Research Article GU J Sci 33(2): 381-392 (2020) DOI:10.35378/gujs.567472

Gazi University
Journal of Science
http://dergipark.gov.tr/gujs

GO: Group Optimization

Mohammad DEHGHANI[1,*], Zeinab MONTAZERI[1], Ali DEHGHANI[2], Om Parkash MALIK[3]

[1]*Shiraz University of Technology, Department of Electrical and Electronics Engineering, Shiraz, Iran*
[2]*Islamic Azad Universities of Estahban, Department of Civil Engineering, Estahban, Iran*
[3]*University of Calgary, Department of Electrical Engineering, Calgary Alberta, Canada*

Highlights

- The paper focuse on proposing a new optimization algorithm called Group Optimization.
- GO can be used to solve optimization constrained and unconstrained problems in different sciences.
- The proposed GO is tested on 23 standard benchmark test functions.
- The performance of GO is also examined on one engineering design problem.
- The results show the merits of the GO as compared to the existing algorithms.

Article Info

Received: 19/05/2019
Accepted: 04/11/2019

Keywords

Heuristic algorithms
Optimization
Group
Group optimization
Swarm-based algorithm

Abstract

This article introduces a modern optimization algorithm to solve optimization problems. Group Optimization (GO) is based on concept that uses all agents to update population of algorithm. Every agent of population could to be used for population updating. For these purpose two groups is specified for any agent. One group for good agents and another group for bad agents. These groups is used for updating position of each agent. twenty-three standard benchmark test functions are evaluated using GO and then results are compared with eight other optimization method.

1. INTRODUCTION

1.1. Motivation

In optimizing each problem, it consists of three parts: constraints, objective functions, and decision variables [1]. In optimization, the goal is to achieve the most appropriate solution among the possible answers, so that first, to observe the constraint of the problem and then to optimize the objective function. [2]. Various optimization algorithms were proposed in recent decades to address various problems of optimization [3-6].

1.2. Literature Survey

Optimization algorithms are used in different science such as: power engineering [7, 8], civil engineering [9], energy planning [10, 11], energy commitment [12], data mining [13], bioinformatics [14], protection [15], and transmission network [16]. Physics related, Evolutionary related, and Swarm related methods are categories of optimization algorithms approaches.

Algorithms based on physics are kinds of algorithms for optimization that simulate physical laws [17]. the one class of the kind that imitates the process of gradual heating and refining of metals is Simulated

*Corresponding author, e-mail:adanbax@gmail.com

Annealing (SA) [18]. Hooke's law is used to invent and design an optimization algorithm as Spring Search Algorithm (SSA) [3, 4]. Based on the relations and equations of the law of gravitational force Gravitation Search Algorithm (GSA) is proposed [19]. Small World Optimization Algorithm (SWOA) Designed by small-world phenomenon process [20], Curved Space Optimization (CSO) based on principles of general relativity theory [21], Artificial Chemical Reaction Optimization Algorithm (ACROA) [22], Charged System Search (CSS) [23], Ray Optimization (RO) algorithm [24], Galaxy-based Search Algorithm (GbSA) [25], Black Hole (BH) [26], and Magnetic Optimization Algorithm (MOA) [27] are several other algorithms focused on physics.

An alternative type of optimization algorithms is evolutionary algorithms that simulate the birth cycle [28]. Algorithms that are included in this group are: Differential Evolution (DE) [29], Genetic Algorithm (GA) [30], Biogeography-based Optimizer (BBO) [31], Evolution Strategy (ES) [32], and Genetic Programming (GP) [33].

One another group of optimization techniques is Swarm-based algorithms, which are Inspired by normal plant cycles, insects activities and animals' social behavior. [34]. For example, an idea of group motion of birds has been used in the design of Particle Swarm Optimization (PSO) [35]. The process of moving ants in achieving the shortest path has been the idea of introducing the Ant Colony Optimization (ACO) [36]. Various other ideas have been used by scientists in the design of these kind of algorithms, such as: Bat-inspired Algorithm (BA) [37], Spotted Hyena Optimizer (SHO) [38], Bat Algorithm (BA) [39], Cuckoo Search (CS) [40], Artificial Bee Colony (ABC) [41], Emperor Penguin Optimizer (EPO) [42], Dragonfly Algorithm (DA) [43], , Donkey Theorem Optimization (DTO) [44], Grasshopper Optimization Algorithm (GOA) [45], ‘Following’ Optimization Algorithm (FOA) [46], and Grey Wolf Optimizer (GWO) [47].

1.3. Paper Contribution

Current research, introduces a modern optimization method named Group Optimization (GO) to address optimization problems in different sciences. In GO every agent could to be influenced for population updating. In this regards two groups is introduced for each agent. One group for good agents called good group and another group for bad agents that called bad group. These groups is used for updating position of each agent.

1.4. Paper Organization

The other sections of this article is arranged as follows that first in Section 2 introduces Group Optimization (GO). Section 3 discusses the study findings and discussion. In the final section, i.e. Section 4 some of the conclusions are expressed.

