

# Elymus Repens Optimization (ERO); A Novel Agricultural-Inspired Algorithm

Mahdi Tourani<sup>1\*</sup>

<sup>1</sup>. Faculty of Engineering, University of Birjand

Received: 01 Apr 2023/ Revised: 27 July 2024/ Accepted: 20 Aug 2024

## Abstract

Optimization plays a crucial role in enhancing productivity within the industry. Employing this technique can lead to a reduction in system costs. There exist various efficient methods for optimization, each with its own set of advantages and disadvantages. Meanwhile, meta-heuristic algorithms offer a viable solution for achieving the optimal working point. These algorithms draw inspiration from nature, physical relationships, and other sources. The distinguishing factors between these methods lie in the accuracy of the final optimal solution and the speed of algorithm execution. The superior algorithm provides both precise and rapid optimal solutions. This paper introduces a novel agricultural-inspired algorithm named Elymus Repens Optimization (ERO). This optimization algorithm operates based on the behavioral patterns of Elymus Repens under cultivation conditions. Elymus repens is inclined to move to areas with more suitable conditions. In ERO, exploration and exploitation are carried out through Rhizome Optimization Operator and Stolon Optimization Operators. These two supplementary activities are used to explore the problem space. The potent combination of these operators, as presented in this paper, resolves the challenges encountered in previous research related to speed and accuracy in optimization issues. After the introduction and simulation of ERO, it is compared with popular search algorithms such as Gravitational Search Algorithm (GSA), Grey Wolf Optimizer (GWO), Particle Swarm Optimization (PSO), and Firefly Algorithm (FA). The solution of 23 benchmark functions demonstrates that the proposed algorithm is highly efficient in terms of accuracy and speed.

**Keywords:** Elymus Repens Optimization; Meta-Heuristic Algorithms; Rhizome Optimization Operator; Stolon Optimization Operator.

## 1- Introduction

Today, the industry faces various pressing problems that require urgent solutions and optimal answers. Contributing to the resolution of these issues can greatly enhance efficiency across multiple fields. There exist diverse approaches to solving optimization problems, including one-by-one counting methods, classical mathematical methods, and optimization methods.

The one-by-one method involves a significant amount of time to solve problems, rendering it practical only for small-scale issues. However, its advantages encompass very high accuracy and zero error.

Conversely, classical mathematical methods, such as derivation methods, require adherence to specific principles and rules for continuous problems. These limitations can make it challenging to employ these methods for solving optimization problems. Nonetheless, classical mathematical methods offer high accuracy, making them an appealing option.

In optimization methods, algorithms begin in an initial space and move intelligently towards an optimal solution. With effective guiding operators, these algorithms conduct smarter searches in problem spaces, ultimately accelerating the process of reaching a final answer. Several desirable features of optimization methods include:

- No limitation in problem modeling
- Universality in covering a wide range of issues
- High speed in determining the optimal answer

In this paper, a powerful method is introduced for optimizing problems by harnessing the positive features of nature to address challenges. One such valuable feature is the growth mechanism of Elymus repens in agricultural land, which provides an innovative approach to problem-solving.

The paper proceeds as follows: Section 2 provides an overview of optimization algorithms. Section 3 introduces the Elymus repens mechanism, and Section 4 presents the new algorithm called Elymus repens optimization. Finally, in Section 5, the performance of this new algorithm is evaluated using 23 sample functions.

## 2- Literature review

Today, optimization algorithms are used as a method for obtaining the optimal solutions to optimization problems [1]. Unlike classical mathematical methods, these algorithms are much more efficient in solving optimization problems. The basis of optimization algorithms is usually nature, physics, and swarm. The final answer obtained from them has high accuracy and suitable speed. Optimization algorithms use two basic components of exploration and exploitation to search the problem space. These two features are very helpful in finding the optimal answer. Exploration provides the algorithm with the ability to search freely without paying attention to the accuracy of the results. On the other hand, paying attention to the information obtained in the previous loops is the basis of exploitation. With an increase of exploration, the algorithm finds random and unpredictable directions, and on the opposite side, with an increase of exploitation, the performance of the algorithm becomes cautious. By the exploration and exploitation, the algorithm will move towards the smart answer.

In the following, some of the popular optimization algorithms are reviewed [2]:

Genetic Algorithm [3], Genetic programming [4], Tabu Search [5], Evolution Strategy [6], Memetic Algorithm [7], Cultural Algorithm [8], Simulated Annealing [9], Differential Evolution [10], Evolutionary Programming [11], Co Evolutionary Algorithm [12], Gradient Evolution Algorithm [13], Imperialistic Competitive Algorithm [14], Biogeography-Based Optimization [15], States of Matter Search [16], Sine Cosine Algorithm [17], Multi-level Cross Entropy Optimizer [18]. These algorithms are modeled on Darwin's theories.

Some algorithms are physics-based optimization algorithms such as: Small-World Optimization Algorithm [19], Central Force Optimization [20], Magnetic Optimization Algorithm [21], Gravitational Search Algorithm [22], Charged System Search [23], Chemical-Reaction Optimization [24], Black Hole [25], Curved Space Optimization [26], Water Evaporation Optimization [27], Ideal Gas Molecular Movement [28], Multi-Verse Optimizer [29], Vibrating Particles System [30].

Some optimizers are swarm-based algorithms: Particle Swarm Optimization [31], Grasshopper Optimization Algorithm [32], Moth-flame Optimization [33], Artificial Fish Swarm Algorithm [34], Honey Bee Optimization [35], Termite Colony Optimization [36], Ant Colony Optimization [37], Shuffled Frog-Leaping [38], Monkey Search [39], Dolphin Partner Optimization [40], Firefly Algorithm [41], Bat Algorithm [1], Bird Mating Optimizer [42], Fruit Fly Optimization [43], Lion Pride Optimizer [44], Krill Herd [45], Grey Wolf Optimizer [46], Cuckoo Search [47], Soccer League Competition Algorithm [48], Dragonfly Algorithm [49], Whale Optimization Algorithm

[50], Salp Swarm Algorithm [51], Harris Hawks Optimization [52], Flying Squirrel Optimizer [53], Ant Lion Optimizer [54]

In addition to these algorithms, some intelligence may be found in nature that can form the basis of other optimization algorithms. One of these is the *Elymus Repens* behavior.

