Avg	0.0020	$\mathbf{2.5758 \times 10^{-26}}$	F_{19}	Avg	-3.8627	$\mathbf{-3.8627}$
	std	0	**0**		std	4.2617×10^{-14}	$\mathbf{7.7006 \times 10^{-42}}$		std	8.3413×10^{-15}	$\mathbf{6.3248 \times 10^{-29}}$
F_7	Avg	0.0206	$\mathbf{7.4501 \times 10^{-4}}$					F_{20}	Avg	-3.0396	$\mathbf{-3.3219}$
	std	2.7152×10^{-18}	$\mathbf{1.9394 \times 10^{-29}}$						std	2.1846×10^{-14}	$\mathbf{1.9860 \times 10^{-25}}$
								F_{21}	Avg	-5.1486	$\mathbf{-10.1531}$
									std	2.9790×10^{-16}	$\mathbf{1.1916 \times 10^{-24}}$
								F_{22}	Avg	-9.0239	$\mathbf{-10.4029}$
									std	1.6484×10^{-12}	$\mathbf{1.3505 \times 10^{-34}}$
								F_{23}	Avg	-8.9045	$\mathbf{-10.5364}$
									std	7.1497×10^{-14}	$\mathbf{5.95808 \times 10^{-45}}$

Table 5. Optimization results of teaching learning based optimization (TLBO) for original and "modified by DM" versions.

		Unimodal				**Multimodal**				**Fixed-Dimension Multimodal**	
		Original	**DM**			**Original**	**DM**			**Original**	**DM**
F_1	Avg	8.3373×10^{-60}	**1.1627×10^{-157}**	F_8	Avg	-7408.6107	**-12569.4866**	F_{14}	Avg	2.2721	**0.9980**
	std	4.9436×10^{-76}	**0**		std	3.0505×10^{-12}	**1.6269×10^{-11}**		std	1.9860×10^{-16}	**0**
F_2	Avg	7.1704×10^{-35}	**1.9426×10^{-80}**	F_9	Avg	10.2485	**0**	F_{15}	Avg	0.0033	**3.9560×10^{-4}**
	std	6.6936×10^{-50}	**1.2562×10^{-96}**		std	5.5608×10^{-15}	**0**		std	1.2218×10^{-17}	**0**
F_3	Avg	2.7531×10^{-15}	**1.0076×10^{-30}**	F_{10}	Avg	0.2757	**4.7961×10^{-15}**	F_{16}	Avg	-1.0316	**-1.0316**
	std	2.6459×10^{-31}	**5.8751×10^{-46}**		std	2.5641×10^{-15}	**1.4111×10^{-30}**		std	1.4398×10^{-15}	**9.9301×10^{-19}**
F_4	Avg	9.4199×10^{-15}	**5.6754×10^{-71}**	F_{11}	Avg	0.6082	**0.0182**	F_{17}	Avg	0.3978	**0.4085**
	std	2.1167×10^{-30}	**2.8775×10^{-86}**		std	1.9860×10^{-16}	**6.2063×10^{-18}**		std	7.4476×10^{-17}	**9.9301×10^{-17}**
F_5	Avg	146.4564	**21.4361**	F_{12}	Avg	0.0203	**1.5705×10^{-32}**	F_{18}	Avg	3.0009	**3**
	std	1.9065×10^{-14}	**2.0654×10^{-21}**		std	7.7579×10^{-19}	**1.2239×10^{-47}**		std	1.5888×10^{-15}	**1.3902×10^{-26}**
F_6	Avg	0.4435	**0**	F_{13}	Avg	0.3293	**1.3497×10^{-32}**	F_{19}	Avg	-3.8609	**-3.8627**
	std	4.2203×10^{-16}	**0**		std	2.1101×10^{-16}	**1.2239×10^{-47}**		std	7.3483×10^{-15}	**9.9301×10^{-45}**
F_7	Avg	0.0017	**3.4102×10^{-4}**					F_{20}	Avg	-3.2014	**-3.3104**
	std	3.87896×10^{-19}	**2.4849×10^{-27}**						std	1.7874×10^{-15}	**9.9301×10^{-18}**
								F_{21}	Avg	-9.1746	**-10.1531**
									std	8.5399×10^{-15}	**0**
								F_{22}	Avg	-10.0389	**-10.4029**
									std	1.5292×10^{-14}	**0**
								F_{23}	Avg	-9.2905	**-10.5364**
									std	1.1916×10^{-15}	**0**

Table 6. Optimization results of grey wolf optimization (GWO) for original and "modified by DM" versions.

		Unimodal				**Multimodal**				**Fixed-Dimension Multimodal**	
		Original	**DM**			**Original**	**DM**			**Original**	**DM**
F_1	Avg	1.09×10^{-58}	$\mathbf{2.84 \times 10^{-278}}$	F_8	Avg	-5885.1172	$\mathbf{-11901.9832}$	F_{14}	Avg	3.7408	**1.3948**
	std	5.1413×10^{-74}	**0**		std	2.0336×10^{-12}	$\mathbf{4.8808 \times 10^{-14}}$		std	6.4545×10^{-15}	$\mathbf{8.44062 \times 10^{-16}}$
F_2	Avg	1.2952×10^{-34}	$\mathbf{1.6523 \times 10^{-137}}$	F_9	Avg	8.5265×10^{-15}	**0**	F_{15}	Avg	0.0063	**0.0043**
	std	1.9127×10^{-50}	$\mathbf{1.2275 \times 10^{-152}}$		std	5.6446×10^{-30}	**0**		std	1.1636×10^{-18}	$\mathbf{3.10317 \times 10^{-28}}$
F_3	Avg	7.4091×10^{-15}	$\mathbf{1.0362 \times 10^{-30}}$	F_{10}	Avg	1.7053×10^{-14}	$\mathbf{1.0835 \times 10^{-14}}$	F_{16}	Avg	-1.0316	$\mathbf{-1.0316}$
	std	5.6446×10^{-30}	$\mathbf{1.2533 \times 10^{-45}}$		std	2.7517×10^{-29}	$\mathbf{2.8223 \times 10^{-30}}$		std	3.9720×10^{-16}	$\mathbf{5.9580 \times 10^{-26}}$
F_4	Avg	1.2599×10^{-14}	$\mathbf{2.5914 \times 10^{-47}}$	F_{11}	Avg	0.0037	**0.0014**	F_{17}	Avg	0.3978	**0.3978**
	std	1.0583×10^{-29}	$\mathbf{2.1742 \times 10^{-63}}$		std	1.2606×10^{-18}	**0**		std	8.6888×10^{-17}	$\mathbf{1.2412 \times 10^{-19}}$
F_5	Avg	26.8607	**4.9282**	F_{12}	Avg	0.0372	$\mathbf{1.0468 \times 10^{-09}}$	F_{18}	Avg	3.0000	**3.0000**
	std	0	**0**		std	4.3444×10^{-17}	$\mathbf{3.2368 \times 10^{-25}}$		std	2.0853×10^{-15}	$\mathbf{2.0853 \times 10^{-18}}$
F_6	Avg	0.6423	$\mathbf{9.6762 \times 10^{-09}}$	F_{13}	Avg	0.5763	$\mathbf{9.4403 \times 10^{-09}}$	F_{19}	Avg	-3.8621	$\mathbf{-3.8627}$
	std	6.2063×10^{-17}	$\mathbf{7.3985 \times 10^{-24}}$		std	2.4825×10^{-16}	$\mathbf{3.6992 \times 10^{-24}}$		std	2.4825×10^{-15}	**0**
F_7	Avg	0.0008	**0.0005**					F_{20}	Avg	-3.2523	$\mathbf{-3.2982}$
	std	7.2730×10^{-20}	$\mathbf{1.9394 \times 10^{-29}}$						std	2.1846×10^{-15}	$\mathbf{1.8867 \times 10^{-30}}$
								F_{21}	Avg	-9.6452	$\mathbf{-10.1531}$
									std	6.5538×10^{-15}	$\mathbf{1.1916 \times 10^{-32}}$
								F_{22}	Avg	-10.4025	$\mathbf{-10.4029}$
									std	1.9860×10^{-15}	$\mathbf{1.1519 \times 10^{-25}}$
								F_{23}	Avg	-10.1302	$\mathbf{-10.5364}$
									std	4.5678×10^{-15}	$\mathbf{2.7804 \times 10^{-20}}$