2. GROUP OPTIMIZATION (GO)

In this section, the simulation and mathematically modeling of GO is presented.
First, initial population of GO is defined in Equation (1):

$$X_i = \left(x_i^1, \dots, x_i^d, \dots, x_i^n\right). \tag{1}$$

Here, x_i^d is the position ‘d’ of agent ‘i’ and n is the number of variables.
The position of best and worst agent are specified in Equations (2) and (3).

$$X_{best} = location\ of\ \min(fit), \tag{2}$$

$$X_{worst} = location\ of\ \max(fit). \tag{3}$$

Here, X_{best} is the position of best agent, X_{worst} is the position of the worst agent and fit is the fitness function.

In GO for each agent, good group and bad group are calculated by Equations (4) and (5)

$$[GG_i]_{N_g \times n} \;\&\; fit(agent \in GG_i) < fit(i), \tag{4}$$

$$[BG_i]_{N_b \times n} \& fit(agent \in BG_i) > fit(i). \tag{5}$$

Here, GG_i is the good group of agent 'i' that includes better agents than the i'th agent, BG_i is the bad group of agent 'i' that includes worse agents than the i'th agent, N_g is the number of agent of good group and N_b is the number of agent of bad group.

Now, in this stage mean of above groups of each agent is calculated in Equations (6) and (7)

$$MGG_i = \begin{cases} X_i & if\ size(GG_i) = 0\ \ or\ \ fit\big(mean(GG_i)\big) > fit(i), \\ mean(GG_i) & else \end{cases} \tag{6}$$

$$MBG_i = \begin{cases} X_i & if\ size(BG_i) = 0\ \ or\ fit\big(mean(BG_i)\big) < fit(i) \\ mean(BG_i) & else \end{cases}. \tag{7}$$

Here, MGG_i is the mean of good group and MBG_i is the mean of bad group.
Finally, the new position of i'th agent is updated by Equations (8) and (9)

$$X'_i = X_i + r_1(X_{best} - X_i) + r_2(MGG_i - X_i) - r_3(X_{worst} - X_i) - r_4(MBG_i - X_i), \tag{8}$$

$$X_i = \begin{cases} X'_i & if\ fit(X'_i) \le fit(i) \\ X_i & else \end{cases}. \tag{9}$$

Here, r_i is the random number in $[0 - 1]$. The flowchart of GO shown in Figure 1.

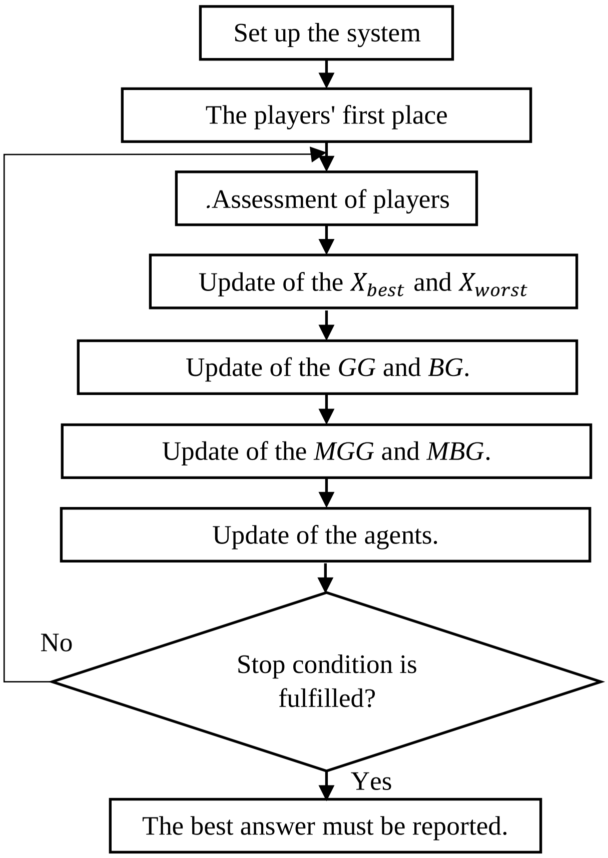

Figure 1. *Flowchart of GO*

3. SIMULATION

GO efficiency is compared to some other algorithms in this section. The algorithms examined in this comparison include: Genetic Algorithm (GA) [30], Particle Swarm Optimization (PSO) [48], Gravitational Search Algorithm (GSA) [19], Teaching–Learning-Based Optimization (TLBO) [49], Grey Wolf Optimizer (GWO) [47], Grasshopper Optimization Algorithm (GOA) [45], Spotted Hyena Optimizer (SHO) [38] and Emperor Penguin Optimizer (EPO) [42].

3.1. Benchmark Test Functions

GO efficiency is measured by twenty-three benchmark standard test functions. These set of functions are classified into three major classes: Unimodal [50], Multimodal [51], and Fixed-dimension Multimodal [51]. Tables 1-3 lists these 23 benchmark test functions.