The introduced algorithms are very effective in industry, energy, medicine and etc. References [55-59] in science, [60-64] in engineering and [65-69] in medical show part of the research conducted with these algorithms in the field of optimization.

## 3- *Elymus Repens*

*Elymus repens* (ER) is a highly competitive, allelopathic, perennial grass. This plant is considered one of the world's most troublesome weeds, reproducing both sexually through seeds and asexually through rhizomes. It is found in temperate regions worldwide, with the exception of Antarctica [70, 71, 72]. The structure and appearance of this plant are depicted in Fig. 1 and Fig. 2.

In Northern Europe, *Elymus repens* is a common and aggressive grass species favored by cereal-dominated crop rotations and nitrogen fertilization [73, 74]. This species can become a pernicious weed, spreading rapidly by underground rhizomes [72] and quickly forming a dense mat of roots in the soil. Even the smallest fragment of the root can regenerate into a new plant [75].

*Elymus repens* is propagated by seeds, rhizomes, or stolons. The creeping stems on the ground surface and the wire-shaped underground stems have numerous short branches and scaly leaves. New aerial organs are formed from the nodes of rhizomes and stolons.

This plant is highly resilient and can thrive in favorable conditions on the ground. These conditions include water, organic, and biological materials. Where these conditions are optimal, the growth of this plant flourishes. On the other hand, this plant can be considered as a "search engine" as it moves towards favorable agricultural positions and covers them using propagation tools such as rhizomes or stolons. Once introduced to an area, it swiftly moves to better conditions and occupies the desired area.

The power and speed of occupying fertile areas by this plant is so high that it prevents the growth of any other type of plant, thus making it one of the most destructive weeds.



Fig. 1. Elymus Repens [76]



Fig. 2. Elymus Repens in the agricultural land

#### 4- Elymus Repens Optimization

This study is centered around the behavior patterns of Elymus Repens within their cultivation environment. In terms of growth and reproduction, this plant initially progresses through seeds and subsequently through rhizomes and stolons (illustrated in Fig. 3) within the cultivation environment. Elymus repens tends to move towards any part of the soil that provides more favorable conditions.

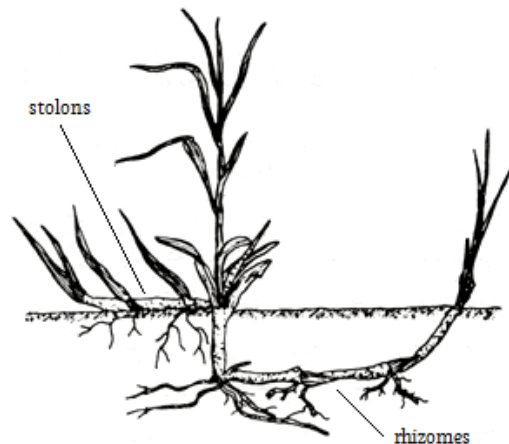


Fig. 3. Rhizomes and Stolons in Elymus Repens

In this paper, this process is modeled as a optimization search algorithm that is named Elymus Repens Optimization (ERO). In the ERO model, the cultivation land of the plant serves as the search space for the problem, with every position within this space being a candidate answer - representing a position of the land with the best cultivation conditions, i.e., the optimal answer. The rhizomes and stolons act as the ERO optimization operators.

To initiate the algorithm, Elymus repens is assumed to be spread across the environment. Any position in the cultivation environment where the reproductive parts of the plant are placed becomes an initial candidate answer. These positions are evaluated using the objective function. Subsequently, the Elymus repens will move towards the optimal answer through the use of rhizomes and stolons.

##### 4-1- Stolon Optimization Operator

Among the reproducible parts of Elymus repens, the part that is in a better environmental condition will spread to its neighboring parts through stolons. The number of neighbors for each position will increase with the improved environmental conditions. Consequently, a part of the plant that is in unfavorable conditions will not be reproduced. This process guides the initial solution towards better alternatives. Equation 1 and Equation 2 demonstrate the new candidate solutions with the stolon operator.

$$\alpha = \beta \left(1 - \frac{it}{T}\right) \quad (1)$$

$$X_{i \in neighbor}^{k,it} = X_{k \in best}^{it-1} + unifrnd(-\alpha, +\alpha) \quad (2)$$

where,  $it$  indicates iteration,  $T$  the maximum of iteration,  $X_{i \in neighbor}^{k,it}$  show  $i$ -th neighbor from  $k$ -th best position,  $X_{k \in best}^{it-1}$  the  $k$ -th best choice position,  $unifrnd$ , a uniform

random number between  $[-\alpha, +\alpha]$  and  $\beta$  is the relationship coefficient.

The best position ( $X_{i\text{best}}$ ) for reproduction is selected using the roulette wheel. This method ensures that better positions have a higher chance of reproducing. This selection process is repeated for all positions, and the resulting new neighbors are generated from the best ones.

#### 4-2- Rhizome Optimization Operator

In 4-1, the k-best position of population generate a number of neighbors. The neighbors related to each k-best position form a group. At this step, in each group, the best neighbor is selected from among the neighbors created by each previous k-best position, and the other neighbors move towards it. Equations 3 to 6 show the new candidate solutions using the rhizome operator.

$$A = \beta \times \alpha \times \text{rand} - \alpha \tag{3}$$

$$C = \beta \times \text{rand} \tag{4}$$

$$D_i^k = \text{abs}(C \times X_{\text{best neighbor}}^{k, it-1} - X_{i \in \text{other neighbor}}^{k, it-1}) \tag{5}$$

$$X_i^{it} = X_{\text{best neighbor}}^{k, it-1} - A \times D_i^k \tag{6}$$

where,  $X_i^{it}$  is new candidate answer, rand shows the random value between  $[0,1]$  and  $X_{\text{best neighbor}}^{k, it-1}$  and  $X_{i \in \text{other neighbor}}^{k, it-1}$ , are the best neighbor and other neighbors for  $k$ -th neighborhood group. Fig. 4 shows the visual performance of rhizomes and stolons operators in ER optimization.