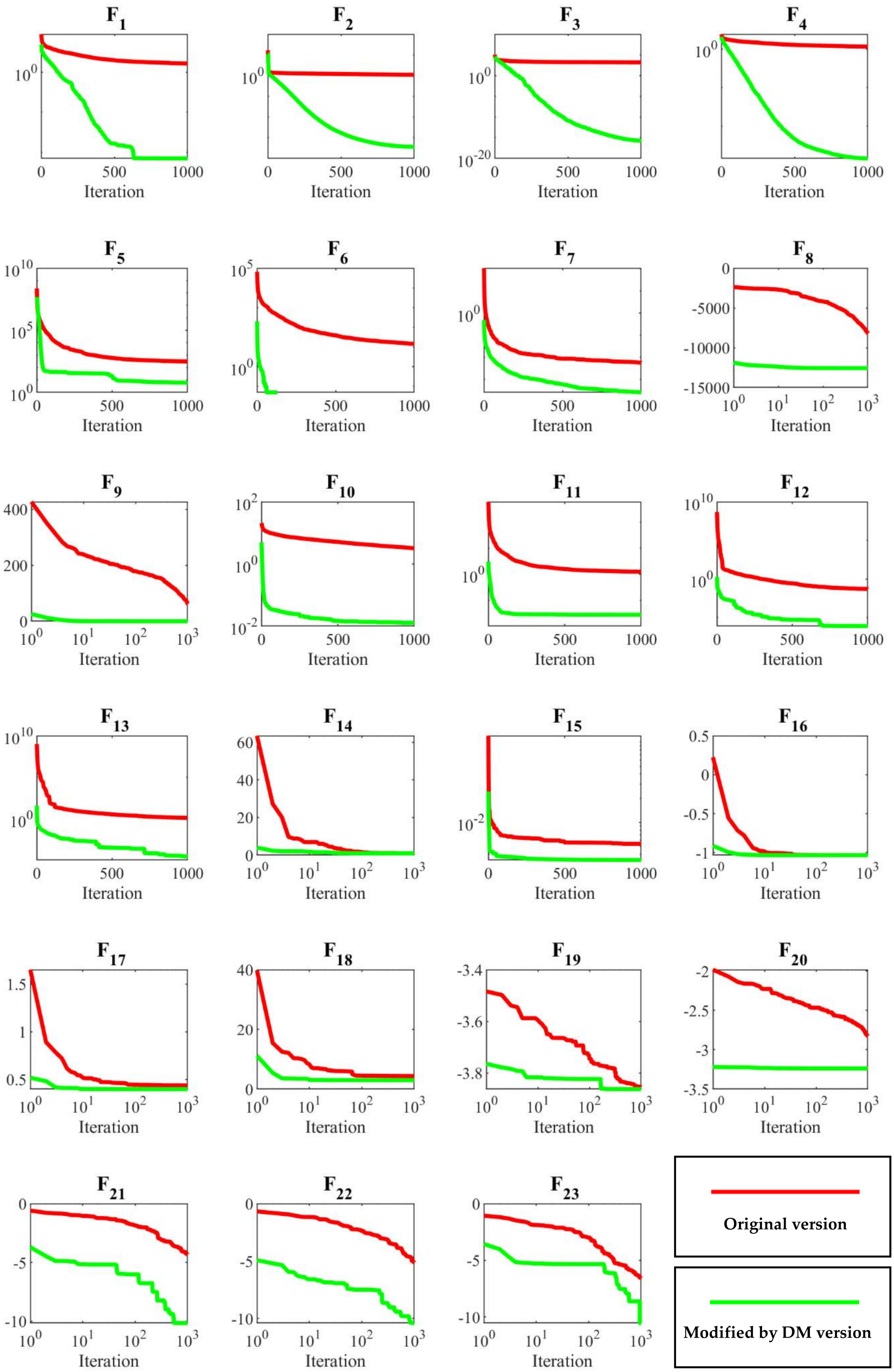

Figure 3. The convergence curves of GA for original and "modified by DM" versions.

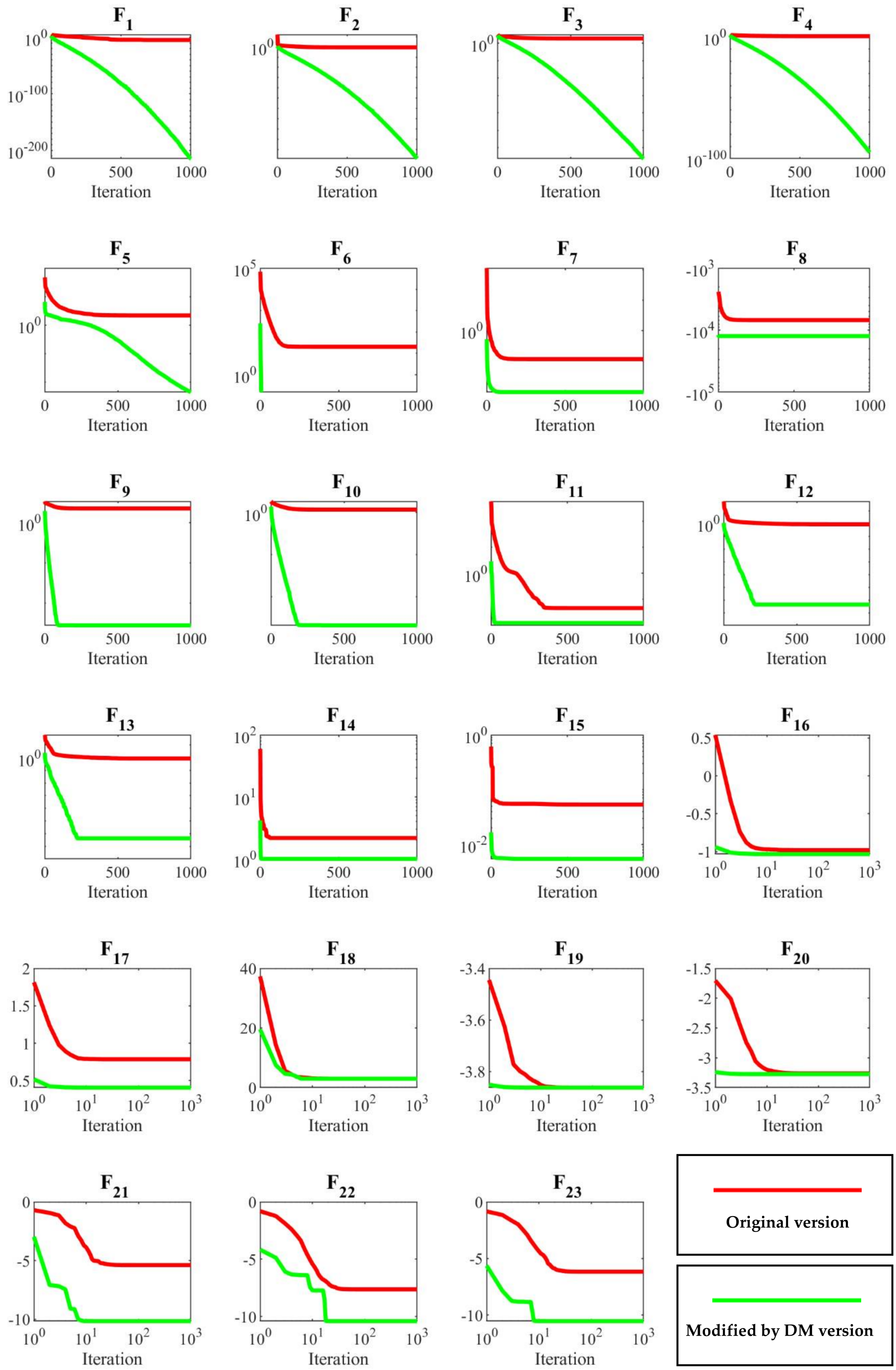

Figure 4. The convergence curves of PSO for original and "modified by DM" versions.