Table 1. *Unimodal test functions*

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

Table 2. *Multimodal test functions*

$F_8(x)=\sum_{i=1}^{m} -x_i\sin(\sqrt{\lvert x_i\rvert})$	$[-500,500]^m$
$F_9(x)=\sum_{i=1}^{m}[x_i^2-10\cos(2\pi x_i)+10]$	$[-5.12,5.12]^m$
$F_{10}(x)=-20\exp\left(-0.2\sqrt{\frac{1}{m}\sum_{i=1}^{m} x_i^2}\right)-\exp\left(\frac{1}{m}\sum_{i=1}^{m} cos(2\pi x_i)\right)+20+e$	$[-32,32]^m$
$F_{11}(x)=\frac{1}{4000}\sum_{i=1}^{m} x_i^2-\prod_{i=1}^{m} cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	$[-600,600]^m$
$F_{12}(x)=\frac{\pi}{m}\left\{10\sin(\pi y_1)+\sum_{i=1}^{m}(y_i-1)^2[1+10\sin^2(\pi y_{i+1})]+(y_n-1)^2\right\}+\sum_{i=1}^{m} u(x_i,10,100,4)$ $u(x_i,a,i,n)=\begin{cases}k(x_i-a)^n & x_i>-a\\ 0 & -a<x_i<a\\ k(-x_i-a)^n & x_i<-a\end{cases}$	$[-50,50]^m$
$F_{13}(x)=0.1\left\{\sin^2(3\pi x_1)+\sum_{i=1}^{m}(x_i-1)^2[1+\sin^2(3\pi x_i+1)]+(x_n-1)^2[1+\sin^2(2\pi x_m)]\right\}+\sum_{i=1}^{m} u(x_i,5,100,4)$	$[-50,50]^m$

Table 3. *Multimodal test functions with fixed dimension*

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

3.2. Evaluation of Unimodal test function with high dimensions

The F_1-F_7 are unimodal test functions used to determine the optimization algorithm's operational capability. Table 4 displays the effects of optimization for these functions. GO performs better than other algorithms in all these functions.

Table 4. *Results for GO and other algorithms in Unimodal test functions*

		GA	PSO	GSA	TLBO	GOA	GWO	SHO	EPO	GO
F_1	Ave	1.95×10^{-12}	4.98×10^{-9}	1.16×10^{-16}	3.55×10^{-2}	2.81×10^{-1}	7.86×10^{-10}	4.61×10^{-23}	5.71×10^{-28}	5.32×10^{-36}
	std	2.01×10^{-11}	1.40×10^{-8}	6.10×10^{-17}	1.06×10^{-1}	1.11×10^{-1}	8.11×10^{-9}	7.37×10^{-23}	8.31×10^{-29}	8.24×10^{-37}
F_2	Ave	6.53×10^{-18}	7.29×10^{-4}	1.70×10^{-1}	3.23×10^{-5}	3.96×10^{-1}	5.99×10^{-20}	1.20×10^{-34}	6.20×10^{-40}	6.25×10^{-49}
	std	5.10×10^{-17}	1.84×10^{-3}	9.29×10^{-1}	8.57×10^{-5}	1.41×10^{-1}	1.11×10^{-17}	1.30×10^{-34}	3.32×10^{-40}	2.35×10^{-46}
F_3	Ave	7.70×10^{-10}	$1.40\times10^{+1}$	$4.16\times10^{+2}$	$4.91\times10^{+3}$	$4.31\times10^{+1}$	9.19×10^{-5}	1.00×10^{-14}	2.05×10^{-19}	7.12×10^{-26}
	std	7.36×10^{-9}	7.13	$1.56\times10^{+2}$	$3.89\times10^{+3}$	8.97	6.16×10^{-4}	4.10×10^{-14}	9.17×10^{-20}	5.61×10^{-29}
F_4	Ave	$9.17\times10^{+1}$	6.00×10^{-1}	1.12	$1.87\times10^{+1}$	8.80×10^{-1}	8.73×10^{-1}	2.02×10^{-14}	4.32×10^{-18}	2.14×10^{-27}
	std	$5.67\times10^{+1}$	1.72×10^{-1}	9.89×10^{-1}	8.21	2.50×10^{-1}	1.19×10^{-1}	2.43×10^{-14}	3.98×10^{-19}	6.85×10^{-30}
F_5	Ave	$5.57\times10^{+2}$	$4.93\times10^{+1}$	$3.85\times10^{+1}$	$7.37\times10^{+2}$	$1.18\times10^{+2}$	$8.91\times10^{+2}$	$2.79\times10^{+1}$	5.07	4.32×10^{-1}
	std	$4.16\times10^{+1}$	$3.89\times10^{+1}$	$3.47\times10^{+1}$	$1.98\times10^{+3}$	$1.43\times10^{+2}$	$2.97\times10^{+2}$	1.84	4.90×10^{-1}	4.85×10^{-2}
F_6	Ave	3.15×10^{-1}	9.23×10^{-9}	1.08×10^{-16}	4.88	3.15×10^{-1}	8.18×10^{-17}	6.58×10^{-1}	7.01×10^{-19}	3.15×10^{-26}
	std	9.98×10^{-2}	1.78×10^{-8}	4.00×10^{-17}	9.75×10^{-1}	9.98×10^{-2}	1.70×10^{-18}	3.38×10^{-1}	4.39×10^{-20}	6.31×10^{-28}
F_7	Ave	6.79×10^{-4}	6.92×10^{-2}	7.68×10^{-1}	3.88×10^{-2}	2.02×10^{-2}	5.37×10^{-1}	7.80×10^{-4}	2.71×10^{-5}	2.16×10^{-9}
	std	3.29×10^{-3}	2.87×10^{-2}	2.77	5.79×10^{-2}	7.43×10^{-3}	1.89×10^{-1}	3.85×10^{-4}	9.26×10^{-6}	1.24×10^{-7}

3.3. Evaluation of Multimodal Test Functions with High Dimensions.

The number of local responses is exponentially increased in the multimodal functions from F_8 to F_{13} by increasing the functional dimensions. Consequently, it is difficult to obtain a minimum answer to these functions. The results of optimization are shown in Table 5 for these functions. GO is now able to work and more quickly find the best possible solution.