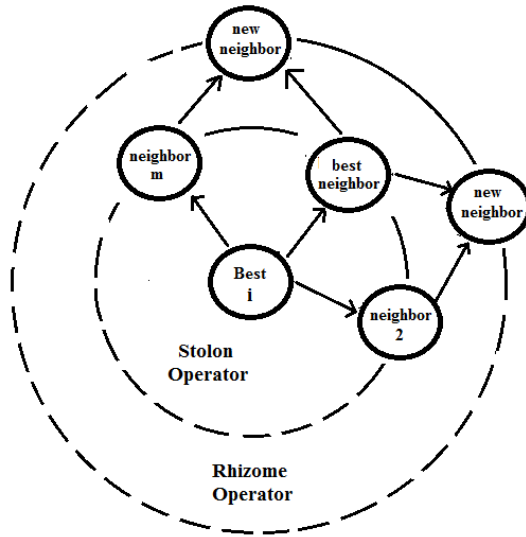


Fig. 4. The stolon and rhizome operators view

The flowchart and the pseudo code of ERO algorithm are presented in Fig. 5 and Fig. 6.

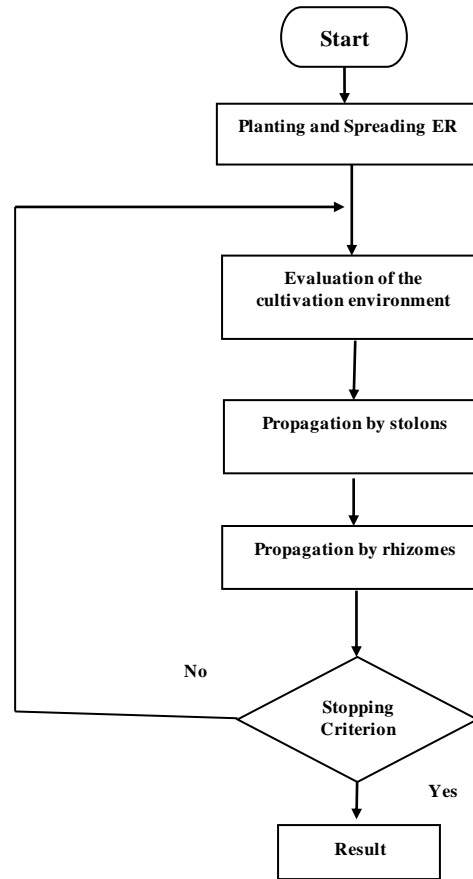


Fig. 5. Flowchart of the Proposed ERO Algorithm

Step 1: Planting the elymus repen and spreading it in the cultivation environment  
 Step 2: Evaluation of the cultivation environment  
 Step 3: Propagation of the elymus repen by stolons  
 Step 4: Propagation of the elymus repen by rhizomes  
 Step 5: if stop criteria has not been reached, Go to step 2.

Fig. 6. The pseudo code of ERO algorithm

### 5- Validation and Computational Experiment

To demonstrate the effectiveness and power of Elymus Repens Optimization as proposed in this paper, it has been evaluated for minimizing 23 case study functions [77]. Table 2 depicts these well-known functions. For the computational testing, the simulations were run on a PC with a 2.30GHz Intel Core i5 processor and 6 gigabytes of RAM.

The aim of the algorithm presented is to minimize the functions listed in the first and second columns of Table 2 in the shortest possible time. The number of variables and function constraints are provided in the fourth and fifth

columns, establishing upper and lower bounds for the function variables. The two-dimensional representations of these functions can be seen in Fig.7 and Fig. 8.

The evaluation of computational algorithms is typically gauged using two criteria: 1- The accuracy of the final solution 2- The computational speed. In this section, following the determination of these criteria for the aforementioned functions, the performance of ERO will be compared with Gray Wolf Optimization (GWO), Gravitational Search Algorithm (GSA), Particle Swarm Optimization (PSO), and Firefly Algorithm (FA).

The Gray Wolf Optimizer (GWO), introduced in 2014, is a novel meta-heuristic inspired by the hunting behavior of gray wolves. This algorithm emulates the hierarchical structure of gray wolf packs, utilizing four distinct types of wolves - alpha, beta, delta, and omega - in its simulation. The process involves three primary hunting stages: searching for prey, surrounding the prey, and ultimately attacking the prey [47].

Table 2. The 23 Benchmark Functions used in experimental study [77]

Name Function	Function	n	Range
<i>Sphere Model</i>	$F_1(x) = \sum_{i=1}^n x_i^2$	30	[-100,100]
<i>Schwefel's problem 2.22</i>	$F_2(x) = \sum_{i=1}^n  x_i  + \prod_{i=1}^n  x_i $	30	[-10,10]
<i>Schwefel's problem 1.2</i>	$F_3(x) = \sum_{i=1}^n \left( \sum_{j=1}^i x_j \right)$	30	[-100,100]
<i>Schwefel's problem 2.21</i>	$F_4(x) = \max_i \{ x_i , 1 \leq i \leq n\}$	30	[-100,100]
<i>Generalized Rosenbrock's function</i>	$F_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	[-30,30]
<i>Step function</i>	$F_6(x) = \sum_{i=1}^n [x_i + 0.5]^2$	30	[-100,100]
<i>Quartic function with noise</i>	$F_7(x) = \sum_{i=1}^n ix_i^4 + \text{random}([0,1])$	30	[-1.28,1.28]
<i>Generalized Schwefel's problem 2.26</i>	$F_8(x) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	30	[-500,500]
<i>Generalized Rastrigin's Function</i>	$F_9(x) = \sum_{i=1}^n (x_i^2 - 10 \cos \cos(2\pi x_i) + 10)$	30	[-5.12,5.12]