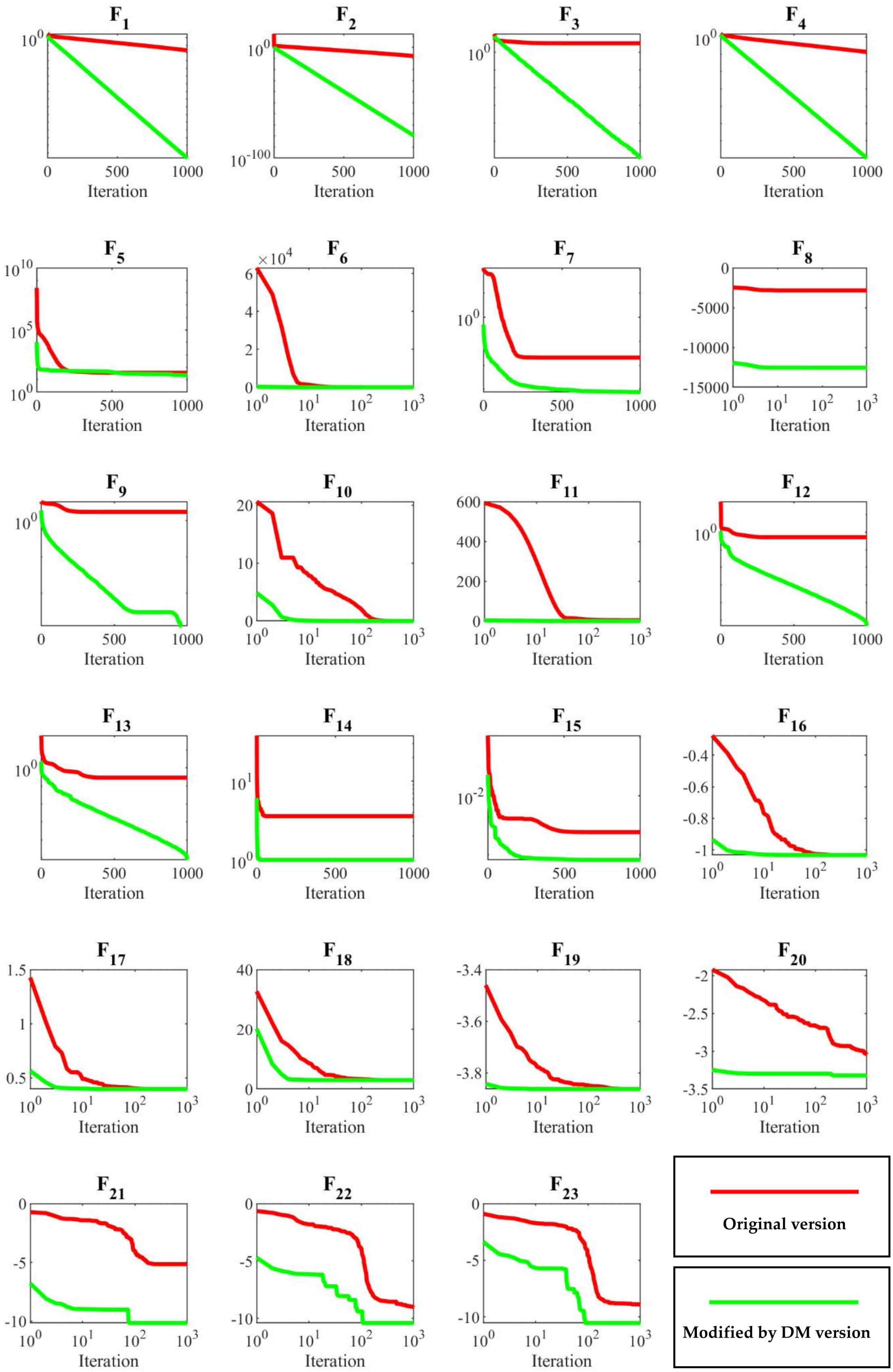

Figure 5. The convergence curves of GSA for original and "modified by DM" versions.