Table 5. *Results for GO and other algorithms in Multimodal test functions*

		GA	PSO	GSA	TLBO	GOA	GWO	SHO	EPO	GO
F_8	Ave	$-5.11\times10^{+2}$	$-5.01\times10^{+2}$	$-2.75\times10^{+2}$	$-3.81\times10^{+2}$	$-6.92\times10^{+2}$	$-4.69\times10^{+1}$	$-6.14\times10^{+2}$	$-8.76\times10^{+2}$	$-1.2\times10^{+4}$
	std	$4.37\times10^{+1}$	$4.28\times10^{+1}$	$5.72\times10^{+1}$	$2.83\times10^{+1}$	$9.19\times10\times10^{+1}$	$3.94\times10^{+1}$	$9.32\times10^{+1}$	$5.92\times10^{+1}$	8.72×10^{-12}
F_9	Ave	1.23×10^{-1}	1.20×10^{-1}	$3.35\times10^{+1}$	$2.23\times10^{+1}$	$1.01\times10^{+2}$	4.85×10^{-2}	4.34×10^{-1}	6.90×10^{-1}	5.62×10^{-4}
	std	$4.11\times10^{+1}$	$4.01\times10^{+1}$	$1.19\times10^{+1}$	$3.25\times10^{+1}$	$1.89\times10^{+1}$	$3.91\times10^{+1}$	1.66	4.81×10^{-1}	3.21×10^{-2}
F_{10}	Ave	5.31×10^{-11}	5.20×10^{-11}	8.25×10^{-9}	$1.55\times10^{+1}$	1.15	2.83×10^{-8}	1.63×10^{-14}	8.03×10^{-16}	2.61×10^{-20}
	std	1.11×10^{-10}	1.08×10^{-10}	1.90×10^{-9}	8.11	7.87×10^{-1}	4.34×10^{-7}	3.14×10^{-15}	2.74×10^{-14}	2.14×10^{-18}
F_{11}	Ave	3.31×10^{-6}	3.24×10^{-6}	8.19	3.01×10^{-1}	5.74×10^{-1}	2.49×10^{-5}	2.29×10^{-3}	4.20×10^{-5}	1.56×10^{-10}
	std	4.23×10^{-5}	4.11×10^{-5}	3.70	2.89×10^{-1}	1.12×10^{-1}	1.34×10^{-4}	5.24×10^{-3}	4.73×10^{-4}	4.15×10^{-7}
F_{12}	Ave	9.16×10^{-8}	8.93×10^{-8}	2.65×10^{-1}	$5.21\times10^{+1}$	1.27	1.34×10^{-5}	3.93×10^{-2}	5.09×10^{-3}	4.87×10^{-5}
	std	4.88×10^{-7}	4.77×10^{-7}	3.14×10^{-1}	$2.47\times10^{+2}$	1.02	6.23×10^{-4}	2.42×10^{-2}	3.75×10^{-3}	3.96×10^{-4}
F_{13}	Ave	6.39×10^{-2}	6.26×10^{-2}	5.73×10^{-32}	$2.81\times10^{+2}$	6.60×10^{-2}	9.94×10^{-8}	4.75×10^{-1}	1.25×10^{-8}	0.00
	std	4.49×10^{-2}	4.39×10^{-2}	8.95×10^{-32}	$8.63\times10^{+2}$	4.33×10^{-2}	2.61×10^{-7}	2.38×10^{-1}	2.61×10^{-7}	0.00

3.4. Evaluation of Multimodal Test Functions with Low Dimensions

Functions F_{14} to F_{23} have a low local response and a low number of dimensions. The optimization results for these functions are shown in Table 6. These results demonstrate the good performance of GO in these types of problems over other algorithms.