*Ackley's function*  $f_{10}(x) = -20 \exp(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}) - \exp(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)) + 20 + e \quad 30 \quad [-32,32]$

Table 2. The 23 Benchmark Functions used in experimental study [77] (continues)

Name Function	Function	n	Range
<i>Generalized Griewank Function</i>	$F_{11}(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos \cos \left( \frac{x_i}{\sqrt{i}} \right) + 1$	30	[-600,600]
<i>Generalized Penalized Functions</i>	$F_{12}(x) = \frac{\pi}{n} \times \{ 10 \sin^2(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 (1 + 10 \sin^2(\pi y_{i+1})) + (y_n - 1) \} + \sum_{i=1}^n u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{1}{4}(x_i + 1)$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a \leq x_i \leq a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	30	[-50,50]
<i>Generalized Penalized Functions</i>	$F_{13}(x) = 0.1 \{ \sin^2(3\pi x_1) + \sum_{i=1}^{n-1} (x_i - 1)^2 (1 + \sin^2(3\pi x_{i+1})) + (x_n - 1)^2 (1 + \sin^2(2\pi x_n)) \} + \sum_{i=1}^n u(x_i, 5, 100, 4)$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a \leq x_i \leq a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	30	[-50,50]
<i>Shekel's Foxholes function</i>	$F_{14}(x) = \left[ \frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - \alpha_{ij})^6} \right]^{-1}$	2	[-65.536,65.536]
<i>Kowalik's function</i>	$F_{15}(x) = \sum_{i=1}^{11} [a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}]^2$	4	[-5,5]

<i>Six-hump camel back function</i>	$F_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	4	[-5,5]
-------------------------------------	-------------------------------------------------------------------------------	---	--------

Table 2. The 23 Benchmark Functions used in experimental study [77] (continues)

Name Function	Function	n	Range
<i>Branin function</i>	$F_{17}(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})\cos x_1 + 10$	2	[-5,5]×[0,10]
<i>Goldstein-Price function</i>	$F_{18}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2^2 - 36x_1x_2 + 27x_2^2)]$	2	[-2,2]
<i>Hartman's family</i>	$F_{19}(x) = -\sum_{i=1}^4 c_i \exp[-\sum_{j=1}^3 a_{ij}(x_j - p_{ij})^2]$	3	[0,1]
<i>Hartman's family</i>	$F_{20}(x) = -\sum_{i=1}^4 c_i \exp[-\sum_{j=1}^6 a_{ij}(x_j - p_{ij})^2]$	6	[0,1]
<i>Shekel's family</i>	$F_{21}(x) = -\sum_{i=1}^5 [(x - a_i)(x - a_i)^T + c_i]^{-1}$	4	[0,10]
<i>Shekel's family</i>	$F_{22}(x) = -\sum_{i=1}^7 [(x - a_i)(x - a_i)^T + c_i]^{-1}$	4	[0,10]
<i>Shekel's family</i>	$F_{23}(x) = -\sum_{i=1}^{10} [(x - a_i)(x - a_i)^T + c_i]^{-1}$	4	[0,10]



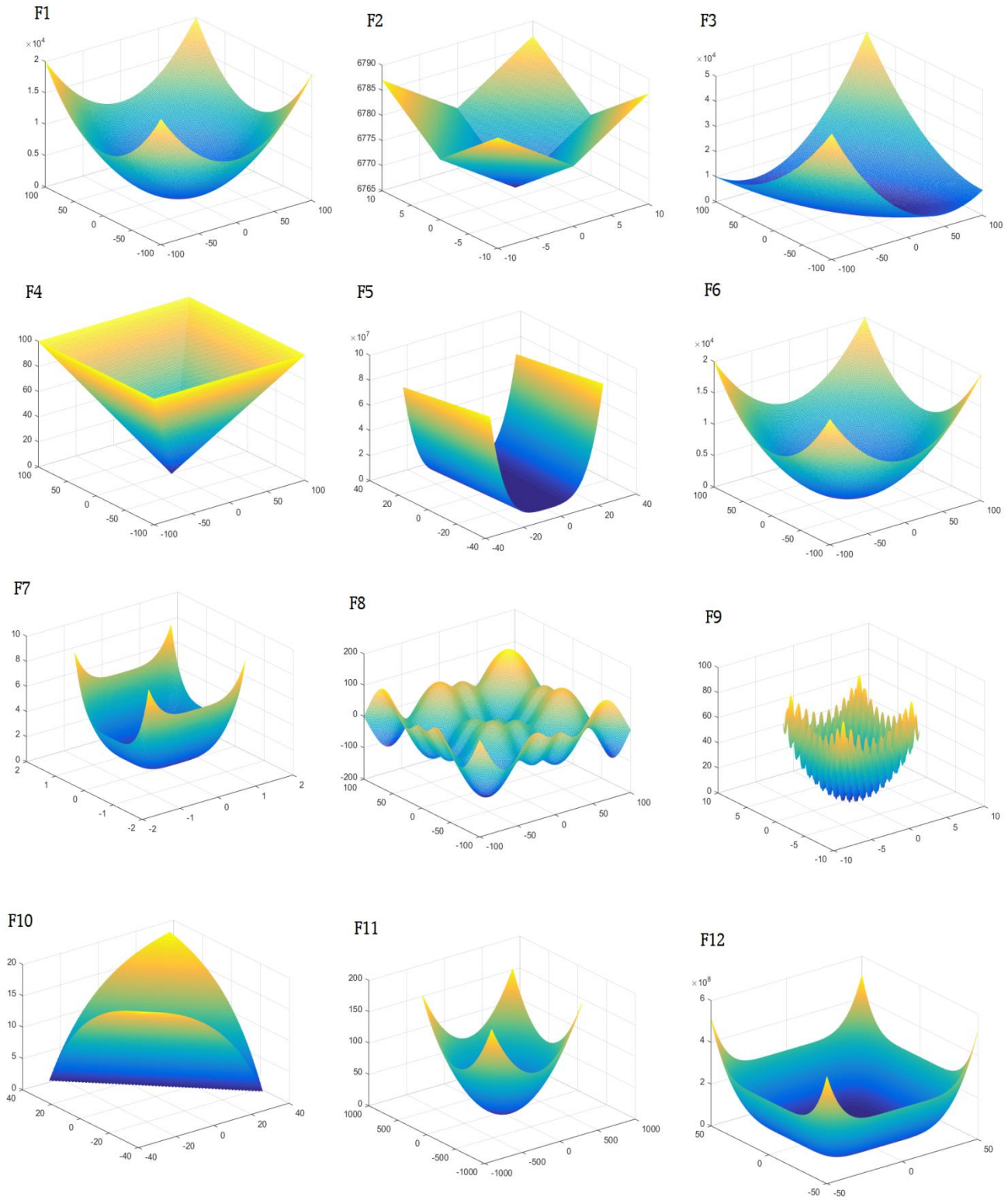
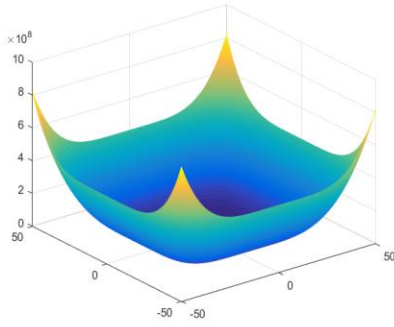