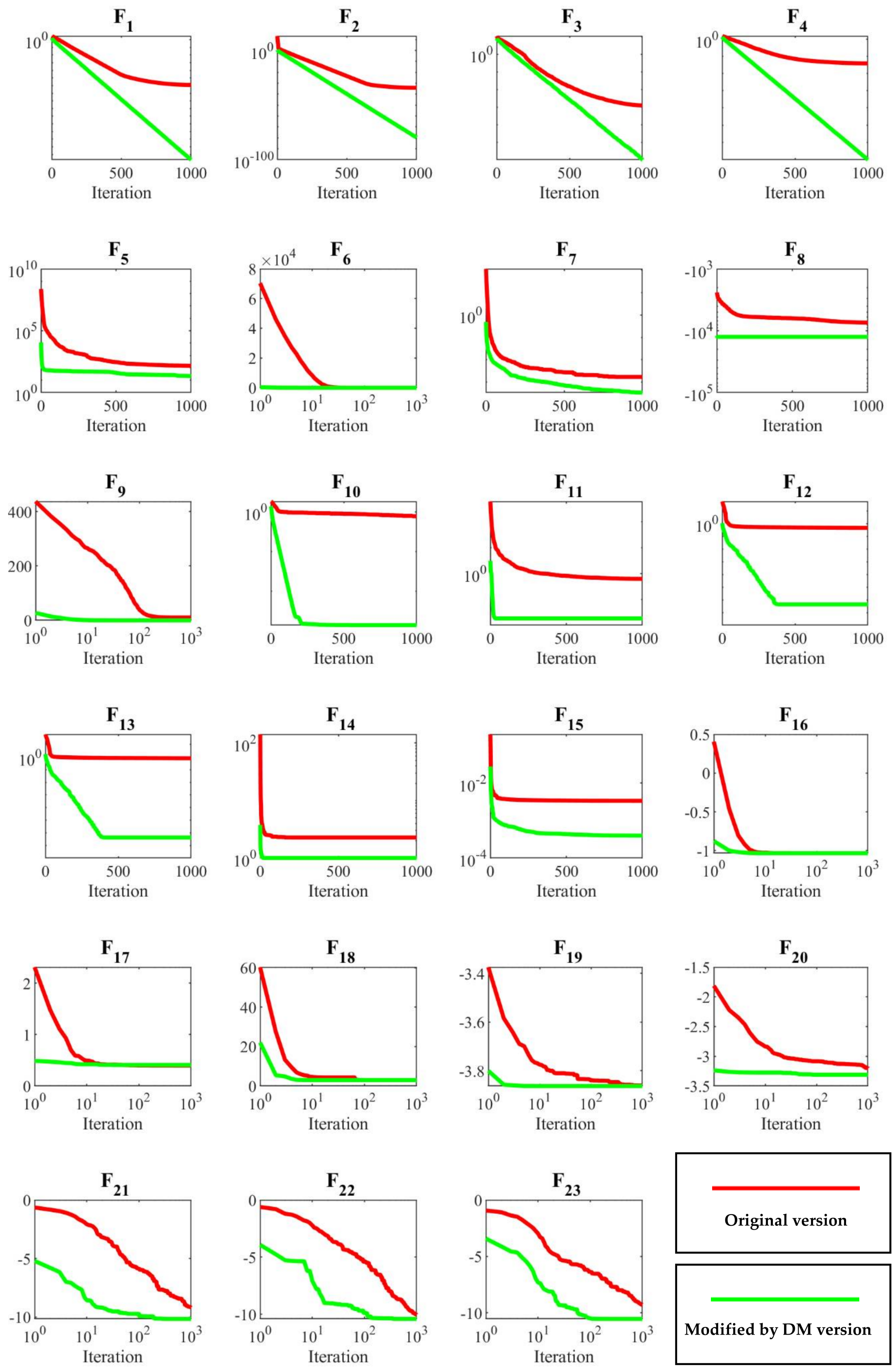

Figure 6. The convergence curves of TLBO for original and "modified by DM" versions.

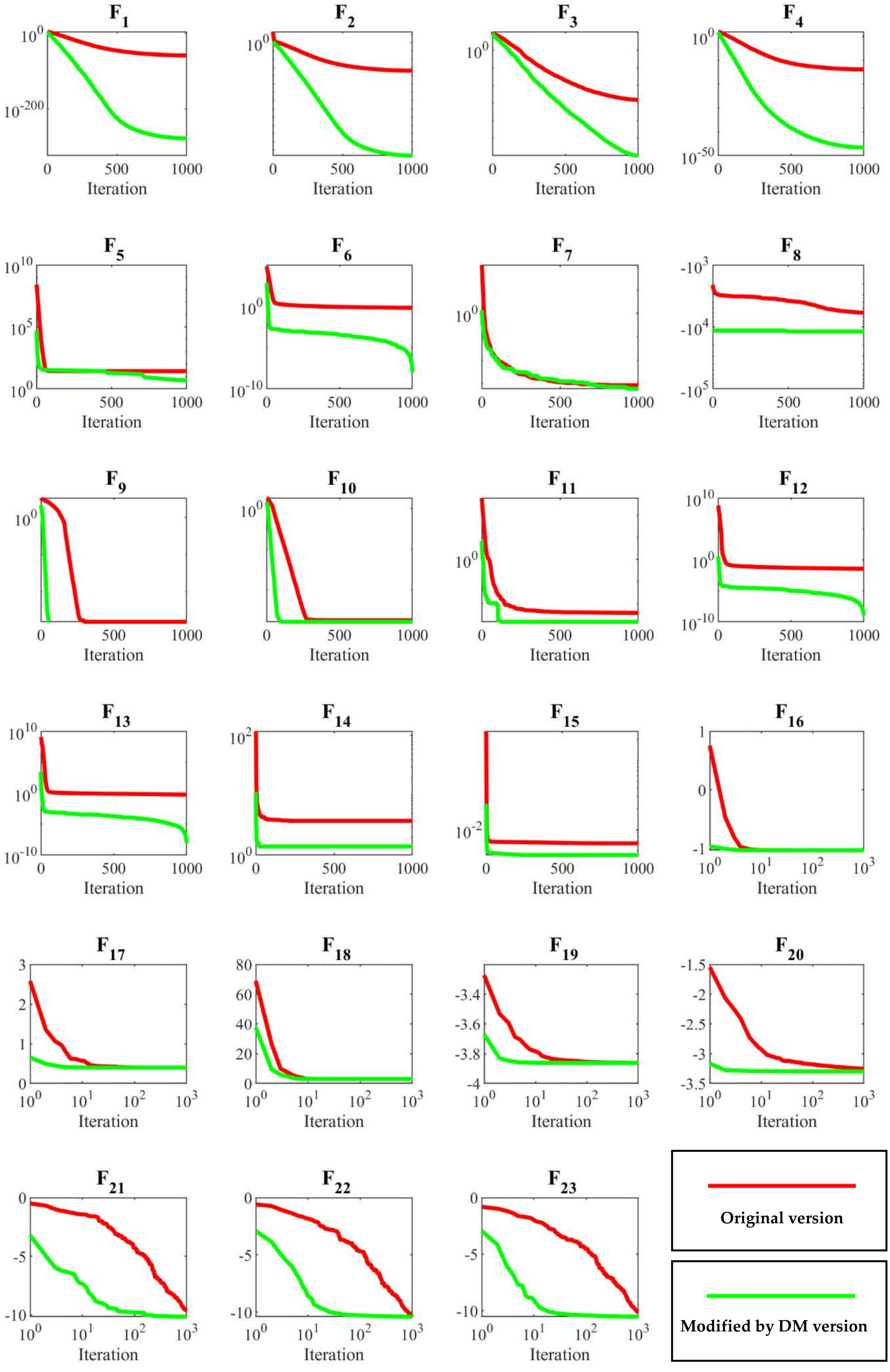

Figure 7. The convergence curves of GWO for original and "modified by DM" versions.

Table 7. Computational time analysis on unimodal objective functions (second).

		GA	MGA	PSO	MPSO	GSA	MGSA	TLBO	MTLBO	GWO	MGWO
F1	P.I.	0.0025	0.0079	0.0011	0.0039	0.0105	0.0129	0.0055	0.0063	0.0025	0.0042
	P.C.R.	2.5218	7.9613	1.1309	3.9634	10.5790	12.9160	5.5465	6.3235	2.5969	4.2516
	O.T.S.	**0.8699**		**0.2407**		**2.5412**		**2.3713**		**1.8153**	
F2	P.I.	0.0024	0.0103	0.0019	0.0050	0.0107	0.0138	0.0056	0.0057	0.0027	0.0031
	P.C.R.	2.4125	10.3794	1.1945	5.0867	10.7194	13.8475	5.6229	5.7781	2.7101	3.1541
	O.T.S.	**0.1031**		**0.2148**		**4.2400**		**2.1668**		**1.1421**	
F3	P.I.	0.0071	0.1630	0.0035	0.0784	0.0126	0091	0.0115	0.0851	0.0075	0.0918
	P.C.R.	7.1174	163.0316	3.5810	78.4169	12.5987	91.2295	11.53338	85.1981	7.5661	91.8338
	O.T.S.	**3.7623**		**3.5364**		**5.5923**		**44.5162**		**37.2164**	
F4	P.I.	0.0023	0.0082	0.0090	0.0038	0.0102	0.0133	0.0035	0.0065	0.0026	0.0075
	P.C.R.	2.3254	8.2181	0.9058	3.8641	10.2537	13.3495	3.5392	6.5216	2.6966	7.5614
	O.T.S.	**0.2793**		**0.0721**		**3.5859**		**1.5757**		**1.2614**	
F5	P.I.	0.0031	0.0300	0.0012	0.0138	0.0106	0.0218	0.0041	0.0177	0.0035	0.0015
	P.C.R.	3.1018	30.0027	1.2819	13.8921	10.6810	21.8228	4.1720	17.7407	3.5823	15.2196
	O.T.S.	**0.2677**		**0.2105**		**9.8639**		**0.2629**		**6.1547**	
F6	P.I.	0.0024	0.0114	0.0007	0.0048	0.0101	0.0146	0.0032	0.0069	0.0027	0.0114
	P.C.R.	2.4156	11.4786	0.7947	4.8365	10.1832	14.6262	3.2010	6.9231	2.7229	11.4774
	O.T.S.	**0.1558**		**0.0870**		**0.5940**		**0.8843**		**0.2376**	
F7	P.I.	0.0047	0.0785	0.0020	0.0381	0.0107	0.0493	0.0067	0.0428	0.0049	0.0390
	P.C.R.	4.7185	78.5709	2.0728	38.1957	10.7416	49.3055	6.7193	42.8670	4.9646	39.0686
	O.T.S.	**4.3080**		**0.1156**		**0.9553**		**6.5309**		**25.3156**	

Table 8. Computational time analysis on multimodal objective functions (second).