Table 6. *Results for GO and other algorithms in Multimodal test functions with low dimension.*

		GA	PSO	GSA	TLBO	GOA	GWO	SHO	EPO	GO
F_{14}	Ave	4.39	2.77	3.61	6.79	$9.98\times10^{+1}$	1.26	3.71	1.08	9.91×10^{-1}
	std	4.41×10^{-2}	2.32	2.96	1.12	9.14×10^{-1}	6.86×10^{-1}	3.86	4.11×10^{-2}	6.52×10^{-12}
F_{15}	Ave	7.36×10^{-2}	9.09×10^{-3}	6.84×10^{-2}	5.15×10^{-2}	7.15×10^{-2}	1.01×10^{-2}	3.66×10^{-2}	8.21×10^{-3}	2.35×10^{-4}
	std	2.39×10^{-3}	2.38×10^{-3}	7.37×10^{-2}	3.45×10^{-3}	1.26×10^{-1}	3.75×10^{-3}	7.60×10^{-2}	4.09×10^{-3}	1.13×10^{-5}
F_{16}	Ave	-1.02	-1.02	-1.02	-1.01	-1.02	-1.02	-1.02	-1.02	-1.03
	std	4.19×10^{-7}	0.00	0.00	3.64×10^{-8}	4.74×10^{-8}	3.23×10^{-5}	7.02×10^{-9}	9.80×10^{-7}	4.52×10^{-10}
F_{17}	Ave	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}
	std	3.71×10^{-17}	9.03×10^{-16}	1.13×10^{-16}	9.45×10^{-15}	1.15×10^{-7}	7.61×10^{-4}	7.00×10^{-7}	5.39×10^{-5}	3.25×10^{-21}
F_{18}	Ave	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00
	std	6.33×10^{-7}	6.59×10^{-5}	3.24×10^{-2}	1.94×10^{-10}	$1.48\times10^{+1}$	2.25×10^{-5}	7.16×10^{-6}	1.15×10^{-8}	5.32×10^{-19}
F_{19}	Ave	-3.81	-3.80	-3.86	-3.73	-3.77	-3.75	-3.84	-3.86	-3.86
	std	4.37×10^{-10}	3.37×10^{-15}	4.15×10^{-1}	9.69×10^{-4}	3.53×10^{-7}	2.55×10^{-3}	1.57×10^{-3}	6.50×10^{-7}	8.67×10^{-11}
F_{20}	Ave	-2.39	-3.32	-1.47	-2.17	-3.23	-2.84	-3.27	-2.81	-3.31
	std	4.37×10^{-1}	2.66×10^{-1}	5.32×10^{-1}	1.64×10^{-1}	5.37×10^{-2}	3.71×10^{-1}	7.27×10^{-2}	7.11×10^{-1}	3.51×10^{-5}
F_{21}	Ave	-5.19	-7.54	-4.57	-7.33	-7.38	-2.28	-9.65	-8.07	-10.15
	std	2.34	2.77	1.30	1.29	2.91	1.80	1.54	2.29	2.32×10^{-3}
F_{22}	Ave	-2.97	-8.55	-6.58	-1.00	-8.50	-3.99	-1.04	-10.01	-10.40
	std	1.37×10^{-2}	3.08	2.64	2.89×10^{-4}	3.02	1.99	2.73×10^{-4}	3.97×10^{-2}	4.52×10^{-8}
F_{23}	Ave	-3.10	-9.19	-9.37	-2.46	-8.41	-4.49	$-1.05\times10^{+1}$	-3.41	-10.55
	std	2.37	2.52	2.75	1.19	3.13	1.96	1.81×10^{-4}	1.11×10^{-2}	4.62×10^{-6}

Figure 2 displays GO convergence curves and other optimization algorithms. GO is highly competitive over other algorithms for optimisation. It draws up convergence curves of three function models. Multimodal test functions with high dimensions such as F_{12} and multimodal test functions with low dimensions such as F_{15} GO converge more accurately and quickly in the search space due to its adaptive mechanism in unimodal functions such as F_5.

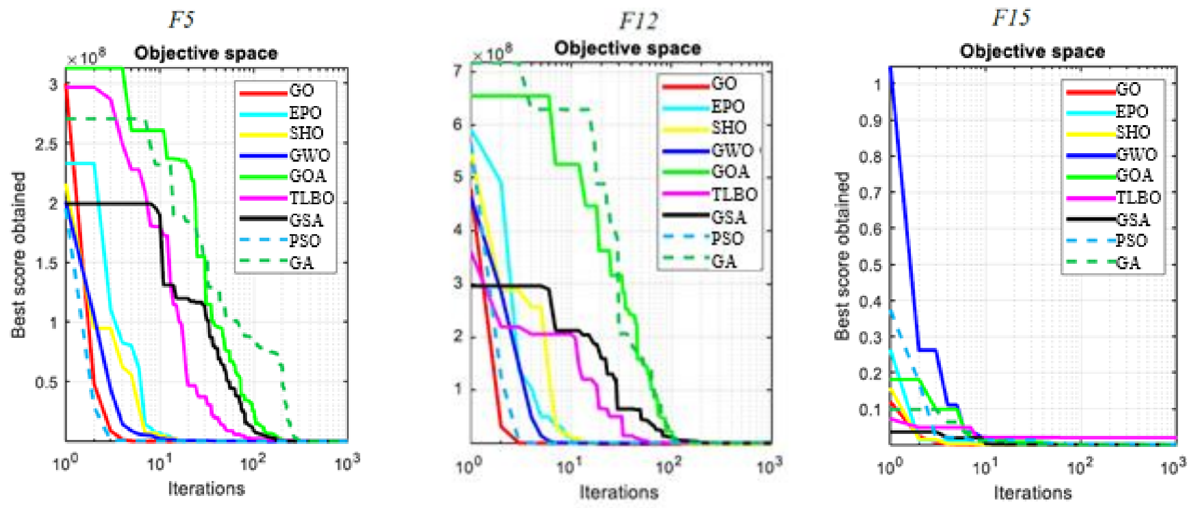

Figure 2.*Convergence curves*

3.5. Pressure Vessel Design

The proposed GO method is evaluated to optimize an engineering design problem named pressure vessel design. The objective function and complete information of this design problem are stated in [52]. The efficiency of GO and other mentioned algorithms is shown in Tables 7 and 8. The numbers in these tables indicate the superiority of the proposed GO over other methods in solving this problem.