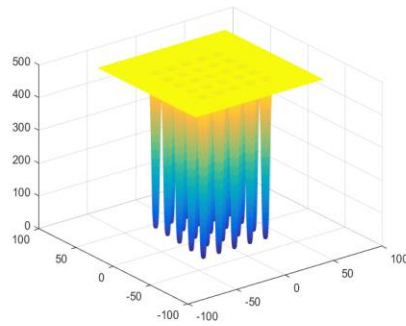
Fig. 7. Graphs of functions (F1 - F12) for n=2



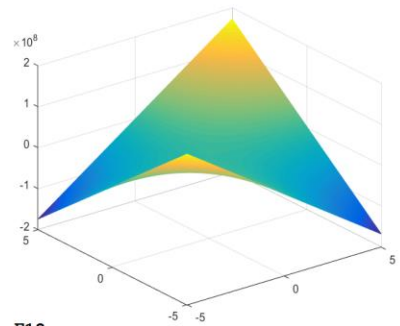
F13



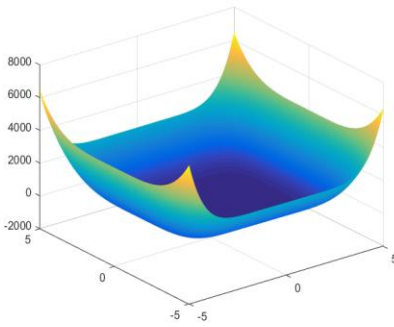
F14



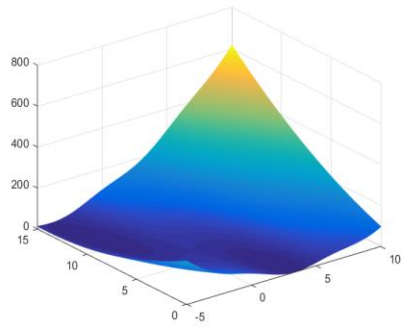
F15



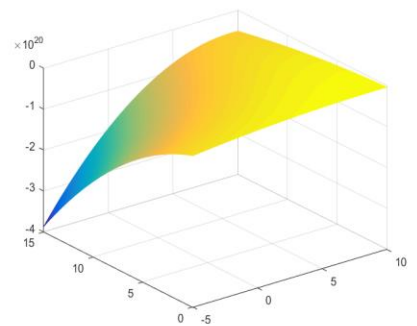
F16



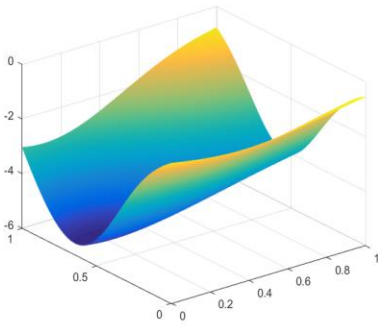
F17



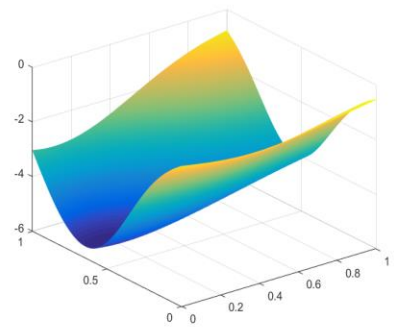
F18



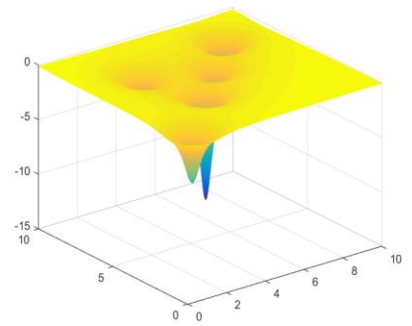
F19



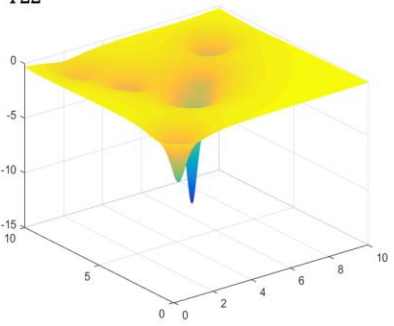
F20



F21



F22



F23

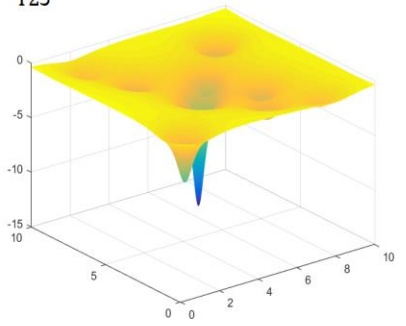


Fig. 8. Graphs of functions (F13-F23) for n=2

Another algorithm discussed in this paper is the Gravitational Search Algorithm (GSA), which operates based on physical laws such as gravity. In GSA, a collection of masses follows rules of movement that affect each other and lead to an improved final optimal answer. GSA was developed in 2009 [78].

The Firefly Algorithm (FA) was introduced by Xinshe Yang, a scholar from Cambridge, in 2008 [79]. FA is a random search algorithm inspired by swarm intelligence, simulating the attraction mechanism between individual fireflies in nature [80].

In 1995, an algorithm based on the intelligent collective behavior of animals in nature was discovered for stochastic optimization. This algorithm, called Particle Swarm Optimization (PSO), has seen advanced versions published. Numerous studies have been conducted and published regarding the effects of its parameters [81].

Table 3 presents the simulation results of the algorithm introduced in this paper (ERO) and compares it with PSO, GSA, FA, and GWO. The results include the mean run time and mean fitness of the best values found in 30 independent runs with separate seeds. The best accuracy of the algorithms is highlighted in green in Table 3, while yellow indicates that ERO is the second-most accurate.

Table 3. The average of final best fitness and the mean running time for 30 runs of minimizing benchmark functions, number of iterations=100

Algorithm	PSO	GSA	FA	GWO	ERO	
F <sub>1</sub> (x)	Mean Fitness	2.75	59137	0.26	1.61×10 <sup>-5</sup>	6.23×10 <sup>-6</sup>
	Mean Time	3.29	11.89	17.6	0.65	0.086
F <sub>2</sub> (x)	Mean Fitness	0.55	2.22	2.37	6.92×10 <sup>-4</sup>	0.01
	Mean Time	3.73	12.43	15.72	0.79	0.09