		GA	MGA	PSO	MPSO	GSA	MGSA	TLBO	MTLBO	GWO	MGWO
F8	P.I.	0.0037	0.0320	0.0015	0.0145	0.0108	0.0261	0.0049	0.0175	0.0034	0.0260
	P.C.R.	3.7004	32.0411	1.5194	14.5251	10.8230	26.1593	4.9084	17.5273	3.4282	26.0069
	O.T.S.	**0**		**0**		**0**		**0**		**0**	
F9	P.I.	0.0029	0.0128	0.0011	0.0057	0.0103	0.0160	0.0041	0.0084	0.0031	0.0154
	P.C.R.	2.9477	12.8883	1.1813	5.7547	10.3758	16.0688	4.1855	8.4233	3.1391	15.3918
	O.T.S.	**0**		**0.0404**		**0.4220**		**0.1464**		**0.3783**	
F10	P.I.	0.0028	0.0149	0.0014	0.0069	0.0105	0.0168	0.0036	0.0092	0.0031	0.0145 3.9092
	P.C.R.	2.8721	14.9436	1.4455	6.9011	10.5083	16.8567	3.6440	9.2561	3.1046	14.5384
	O.T.S.	**0.1703**		**0.0755**		**5.2755**		**0.4661**		**1.3942**	
F11	P.I.	0.0038	0.0403	0.0015	0.0196	0.0110	0.0288	0.0049	0.0212	0.0039	0.0289
	P.C.R.	3.8680	40.3218	1.5616	19.6877	11.0021	28.8871	4.9292	21.2885	3.9092	28.9343
	O.T.S.	**0.3821**		**0.6098**		**0.6825**		**0.3316**		**1.7435**	
F12	P.I.	0.0102	0.2506	0.0047	0.1188	0.0135	0.1307	0.0148	0.1294	0.0111	0.1383
	P.C.R.	10.2150	250.5924	4.7914	118.8262	13.5497	130.7448	14.8773	129.4573	11.1936	138.3059
	O.T.S.	**0.8169**		**0.1446**		**3.1468**		**1.3416**		**1.5073**	
F13	P.I.	0.0099	0.2358	0.0046	0.1201	0.0141	0.1276	0.0145	0.1311	0.0104	0.1414
	P.C.R.	9.9011	235.8891	4.6563	120.1492	14.1304	127.6587	14.5221	131.1560	10.4009	141.4475
	O.T.S.	**1.0468**		**0.3086**		**2.0328**		**0.9692**		**1.2553**	

Table 9. Computational time analysis on fixed-dimension multimodal objective functions (second).

		GA	MGA	PSO	MPSO	GSA	MGSA	TLBO	MTLBO	GWO	MGWO
F14	P.I.	0.0010	0.0472	0.0082	0.0249	0.0104	0.0264	0.0269	0.0434	0.0161	0.0328
	P.C.R.	1.0468	47.2794	8.2528	24.9162	10.4771	26.4766	26.9827	43.4908	16.1510	32.8053
	O.T.S.	**0.6287**		**0.1399**		**0.4855**		**0.2652**		**0.3096**	
F15	P.I.	0.0023	0.0034	0.0007	0.0013	0.0038	0.0047	0.0038	0.0039	0.0014	0.0029
	P.C.R.	2.3853	3.4227	0.7764	1.3148	3.8712	4.7457	3.8933	3.9249	1.4375	2.9109
	O.T.S.	**0.1309**		**0**		**0.9101**		**0.3985**		**0.1430**	
F16	P.I.	0.0020	0.0031	0.0006	0.0008	0.0035	0.0038	0.0029	0.0033	0.0011	0.0018
	P.C.R.	2.0042	3.1025	0.6528	0.8631	3.5103	3.8281	2.9861	3.3501	1.1685	1.8361
	O.T.S.	**0.2672**		**0.0442**		**0.5542**		**0.1176**		**0.1103**	
F17	P.I.	0.0019	0.0021	0.0005	0.0006	0.0033	0.0038	0.0027	0.0029	0.0010	0.0014
	P.C.R.	1.9120	2.1799	0.5358	0.5988	3.3251	3.8600	2.7069	2.9407	1.0611	1.4350
	O.T.S.	**0.1188**		**0**		**0.6116**		**0.2421**		**0.2306**	

Table 9. *Cont.*

		GA	MGA	PSO	MPSO	GSA	MGSA	TLBO	MTLBO	GWO	MGWO
F18	P.I.	0.0018	0.0025	0.0004	0.0006	0.0035	0.0037	0.0029	0.0030	0.0010	0.0015
	P.C.R.	1.8757	2.5166	0.4211	0.6029	3.4917	3.7297	2.9081	3.0689	1.0903	1.5060
	O.T.S.	**0.0979**		**0.0576**		**1.1963**		**0.1772**		**0.2609**	
F19	P.I.	0.0025	0.0037	0.0008	0.0014	0.0037	0.0048	0.0035	0.0041	0.0016	0.0026
	P.C.R.	2.5233	3.7541	0.8540	1.4255	3.7397	4.8479	3.5075	4.1393	1.6122	2.6632
	O.T.S.	**0.4878**		**0.0811**		**1.4218**		**0.0983**		**0.4146**	
F20	P.I.	0.0028	0.0051	0.0009	0.0021	0.0045	0.0056	0.0036	0.0083	0.0017	0.0042
	P.C.R.	2.8411	5.1640	0.9246	2.1016	4.5773	5.6775	3.6700	4.8358	1.7797	4.2165
	O.T.S.	**0**		**0.0846**		**0**		**0.1673**		**0.1526**	
F21	P.I.	0.0027	0.0056	0.0010	0.0023	0.0044	0.0057	0.0041	0.0054	0.0020	0.0045
	P.C.R.	2.7466	5.6009	1.0173	2.3777	4.4041	5.7909	4.1728	5.4967	2.0670	4.5858
	O.T.S.	**0.1158**		**0.0913**		**0**		**2.1696**		**0.5408**	
F22	P.I.	0.0030	0.0071	0.0010	0.0030	0.0041	0.0064	0.0048	0.0065	0.0022	0.0050
	P.C.R.	3.0287	7.1416	1.0454	3.0463	4.1996	6.4820	4.8034	6.5908	2.2729	5.0265
	O.T.S.	**0.1123**		**0.3930**		**1.2738**		**0.4395**		**0.6946**	
F23	P.I.	0.0035	0.0089	0.0012	0.0039	0.0045	0.0075	0.0051	0.0080	0.0026	0.0065
	P.C.R.	3.5326	8.9343	1.2184	3.9863	4.5094	7.5026	5.1003	8.0755	2.6291	6.5630
	O.T.S.	**2.0295**		**0.0757**		**1.2981**		**0.2980**		**0.7469**	

4.3. Discussion

Exploitation and exploration abilities are two important indicators in evaluating optimization algorithms. The exploitation ability of an optimization algorithm means its power to provide a quasi-optimal solution. An algorithm that offers a better quasi-optimal solution than another algorithm has a higher exploitation ability. The unimodal objective functions F_1 to F_7, which have only one global optimal solution without local solutions, are applied to analyze the exploitation ability of optimization algorithms. The results presented in Tables 2–6 show that the proposed DM by modifying the optimization algorithms is able to increase the exploitation ability of the optimization algorithms and as a result more suitable quasi-optimal solutions are provided by the modified version.