Table 7. *Comparison results for pressure vessel design problem*

Algorithm	Optimum variables				Optimum cost
	T_s	T_h	R	L	
GO	0.778099	0.383241	40.315121	200.00000	5880.0700
EPO	0.778210	0.384889	40.315040	200.00000	5885.5773
SHO	0.779035	0.384660	40.327793	199.65029	5889.3689
GWO	0.778961	0.384683	40.320913	200.00000	5891.3879
GOA	0.845719	0.418564	43.816270	156.38164	6011.5148
TLBO	0.817577	0.417932	41.74939	183.57270	6137.3724
GSA	1.085800	0.949614	49.345231	169.48741	11550.2976
PSO	0.752362	0.399540	40.452514	198.00268	5890.3279
GA	1.099523	0.906579	44.456397	179.65887	6550.0230

Table 8. *Statistical result for pressure vessel design problem.*

Algorithms	Best	Mean	Worst	Std. Dev.	Median
GO	5880.0700	5884.1401	5891.3099	024.341	5883.5153
EPO	5885.5773	5887.4441	5892.3207	002.893	5886.2282
SHO	5889.3689	5891.5247	5894.6238	013.910	5890.6497
GWO	5891.3879	6531.5032	7394.5879	534.119	6416.1138
GOA	6011.5148	6477.3050	7250.9170	327.007	6397.4805
TLBO	6137.3724	6326.7606	6512.3541	126.609	6318.3179
GSA	11550.2976	23342.2909	33226.2526	5790.625	24010.0415
PSO	5890.3279	6264.0053	7005.7500	496.128	6112.6899
GA	6550.0230	6643.9870	8005.4397	657.523	7586.0085

3.6. Wilcoxon Signed Rank Test

In two groups depending on each other, Wilcoxon signed rank test [53] is used to compare the results. Considering fitness function, the Wilcoxon test was conducted on a level of confidence of 95% (the zero-hypothesis in this test is representative of lack of difference and the other hypothesis indicates the differences). Tables 9 and 10 show the results of Wilcoxon signed rank test on twenty-three fitness functions which number 1 means better, number 0 means equal, and number -1 means worse. These results indicate the superiority of the proposed GO algorithm over other compared algorithms.

***Table** 9. Result of wilcoxon signed rank test on F_1-F_{23}.*

	EPO	SHO	GWO	GOA	TLBO	GSA	PSO	GA
F_1	-1	-1	-1	-1	-1	-1	-1	-1
F_2	-1	-1	-1	-1	-1	-1	-1	-1
F_3	-1	-1	-1	-1	-1	-1	-1	-1
F_4	-1	-1	-1	-1	-1	-1	-1	-1
F_5	-1	-1	-1	-1	-1	-1	-1	-1
F_6	-1	-1	-1	-1	-1	-1	-1	-1
F_7	-1	-1	-1	-1	-1	-1	-1	-1
F_8	-1	-1	-1	-1	-1	-1	-1	-1
F_9	-1	-1	-1	-1	-1	-1	-1	-1
F_{10}	-1	-1	-1	-1	-1	-1	-1	-1
F_{11}	-1	-1	-1	-1	-1	-1	-1	-1
F_{12}	-1	-1	-1	-1	-1	-1	-1	-1
F_{13}	-1	-1	-1	-1	-1	-1	-1	-1
F_{14}	-1	-1	-1	-1	-1	-1	-1	-1
F_{15}	-1	-1	-1	-1	-1	-1	-1	-1
F_{16}	-1	-1	-1	-1	-1	-1	-1	-1
F_{17}	-1	-1	-1	-1	-1	-1	-1	-1
F_{18}	-1	-1	-1	-1	-1	-1	-1	-1
F_{19}	-1	-1	-1	-1	-1	-1	-1	-1
F_{20}	-1	-1	-1	-1	-1	-1	-1	-1
F_{21}	-1	-1	-1	-1	-1	-1	-1	-1
F_{22}	-1	-1	-1	-1	-1	-1	-1	-1
F_{23}	-1	-1	-1	-1	-1	-1	-1	-1

***Table 10.** Result of wilcoxon signed rank test on pressure vessel design problem.*

	EPO	SHO	GWO	GOA	TLBO	GSA	PSO	GA
Pressure vessel design	-1	-1	-1	-1	-1	-1	-1	-1

4. CONCLOUSION

In this paper, a novel optimization method called Group Optimization (GO) is introduced. GO is based on concept that uses all agents to update population of algorithm. Every agent of population could to be used for population updating. For these purpose two groups is specified for any agent. One group for good agents and another group for bad agents. These groups is used for updating position of each agent.
GO has been tested on 23 benchmark test functions. The results demonstrate that GO has good performance as compared with GA, PSO, GSA, TLBO, GWO, GOA, SHO and EPO. The results on the unimodal and multimodal test functions show the superior exploitation and exploration capability of GO.
In future works, the authors propose several ideas for study. One may create a binary variant of GO as an important potential contribution. GO may also be used to overcome many-objective real-life optimization as well as multi-objective problems.