F <sub>3</sub> (x)	Mean Fitness	1035	99701	1278	17.85	4.6×10 <sup>-3</sup>
	Mean Time	3.27	21.8	13.37	0.64	0.089
F <sub>4</sub> (x)	Mean Fitness	4.75	82.65	6.25	0.19	3.07×10 <sup>-4</sup>
	Mean Time	3.56	14.63	13.16	0.44	0.09
F <sub>5</sub> (x)	Mean Fitness	252	3.85×10 <sup>7</sup>	294	28.23	9.2×10 <sup>-3</sup>
	Mean Time	3.4	13.42	13.37	0.49	0.09
F <sub>6</sub> (x)	Mean Fitness	6.67	5.96×10 <sup>4</sup>	0.87	0.03	0
	Mean Time	3.49	23.9	11.74	0.63	0.087
F <sub>7</sub> (x)	Mean Fitness	0.029	0.27	0.046	0.005	0.0098
	Mean Time	2.97	22.9	14.09	0.77	0.10
F <sub>8</sub> (x)	Mean Fitness	-67993	-2546	-2705	-6003	-10727
	Mean	2.96	13.97	12.85	0.87	0.01

		Time				
Algorithm		PSO	GSA	FA	GWO	ERO
F <sub>9</sub> (x)	Mean Fitness	37.54	62	97	21.16	0.0027
	Mean Time	2.91	21.63	12.21	0.6548	0.09
F <sub>10</sub> (x)	Mean Fitness	1.42	19	0.91	0.001	0.0014
	Mean Time	3.03	24.83	6.43	1.138	0.1
F <sub>11</sub> (x)	Mean Fitness	0.97	563	0.21	0.018	5.8×10 <sup>-7</sup>
	Mean Time	3.43	24.9	11.26	0.5157	0.09
F <sub>12</sub> (x)	Mean Fitness	0.54	2.52×10 <sup>8</sup>	0.29	0.071	3.08×10 <sup>-7</sup>
	Mean Time	4.49	22.82	13.5	1.912	0.14
F <sub>13</sub> (x)	Mean Fitness	0.5	4.91×10 <sup>8</sup>	1.17	0.79	2.6×10 <sup>-7</sup>
	Mean Time	4.88	18.91	12.55	1.69	0.13
F <sub>14</sub> (x)	Mean Fitness	1.75	1.77	3.91	2.18	6.14
	Mean Time	3.88	6.62	12.73	0.72	0.11
F <sub>15</sub> (x)	Mean Fitness	0.001	0.001	0.001	0.0032	7.3×10 <sup>-7</sup>
	Mean Time	0.99	15.99	12.49	0.4846	0.09
F <sub>16</sub> (x)	Mean Fitness	-1.03	-1.03	-1.03	-1.03	-1.00
	Mean Time	3.45	15.2	5.89	0.33	0.08
F <sub>17</sub> (x)	Mean Fitness	0.4	0.4	0.4	0.4	2.29
	Mean Time	3.44	12.58	13.25	0.37	0.08
F <sub>18</sub> (x)	Mean Fitness	-592103	-576415	-592103	-529210	-591830
	Mean Time	3.96	13.92	12.29	0.3452	0.08
F <sub>19</sub> (x)	Mean Fitness	-3.89	-3.85	-3.86	-3.86	-3.67
	Mean Time	3.35	14.68	12.06	0.45	0.09
F <sub>20</sub> (x)	Mean Fitness	-3.27	-3.03	-3.22	-3.23	-2.40
	Mean Time	4.02	15.28	13.22	0.53	0.09
F <sub>21</sub> (x)	Mean Fitness	-6.97	-6.24	-7.81	-9.54	-9.86
	Mean Time	5.05	14.31	12.72	0.97	0.13
F <sub>22</sub> (x)	Mean Fitness	-7.88	-8.82	-10.14	-10.12	-10.37
	Mean Time	5.24	14.39	13.9	1.21	0.13
F <sub>23</sub> (x)	Mean Fitness	-6.6	-8.82	-10.53	-10.24	-10.45
	Mean Time	5.57	17.1	14.37	1.68	0.14