The exploration ability means the power of the optimization algorithm to scan the search space of an optimization problem. Given that the basis of optimization algorithms is random scanning of the search space, an algorithm that scans the search space more accurately is able move towards a quasi-optimal solution by escape from local optimal solutions. In the second and third category of objective functions F_8 to F_{23}, there are multiple local solutions besides the global optimum which are useful to analyze the local optima avoidance and an explorative ability of an algorithm. Tables 2–6 show that the modified version with the DM of optimization algorithms has a higher exploration ability than the original version.

The convergence curves shown in Figures 3–7 visually show the effect of the proposed DM on the modifying the optimization algorithms. In these figures it is clear that the modified version moves with more convergence towards the quasi-optimal solution.

The simulation results of optimization algorithms to solve the optimization problems show that the modified version of the optimization algorithms with the DM are much more competitive than the its original version. Therefore, the proposed method has the ability to be implemented on a variety of optimization algorithms in order to solve various optimization problems.

The result of computational time analysis for both original and modified by DM versions is presented in Tables 7–9. In these tables, three different time criteria are presented, which are the average time per iteration (P.I.), the average time per complete run (P.C.R.), and overall time required for the original and modified algorithm to achieve similar objective function value (O.T.S.). Due to the addition of a correction phase based on proposed DM, P.I. and P.C.R. have been increased compared to the original version. Table 7 shows that except for four cases (TLBO: F_3, GWO: F_3, F_5, and F_7), in all unimodal objective functions, the modified version provides the final solution of the original version in less time. Tables 8 and 9 show that the modified version of the studied algorithms for all F_8 to F_{23} objective functions provides the final solution of the original version in less time.

5. Conclusions

There are various optimization problems in different sciences that should be optimized using the appropriate method. The optimization algorithm is one method to solve such problems, and it can provide a quasi-optimal solution by random scanning in the search space. Many optimization algorithms have been proposed by researchers which have been applied by scientists to solve optimization problems. The performance of optimization algorithms in achieving quasi-optimal solutions is improved by modifying optimization algorithms. In this paper, a new modification method has been presented for optimization algorithms called Dehghani method (DM). The main idea of the proposed DM is to improve and strengthen the best member of the population using the information of the population members. In DM, all members of a population, even the worst one, can contribute to the development of the population. The various stages of DM have been described and then has been modeled mathematically. The DM has been implemented on five different optimization algorithms including GA, PSO, GSA, TLBO, and GWO. The effect of the proposed method on modifying the performance of optimization algorithms in solving optimization problems has been evaluated on a set of twenty-three standard objective functions. In this evaluation, the results of optimizing the objective functions set has been presented for both the original and the modified by DM version of the optimization algorithms. The results of simulation and implementation of DM on the mentioned optimization algorithms with the aim of optimizing the optimization problems show that the proposed method improves the performance of the optimization algorithms. The optimization of different objective functions in the three groups unimodal, multimodal, and fixed-dimension multimodal functions indicates that the modified version with the proposed method is much more competitive than the original version. Moreover, the convergence curves visually show that the modified version moves with more convergence towards the quasi-optimal solution.

The authors suggest several ideas and proposals for future studies and perspectives of this study to researchers. The main potential for these ideas is to be found in modifying various optimization algorithms using DM. DM may also be used to overcome many objective real-life optimizations as well as multi-objective problems.

Author Contributions: Conceptualization, M.D., Z.M., A.D., H.S., O.P.M. and J.M.G.; methodology, M.D., Z.M. and A.D.; software, M.D., Z.M. and H.S.; validation, J.M.G., G.D., A.D., H.S., O.P.M., R.A.R.-M., C.S., D.S. and A.E.; formal analysis, A.D., A.E.; investigation, M.D. and Z.M.; resources, A.D., A.E. and J.M.G.; data curation, G.D.; writing—original draft preparation, M.D. and Z.M.; writing—review and editing, A.D., R.A.R.-M., D.S., C.S., A.E., O.P.M., G.D., and J.M.G.; visualization, M.D.; supervision, M.D., Z.M. and A.D.; project administration, A.D. and Z.M.; funding acquisition, C.S., D.S. and R.A.R.-M. All authors have read and agreed to the published version of the manuscript.

Funding: The current project was funded by Tecnológico de Monterrey and FEMSA Foundation (grant: CAMPUSCITY Project).

Conflicts of Interest: The authors declare no conflict of interest. The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Abbreviations

Acronym	Definition
ABC	Artificial Bee Colony
ACO	Ant Colony Optimization
ACROA	Artificial Chemical Reaction Optimization Algorithm
BA	Bat-inspired Algorithm
BBO	Biogeography-Based Optimizer
BH	Black Hole
BSS	Best-Suggested Solution
CS	Cuckoo Search
CSO	Curved Space Optimization
DM	Dehghani Method
DGO	Dice Game Optimizer
DGO	Darts Game Optimizer
DPO	Doctor and Patient Optimization

DE	Differential Evolution
DTO	Donkey Theorem Optimization
ES	Evolution Strategy
EPO	Emperor Penguin Optimizer
FOA	Following Optimization Algorithm
FGBO	Football Game Based Optimization
GA	Genetic Algorithm
GP	Genetic Programming
GO	Group Optimization
GOA	Grasshopper Optimization Algorithm
GSA	Gravitational Search Algorithm
GbSA	Galaxy-based Search Algorithm
GWO	Grey Wolf Optimizer
HOGO	Hide Objects Game Optimization
MLO	Multi Leader Optimizer
OSA	Orientation Search Algorithm
PSO	Particle Swarm Optimization
RSO	Rat Swarm Optimizer
RO	Ray Optimization
SHO	Spotted Hyena Optimizer
SGO	Shell Game Optimization
SWOA	Small World Optimization Algorithm
SS	Suggested Solution
TLBO	Teaching-Learning-Based Optimization
OF	Objective Function

Appendix A

Table A1. Unimodal objective functions.

$F_1(x) = \sum_{i=1}^{m} x_i^2$	$[-100, 100]^m$
$F_2(x) = \sum_{i=1}^{m} \lvert x_i \rvert + \prod_{i=1}^{m} \lvert x_i \rvert$	$[-10, 10]^m$
$F_3(x) = \sum_{i=1}^{m} \left(\sum_{j=1}^{i} x_i \right)^2$	$[-100, 100]^m$
$F_4(x) = \max\{ \lvert x_i \rvert ,\ 1 \le i \le m \}$	$[-100, 100]^m$
$F_5(x) = \sum_{i=1}^{m-1} \left[100 \left(x_{i+1} - x_i^2 \right)^2 + (x_i - 1)^2) \right]$	$[-30, 30]^m$
$F_6(x) = \sum_{i=1}^{m} ([x_i + 0.5])^2$	$[-100, 100]^m$
$F_7(x) = \sum_{i=1}^{m} i x_i^4 + random(0, 1)$	$[-1.28, 1.28]^m$

Table A2. Multimodal objective functions.