CONFLICTS OF INTEREST

No conflict of interest was declared by the authors.

REFERENCES

[1] Mirjalili, S., "Introduction to Evolutionary Single-Objective Optimisation, in Evolutionary Algorithms and Neural Networks", Springer, 3-14, (2019).

[2] Bäck, T., Fogel, D. B., and Michalewicz, Z. "Evolutionary computation 1: Basic algorithms and operators", CRC press, (2018).

[3] Dehghani, M., Montazeri, Z., Dehghani, A., Nouri, N., and Seifi, A., "BSSA: Binary spring search algorithm", in 2017 IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI), Tehran, 0220-0224, (2017).

[4] Dehghani, M., Montazeri, Z., Dehghani, A., and Seifi, A., "Spring search algorithm: A new meta-heuristic optimization algorithm inspired by Hooke's law", in 2017 IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI), Tehran, 0210-0214, (2017).

[5] Dehghani, M., Montazeri, Z., Malik, O.P., Ehsanifar, A., and Dehghani, A., "OSA: orientation search algorithm", International Journal of Industrial Electronics, Control and Optimization, 2: 99-112, (2019).

[6] Dehghani, M., Montazeri, Z., and Malik, O.P., "DGO: dice game optimizer", Gazi University Journal of Science, 32: 871-882, (2019).

[7] Dehghani, M., Mardaneh, M., Montazeri, Z., Ehsanifar, A., Ebadi, M.J., and Grechko, O.M., "Spring search algorithm for simultaneous placement of distributed generation and capacitors", Електротехніка і Електромеханіка, 6: 68-73, (2018).

[8] Montazeri, Z., and Niknam, T., "Optimal Utilization of Electrical Energy from Power Plants Based on Final Energy Consumption Using Gravitational Search Algorithm", Електротехніка і Електромеханіка, 4: 70-73, (2018).

[9] Bekdaş, G., Nigdeli, S.M., Kayabekir, A.E., and X.-S. Yang, "Optimization in civil engineering and metaheuristic algorithms: a review of state-of-the-art developments", in Computational Intelligence, Optimization and Inverse Problems with Applications in Engineering, Springer: 111-137, (2019).

[10] Montazeri, Z., and Niknam, T., "Energy carriers management based on energy consumption", in 2017 IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI), Tehran: 0539-0543, (2017).

[11] Dehghani, M., Montazeri, Z., Ehsanifar, A., Seifi, A., Ebadi, M., and Grechko, O., "Planning of energy carriers based on final energy consumption using dynamic programming and particle swarm optimization", Електротехніка і Електромеханіка, 5: 62-71, (2018).

[12] Dehghani, M., Montazeri, Z., and Malik, O.P., "Energy commitment: a planning of energy carrier based on energy consumption", Електротехніка і Електромеханіка, 4: 69-72, (2019).

[13] AbouEisha, H., Amin, T., Chikalov, I., Hussain, S., and Moshkov, M., "Extensions of Dynamic Programming for Combinatorial Optimization and Data Mining", Springer, (2019).

[14] Antonov, I.V., Mazurov, E., Borodovsky, M., and Medvedeva, Y.A., "Prediction of lncRNAs and their interactions with nucleic acids: benchmarking bioinformatics tools", Briefings in bioinformatics, 20: 551-564, (2018).

[15] Ehsanifar, A., Dehghani, M., and Allahbakhshi, M., "Calculating the leakage inductance for transformer inter-turn fault detection using finite element method", in 2017 Iranian Conference on Electrical Engineering (ICEE), Tehran, 1372-1377, (2017).

[16] Dehbozorgi, S., Ehsanifar, A., Montazeri, Z., Dehghani, M., and Seifi, A., "Line loss reduction and voltage profile improvement in radial distribution networks using battery energy storage system", in 2017 IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI), Tehran, 0215-0219, (2017).

[17] Biswas, A., Mishra, K., Tiwari, S., and Misra, A., "Physics-inspired optimization algorithms: a survey", Journal of Optimization, (2013).

[18] Kirkpatrick, S., Gelatt, C.D., and Vecchi, M.P., "Optimization by simulated annealing", science, 220: 671-680, (1983).

[19] Rashedi, E., Nezamabadi-Pour, H., and Saryazdi, S., "GSA: a gravitational search algorithm", Information Sciences, 179: 2232-2248, (2009).

[20] Du, H., Wu, X., and Zhuang, J., "Small-world optimization algorithm for function optimization", in International Conference on Natural Computation: 264-273, (2006).

[21] Moghaddam, F.F., Moghaddam, R.F., and Cheriet, M., "Curved space optimization: A random search based on general relativity theory", arXiv preprint arXiv: 208.2214, (2012).

[22] Alatas, B., "ACROA: artificial chemical reaction optimization algorithm for global optimization", Expert Systems with Applications, 38: 13170-13180, (2011).