To obtain the performance rating for these 5 algorithms (ERO, PSO, GSA, FA and GWO), the Eq. (7) is suggested:

$$Mean R = \frac{\sum_{i=1}^5 (\#Rank i) \times i}{\sum_{i=1}^5 i} \tag{7}$$

where *Mean R* indicates the average weighted rank and *#Rank i* represents the number of rank *i* in all test functions. Tables 4 and 5 display the results of the number of ranks for each algorithm across all test functions, as well as the final rank among the algorithms based on Eq. (7). In these tables, *#Ri* denotes the number of rank *i* in all test functions. The green color in Tables 4 and 5 highlights the best performance of the algorithms.

Table 4. The results of the number of accuracy ranks for each algorithm in the all test functions and the final rank among the algorithms

	#R1	#R2	#R3	#R4	#R5	Mean R	Final Rank
<b>ERO</b>	15	2	1	0	5	2.93	1
<b>GWO</b>	1	14	5	1	2	3.86	2
<b>PSO</b>	6	3	6	6	2	4.27	3
<b>FA</b>	1	3	10	7	2	5	4
<b>GSA</b>	0	1	1	9	12	6.73	5

Table 5. The results of the number of running time ranks for each algorithm in the all test functions and the final rank among the algorithms

	#R1	#R2	#R3	#R4	#R5	Mean R	Final Rank
<b>ERO</b>	23	0	0	0	0	1.53	1
<b>GWO</b>	0	23	0	0	0	3.06	2
<b>PSO</b>	0	0	23	0	0	4.6	3
<b>GSA</b>	0	0	19	4	0	4.86	4
<b>FA</b>	0	0	4	19	0	5.86	5

When comparing algorithms to determine the best performance, both speed and accuracy should be considered together. Therefore, based on the results, it is evident that Elymus Repens Optimization (ERO) demonstrates the best overall performance in terms of accuracy and speed indexes.

### 6- Conclusions and Future Work

Optimization is one of the most important processes in the industry. Among the various methods, meta-heuristic algorithms are the most powerful for optimization. This paper introduces a new algorithm called Elymus Repens Optimization (ERO) based on the behavior of Elymus Repens in agricultural land. The effectiveness and power of ERO are then evaluated using 23 well-known benchmark functions to demonstrate its capabilities. Following this simulation, the performance of ERO is compared with other optimization algorithms such as Gray Wolf Optimization (GWO), Firefly Algorithm (FA),

Particle Swarm Optimization (PSO), and Gravitational Search Algorithm (GSA). Results indicate that the proposed algorithm is highly efficient in terms of accuracy and speed.

Based on the desirable result of the algorithm, presented in this paper (ERO), it is recommended that this be implemented for optimization problems in the industry.

### References

- [1]. S.A. Mirjalili, "The Ant Lion Optimizer", *Advances in Engineering Software*, Vol. 83, pp. 80–98, 2015.
- [2]. F. MiarNaeimi, G.R. Azizyan, M. Rashki, "Horse herd optimization algorithm: A nature-inspired algorithm for high-dimensional optimization problems", *Knowledge-Based Systems*, Vol. 213, pp. 1-17, 2021.
- [3]. J.H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, MIT press, 1992.
- [4]. J.R. Koza, *Genetic Programming: On the Programming of Computers By Means of Natural Selection*, MIT press, 1992.
- [5]. F. Glover, "Tabu search—Part I", *ORSA J. Comput.* Vol. 1, No. 3, pp.190–206, 1989.
- [6]. I. Rechenberg, J.M. Zurada, R.J. Marks II, C. Goldberg, *Evolution strategy, in computational intelligence: Imitating life*, in: *Computational Intelligence Imitating Life*, IEEE Press, Piscataway, 1994.
- [7]. N.J. Radcliffe, P.D. Surry, "Formal Memetic Algorithms", in: *AISB Workshop on Evolutionary Computing*, Springer, pp. 1–16, 1994.
- [8]. R.G. Reynolds, "An introduction to cultural algorithms", in: *Proceedings of the Third Annual Conference on Evolutionary Programming*, World Scientific, pp. 131–139, 1994.
- [9]. S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, "Optimization by simulated annealing", *Science*, Vol. 220, No. 4598, pp. 671–680, 1983.
- [10]. R. Storn, K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces", *J. Global Optim.* Vol. 11, No.4, pp. 341–359, 1997.
- [11]. X. Yao, Y. Liu, G. Lin, "Evolutionary programming made faster", *IEEE Trans. Evol. Comput.* Vol. 3, No. 2, pp. 82–102, 1999.
- [12]. Y.K. Kim, J.Y. Kim, Y. Kim, "A coevolutionary algorithm for balancing and sequencing in mixed model assembly lines", *Appl. Intell.* Vol. 13, No. 3, pp. 247–258, 2000.
- [13]. A. Sinha, D.E. Goldberg, "A Survey of Hybrid Genetic and Evolutionary Algorithms", *IlliGAL report*, Vol. 2003004, 2003.
- [14]. E. Atashpaz-Gargari, C. Lucas, "Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition", in: *2007 IEEE Congress on Evolutionary Computation*, IEEE, pp. 4661–4667, 2007.
- [15]. D. Simon, "Biogeography-based optimization", *IEEE Trans. Evol. Comput.* Vol. 12, No. 6, pp. 702–713, 2008.
- [16]. E. Cuevas, A. Echavarría, M.A. Ramírez-Ortegón, "An optimization algorithm inspired by the states of matter