$F_8(x) = \sum_{i=1}^{m} -x_i \ \sin\left(\sqrt{\lvert x_i \rvert} \right)$	$[-500, 500]^m$
$F_9(x) = \sum_{i=1}^{m} \left[x_i^2 - 10 \cos(2\pi x_i) + 10 \right]$	$[-5.12, 5.12]^m$
$F_{10}(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{m} \sum_{i=1}^{m} x_i^2} \right) - \exp\left(\frac{1}{m} \sum_{i=1}^{m} cos(2\pi x_i) \right) + 20 + e$	$[-3.2, 3.2]^m$
$F_{11}(x) = \frac{1}{4000} \sum_{i=1}^{m} x_i^2 - \prod_{i=1}^{m} cos\left(\frac{x_i}{\sqrt{i}} \right) + 1$	$[-600, 600]^m$
$F_{12}(x) = \frac{\pi}{m} \left\{ 10 \sin(\pi y_1) + \sum_{i=1}^{m} (y_i - 1)^2 \left[1 + 10 \sin^2(\pi y_{i+1}) \right] + (y_n - 1)^2 \right\} + \sum_{i=1}^{m} u(x_i, 10, 100, 4)$ $u(x_i, a, i, n) = \begin{cases} k(x_i - a)^n & x_i > -a \\ 0 & -a < x_i < a \\ k(-x_i - a)^n & x_i < -a \end{cases}$	$[-50, 50]^m$
$F_{13}(x) = 0.1 \left\{ \sin^2(3\pi x_1) + \sum_{i=1}^{m} (x_i - 1)^2 \left[1 + \sin^2(3\pi x_i + 1) \right] + (x_n - 1)^2 \left[1 + \sin^2(2\pi x_m) \right] \right\} + \sum_{i=1}^{m} u(x_i, 5, 100, 4)$	$[-50, 50]^m$

Table A3. Multimodal objective functions with fixed dimension.

$F_{14}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j+\sum_{i=1}^{2}(x_i-a_{ij})^6}\right)^{-1}$	$[-65.53, 65.53]^2$
$F_{15}(x) = \sum_{i=1}^{11}\left[a_i - \frac{x_1(b_i^2+b_ix_2)}{b_i^2+b_ix_3+x_4}\right]^2$	$[-5, 5]^4$
$F_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	$[-5, 5]^2$
$F_{17}(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)cosx_1 + 10$	$[-5, 10] \times [0, 15]$
$F_{18}(x) = \left[1 + (x_1 + x_2 + 1)^2\left(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2\right)\right] \times\left[30 + (2x_1 - 3x_2)^2 \times \left(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2\right)\right]$	$[-5, 5]^2$
$F_{19}(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{3} a_{ij}\left(x_j - P_{ij}\right)^2\right)$	$[0, 1]^3$
$F_{20}(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{6} a_{ij}\left(x_j - P_{ij}\right)^2\right)$	$[0, 1]^6$
$F_{21}(x) = -\sum_{i=1}^{5}\left[(X - a_i)(X - a_i)^T + 6c_i\right]^{-1}$	$[0, 10]^4$
$F_{22}(x) = -\sum_{i=1}^{7}\left[(X - a_i)(X - a_i)^T + 6c_i\right]^{-1}$	$[0, 10]^4$
$F_{23}(x) = -\sum_{i=1}^{10}\left[(X - a_i)(X - a_i)^T + 6c_i\right]^{-1}$	$[0, 10]^4$

References

1. Dehghani, M.; Montazeri, Z.; Dehghani, A.; Ramirez-Mendoza, R.A.; Samet, H.; Guerrero, J.M.; Dhiman, G. MLO: Multi leader optimizer. *Int. J. Intell. Eng. Syst.* **2020**, *13*, 364–373.
2. Dhiman, G.; Garg, M.; Nagar, A.; Kumar, V.; Dehghani, M. A novel algorithm for global optimization: Rat swarm optimizer. *J. Ambient Intell. Humaniz. Comput.* **2020**. [CrossRef]
3. Dehghani, M.; Mardaneh, M.; Guerrero, J.M.; Malik, O.P.; Kumar, V. Football game based optimization: An application to solve energy commitment problem. *Int. J. Intell. Eng. Syst.* **2020**, *13*, 514–523. [CrossRef]
4. Dehghani, M.; Montazeri, Z.; Malik, O.P. Energy commitment: A planning of energy carrier based on energy consumption. *Электротехника Электромеханика* **2019**, *4*, 69–72. [CrossRef]
5. Liu, J.; Dong, H.; Jin, T.; Liu, L.; Manouchehrinia, B.; Dong, Z. Optimization of hybrid energy storage systems for vehicles with dynamic on-off power loads using a nested formulation. *Energies* **2018**, *11*, 2699. [CrossRef]
6. Carpinelli, G.; Mottola, F.; Proto, D.; Russo, A.; Varilone, P. A hybrid method for optimal siting and sizing of battery energy storage systems in unbalanced low voltage microgrids. *Appl. Sci.* **2018**, *8*, 455. [CrossRef]
7. Ehsanifar, A.; Dehghani, M.; Allahbakhshi, M. Calculating the leakage inductance for transformer inter-turn fault detection using finite element method. In Proceedings of the 2017 Iranian Conference on Electrical Engineering (ICEE), Tehran, Iran, 2–4 May 2017; pp. 1372–1377.
8. Dehghani, M.; Montazeri, Z.; Malik, O.P. Optimal sizing and placement of capacitor banks and distributed generation in distribution systems using spring search algorithm. *Int. J. Emerg. Electr. Power Syst.* **2020**, *21*, 20190217. [CrossRef]
9. Dehghani, M.; Montazeri, Z.; Malik, O.P.; Al-Haddad, K.; Guerrero, J.M.; Dhiman, G. A new methodology called dice game optimizer for capacitor placement in distribution systems. *Электротехника Электромеханика* **2020**. [CrossRef]
10. Dehbozorgi, S.; Ehsanifar, A.; Montazeri, Z.; Dehghani, M.; Seifi, A. Line loss reduction and voltage profile improvement in radial distribution networks using battery energy storage system. In Proceedings of the IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI), Tehran, Iran, 22 December 2017; pp. 215–219.
11. Montazeri, Z.; Niknam, T. Optimal utilization of electrical energy from power plants based on final energy consumption using gravitational search algorithm. *Электротехника Электромеханика*, **2018**, *4*, 70–73. [CrossRef]