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

[24] Kaveh, A., and Khayatazad, M., "A new meta-heuristic method: ray optimization", Computers & structures, 112: 283-294, (2012).

[25] Shah-Hosseini, H., "Principal components analysis by the galaxy-based search algorithm: a novel metaheuristic for continuous optimisation", International Journal of Computational Science and Engineering, 6: 132-140, (2011).

[26] Hatamlou, A., "Black hole: A new heuristic optimization approach for data clustering", Information Sciences, 222: 175-184, (2013).

[27] Tayarani-N, M-H., and Akbarzadeh-T, M., "Magnetic optimization algorithms a new synthesis", in 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence): 2659-2664, (2008).

[28] Karkalos, N.E., Markopoulos, A.P., and Davim, J.P., "Evolutionary-Based Methods", in Computational Methods for Application in Industry 4.0, Springer: 11-31, (2019).

[29] Storn, R., and Price, K., "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces", Journal of Global Optimization, 11: 341-359, (1997).

[30] Mirjalili, S., "Genetic Algorithm", in Evolutionary Algorithms and Neural Networks, Springer: 43-55, (2019).

[31] Mirjalili, S., "Biogeography-Based Optimisation", in Evolutionary Algorithms and Neural Networks, Springer: 57-72, (2019).

[32] Beyer, H-G., and Schwefel, H-P., “Evolution strategies–A comprehensive introduction”, Natural Computing, 1: 3-52, (2002).

[33] Koza, J.R., “Genetic programming as a means for programming computers by natural selection”, Statistics and Computing, 4: 87-112, (1994).

[34] Lim, S.M., and Leong, K.Y., “A Brief Survey on Intelligent Swarm-Based Algorithms for Solving Optimization Problems”, in Nature-inspired Methods for Stochastic, Robust and Dynamic Optimization, ed: IntechOpen, (2018).

[35] Bansal, J.C., “Particle Swarm Optimization”, in Evolutionary and Swarm Intelligence Algorithms, Springer: 11-23, (2019).’

[36] Dorigo, M., and Stützle, T., “Ant colony optimization: overview and recent advances”, in Handbook of metaheuristics, Springer: 311-351, (2019).

[37] Yang, X.S., “A new metaheuristic bat-inspired algorithm”, in Nature inspired cooperative strategies for optimization (NICSO 2010), Springer: 65-74, (2010).

[38] Dhiman, G., and Kumar, V., “Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications”, Advances in Engineering Software, 114: 48-70, (2017).

[39] Yang, X-S., and Hossein Gandomi, A., “Bat algorithm: a novel approach for global engineering optimization”, Engineering Computations, 29: 464-483, (2012).

[40] Gandomi, A.H., Yang, X-S., and Alavi, A.H., “Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems”, Engineering with Computers, 29: 17-35, (2013).

[41] Karaboga, D., and Basturk, B., “On the performance of artificial bee colony (ABC) algorithm”, Applied Soft Computing, 8: 687-697, (2008).

[42] Dhiman, G., and Kumar, V., “Emperor penguin optimizer: A bio-inspired algorithm for engineering problems”, Knowledge-Based Systems, 159: 20-50, (2018).

[43] Mirjalili, S., “Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems”, Neural Computing and Applications, 27: 1053-1073, (2016).

[44] Dehghani, M., Mardaneh, M., Malik, O. P., and NouraeiPour, S. M., “DTO: Donkey Theorem Optimization”, in 2019 27th Iranian Conference on Electrical Engineering (ICEE), Yazd, 1855-1859, (2019).

[45] Saremi,S., Mirjalili, S., and Lewis, A., “Grasshopper optimisation algorithm: theory and application”, Advances in Engineering Software, 105: 30-47, (2017).

[46] Dehghani, M., Mardaneh, M., and Malik, O.P., “FOA: ‘Following’ Optimization Algorithm for solving Power engineering optimization problems”, Journal of Operation and Automation in Power Engineering, 8: 118-130, (2019).

[47] Mirjalili, S., Mirjalili, S.M., and Lewis, A., “Grey wolf optimizer”, Advances in Engineering Sofware, 69:46-61,(2014).

[48] Mirjalili, S., “Particle Swarm Optimisation”, in Evolutionary Algorithms and Neural Networks, Springer: 15-31, (2019).

[49] Rao, R.V., Savsani, V.J., and Vakharia, D., "Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems", Computer-Aided Design, 43: 303-315, (2011).

[50] Digalakis, J.G., and Margaritis, K.G., "On benchmarking functions for genetic algorithms", International Journal of Computer Mathematics, 77: 481-506, (2001).

[51] Wu, L., Liu, Q., Tian, X., Zhang, J., and Xiao, W., "A new improved fruit fly optimization algorithm IAFOA and its application to solve engineering optimization problems", Knowledge-Based Systems, 144: 153-173, (2018).

[52] Kannan, B., and Kramer, S.N., "An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design", Journal of Mechanical Design, 116: 405-411, (1994).

[53] Woolson, R., "Wilcoxon signed-rank test", Wiley encyclopedia of clinical trials, 1-3, (2007).