12. Dehghani, M.; Mardaneh, M.; Montazeri, Z.; Ehsanifar, A.; Ebadi, M.; Grechko, O. Spring search algorithm for simultaneous placement of distributed generation and capacitors. *Електротехніка Електромеханіка* **2018**, *6*, 68–73. [CrossRef]
13. Yu, J.; Kim, C.-H.; Wadood, A.; Khurshiad, T.; Rhee, S.-B. A novel multi-population based chaotic JAYA algorithm with application in solving economic load dispatch problems. *Energies* **2018**, *11*, 1946. [CrossRef]
14. Sleesongsom, S.; Bureerat, S. Topology optimisation using MPBILs and multi-grid ground element. *Appl. Sci.* **2018**, *8*, 271. [CrossRef]
15. Dehghani, M.; Montazeri, Z.; Ehsanifar, A.; Seifi, A.; Ebadi, M.; Grechko, O. Planning of energy carriers based on final energy consumption using dynamic programming and particle swarm optimization. *Электротехника Электромеханика* **2018**, *5*, 62–71. [CrossRef]
16. Montazeri, Z.; Niknam, T. Energy carriers management based on energy consumption. In Proceedings of the IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI), Tehran, Iran, 22 December 2017; pp. 539–543.
17. Dehghani, M.; Mardaneh, M.; Malik, O.P.; Guerrero, J.M.; RMorales-Menendez, R.; Ramirez-Mendoza, A.; Matas, J.; Abusorrah, A. Energy commitment for a power system supplied by a multiple energy carriers system using following optimization algorithm. *Appl. Sci.* **2020**, *10*, 5862. [CrossRef]
18. Kennedy, J.; Eberhart, R. Particle swarm optimization. In *Proceeding of the IEEE International Conference on Neural Networks, Perth, WA, Australia*; IEEE Service Center: Piscataway, NJ, USA, 1942; Volume 1948.
19. Dorigo, M.; Stützle, T. Ant colony optimization: Overview and recent advances. In *Handbook of Metaheuristics*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 311–351.
20. Ning, J.; Zhang, C.; Sun, P.; Feng, Y. Comparative study of ant colony algorithms for multi-objective optimization. *Information* **2019**, *10*, 11. [CrossRef]
21. Dhiman, G.; Kumar, V. Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications. *Adv. Eng. Softw.* **2017**, *114*, 48–70. [CrossRef]
22. Dehghani, M.; Montazeri, Z.; Dehghani, A.; Malik, O.P. GO: Group optimization. *Gazi Univ. J. Sci.* **2020**, *33*, 381–392. [CrossRef]
23. Karaboga, D.; Basturk, B. Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems. In *International Fuzzy Systems Association World Congress*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 789–798.
24. Dehghani, M.; Mardaneh, M.; Malik, O. FOA: 'Following' optimization algorithm for solving power engineering optimization problems. *J. Oper. Autom. Power Eng.* **2020**, *8*, 57–64.
25. Yang, X.-S. A new metaheuristic bat-inspired algorithm. In *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 65–74.
26. Dhiman, G.; Kumar, V. Emperor penguin optimizer: A bio-inspired algorithm for engineering problems. *Knowl. Based Syst.* **2018**, *159*, 20–50. [CrossRef]
27. Gandomi, A.H.; Yang, X.-S.; Alavi, A.H. Cuckoo search algorithm: A metaheuristic approach to solve structural optimization problems. *Eng. Comput.* **2013**, *29*, 17–35. [CrossRef]
28. Dehghani, M.; Mardaneh, M.; Malik, O.P.; NouraeiPour, S.M. DTO: Donkey Theorem Optimization. In Proceedings of the 27th Iranian Conference on Electrical Engineering (ICEE), Yazd, Iran, 30 April–2 May 2019; pp. 1855–1859.
29. Sarzaeim, P.; Bozorg-Haddad, O.; Chu, X. Teaching-learning-based optimization (TLBO) algorithm. In *Advanced Optimization by Nature-Inspired Algorithms*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 51–58.
30. Saremi, S.; Mirjalili, S.; Lewis, A. Grasshopper optimisation algorithm: Theory and application. *Adv. Eng. Softw.* **2017**, *105*, 30–47. [CrossRef]
31. Dehghani, M.; Mardaneh, M.; Guerrero, J.M.; Malik, O.P.; Ramirez-Mendoza, R.A.; Matas, J.; Vasquez, J.C.; Parra-Arroyo, L. A new "Doctor and Patient" optimization algorithm: An application to energy commitment problem. *Appl. Sci.* **2020**, *10*, 5791. [CrossRef]
32. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
33. Dehghani, M.; Montazeri, Z.; Saremi, S.; Dehghani, A.; Malik, O.P.; Al-Haddad, K.; Guerrero, J.M. HOGO: Hide objects game optimization. *Int. J. Intell. Eng. Syst.* **2020**, *13*, 216–225.
34. Dehghani, M.; Montazeri, Z.; Malik, O.P.; Ehsanifar, A.; Dehghani, A. OSA: Orientation search algorithm. *Int. J. Ind. Electr. Control Optim.* **2019**, *2*, 99–112.

35. Dehghani, M.; Montazeri, Z.; Malik, O.P.; Dhiman, G.; Kumar, V. BOSA: Binary orientation search algorithm. *Int. J. Innov. Technol. Explor. Eng.* **2019**, *9*, 5306–5310.
36. Dehghani, M.; Montazeri, Z.; Malik, O.P. DGO: Dice game optimizer. *Gazi Univ. J. Sci.* **2019**, *32*, 871–882. [CrossRef]
37. Dehghani, M.; Montazeri, Z.; Malik, O.P.; Givi, H.; Guerrero, J.M. Shell game optimization: A novel game-based algorithm. *Int. J. Intell. Eng. Syst.* **2020**, *13*, 246–255. [CrossRef]
38. Dehghani, M.; Montazeri, Z.; Givi, H.; Guerrero, J.M.; Dhiman, G. Darts game optimizer: A new optimization technique based on darts game. *Int. J. Intell. Eng. Syst.* **2020**, *13*, 286–294. [CrossRef]
39. Dehghani, M.; Montazeri, Z.; Dehghani, A.; Seifi, A. Spring search algorithm: A new meta-heuristic optimization algorithm inspired by Hooke's law. In Proceedings of the 2017 IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI), Tehran, Iran, 22 December 2017; pp. 210–214.
40. Dehghani, M.; Montazeri, Z.; Dehghani, A.; Nouri, N.; Seifi, A. BSSA: Binary spring search algorithm. In Proceedings of the 2017 IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI), Tehran, Iran, 22 December 2017; pp. 220–224.
41. Moghaddam, F.F.; Moghaddam, R.F.; Cheriet, M. Curved space optimization: A random search based on general relativity theory. *arXiv* **2012**, arXiv:1208.2214.
42. Hatamlou, A. Black hole: A new heuristic optimization approach for data clustering. *Inf. Sci.* **2013**, *222*, 175–184. [CrossRef]
43. Kaveh, A.; Khayatazad, M. A new meta-heuristic method: Ray optimization. *Comput. Struct.* **2012**, *112*, 283–294. [CrossRef]
44. Alatas, B. ACROA: Artificial chemical reaction optimization algorithm for global optimization. *Expert Syst. Appl.* **2011**, *38*, 13170–13180. [CrossRef]
45. Shah-Hosseini, H. Principal components analysis by the galaxy-based search algorithm: A novel metaheuristic for continuous optimization. *Int. J. Comput. Sci. Eng.* **2011**, *6*, 132–140.
46. Du, H.; Wu, X.; Zhuang, J. Small-world optimization algorithm for function optimization. In *International Conference on Natural Computation*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 264–273. [CrossRef]
47. Karkalos, N.E.; Markopoulos, A.P.; Davim, J.P. Evolutionary-based methods. In *Computational Methods for Application in Industry 4.0*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 11–31. [CrossRef]
48. Mirjalili, S. Biogeography-based optimisation. In *Evolutionary Algorithms and Neural Networks*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 57–72. [CrossRef]
49. Storn, R.; Price, K. Differential evolution-A simple and efficient adaptive scheme for global optimization over continuous spaces. *Berkeley ICSI* **1995**, *11*, 341–359. [CrossRef]
50. Tang, K.-S.; Man, K.-F.; Kwong, S.; He, Q. Genetic algorithms and their applications. *IEEE Signal Process. Mag.* **1996**, *13*, 22–37. [CrossRef]
51. Beyer, H.-G.; Schwefel, H.-P. Evolution strategies—A comprehensive introduction. *Nat. Comput.* **2002**, *1*, 3–52. [CrossRef]
52. Koza, J.R. Genetic Programming: A Paradigm for Genetically Breeding Populations of Computer Programs to Solve Problems. PhD Thesis, Stanford University, Stanford, CA, USA, 1990.
53. Rashedi, E.; Nezamabadi-Pour, H.; Saryazdi, S. GSA: A gravitational search algorithm. *Inf. Sci.* **2009**, *179*, 2232–2248. [CrossRef]
54. Dhghani, M.; Montazeri, Z.; Dhiman, G.; Malik, O.P.; Morales-Menendez, R.; Ramirez-Mendoza, R.A.; Dehghani, A.; Guerrero, J.M.; Parra-Arroyo, L. A spring search algorithm applied to engineering optimization problems. *Appl. Sci.* **2020**, *10*, 6173. [CrossRef]

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.