- that improves the balance between exploration and exploitation", *Appl. Intell.* Vol. 40, No. 2, pp. 256–272, 2014.
- [17]. S. Mirjalili, "SCA: A sine cosine algorithm for solving optimization problems", *Knowl.-Based Syst.*, Vol. 96, pp. 120–133, 2016.
- [18]. F. MiarNaeimi, G. Azizyan, M. Rashki, "Multi-level cross entropy optimizer (MCEO): An evolutionary optimization algorithm for engineering problems", *Eng. Comput.*, Vol. 34, No. 4, 2018.
- [19]. H. Du, X. Wu, J. Zhuang, "Small-world optimization algorithm for function optimization", in: *International Conference on Natural Computation*, Springer, pp. 264–273, 2006.
- [20]. R.A. Formato, "Central force optimization: A new metaheuristic with applications in applied electromagnetics", in: *Progress in Electromagnetics Research*, PIER 77, pp. 425–491, 2007.
- [21]. M.H. Tayarani-N, M.R. Akbarzadeh-T, "Magnetic optimization algorithms a new synthesis", in: *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pp. 2659–2664, 2008.
- [22]. E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, "GSA: A gravitational search algorithm", *Inf. Sci.*, Vol. 179, No. 13, pp. 2232–2248, 2009.
- [23]. A. Kaveh, S. Talatahari, "A novel heuristic optimization method: Charged system search", *Acta Mech.* Vol. 213, pp. 267–289, 2010.
- [24]. A.Y.S. Lam, V.O.K. Li, "Chemical-reaction-inspired metaheuristic for optimization", *IEEE Trans. Evol. Comput.*, Vol. 14, No 3, pp. 381–399, 2010.
- [25]. A. Hatamlou, "Black hole: A new heuristic optimization approach for data clustering", *Inf. Sci.*, Vol. 222, pp. 175–184, 2013.
- [26]. F.F. Moghaddam, R.F. Moghaddam, M. Cheriet, "Curved space optimization: A random search based on general relativity theory", *arXiv*, Vol. 1208, No. 2214, 2012.
- [27]. A. Kaveh, T. Bakhshpoori, "Water evaporation optimization: A novel physically inspired optimization algorithm", *Comput. Struct.*, Vol. 167, pp. 69–85, 2016.
- [28]. H. Varae, M.R. Ghasemi, "Engineering optimization based on ideal gas molecular movement algorithm", *Eng. Comput.* Vol. 33, No. 1, pp. 71–93, 2017.
- [29]. S. Mirjalili, S.M. Mirjalili, A. Hatamlou, "Multi-verse optimizer: A natureinspired algorithm for global optimization", *Neural Comput. Appl.*, Vol. 27, No. 2, pp. 495–513, 2016.
- [30]. A. Kaveh, M.I. Ghazaan, "A new meta-heuristic algorithm: Vibrating particles system", *Sci. Iran. Trans. A Civ. Eng.*, Vol. 24, No 2, pp. 551-566, 2017.
- [31]. R. Eberhart, J. Kennedy, "A new optimizer using particle swarm theory", in: *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, IEEE, pp. 39–43, 1995.
- [32]. S. Saremi, S. Mirjalili, A. Lewis, "Grasshopper optimisation algorithm: Theory and application", *Adv. Eng. Softw.*, Vol. 105, pp. 30–47, 2017.
- [33]. S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm", *Knowl.-Based Syst.*, Vol. 89, pp.228–249, 2015.
- [34]. X.L. Li, "A New Intelligent Optimization-Artificial Fish Swarm Algorithm", (Doctor thesis), Zhejiang University of Zhejiang, China, 2003.
- [35]. D. Karaboga, "An Idea Based on Honey Bee Swarm for Numerical Optimization", Technical report-tr06, Erciyes university, engineering faculty, computer., 2005.
- [36]. M. Roth, "Termite: A swarm intelligent routing algorithm for mobile wireless ad-hoc networks", Presented to the Faculty of the Graduate School of Cornell University in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy, 2005.
- [37]. M. Dorigo, M. Birattari, T. Stutzle, "Ant colony optimization", *IEEE Comput. Intell. Mag.* Vol. 1, No. 4, pp. 28–39, 2006.
- [38]. M. Eusuff, K. Lansey, F. Pasha, "Shuffled frog-leaping algorithm: A memetic meta-heuristic for discrete optimization", *Eng. Optim.*, Vol. 38, No. 2, pp. 129–154, 2006.
- [39]. A. Mucherino, O. Seref, "Monkey search: A novel metaheuristic search for global optimization", in: *AIP Conference Proceedings*, American Institute of Physics, pp. 162–173, 2007.
- [40]. Y. Shiqin, J. Jianjun, Y. Guangxing, "A dolphin partner optimization", in: *Intelligent Systems, GCIS'09. WRI Global Congress On, IEEE*, pp. 124–128, 2009.
- [41]. X.S. Yang, "Firefly algorithm, stochastic test functions and design optimisation", *arXiv*, Vol. 1003, No. 1409, 2010.
- [42]. X.S. Yang, "A new metaheuristic bat-inspired algorithm", in: *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, Springer, pp. 65–74, 2010.
- [43]. A. Askarzadeh, A. Rezaadeh, "A new heuristic optimization algorithm for modeling of proton exchange membrane fuel cell: Bird mating optimizer", *Int. J. Energy Res.*, Vol. 37, No. 10, pp.1196–1204, 2013.
- [44]. W.T. Pan, "A new fruit fly optimization algorithm: Taking the financial distress model as an example", *Knowl.-Based Syst.*, Vol. 26, pp. 69–74, 2012.
- [45]. B. Wang, X. Jin, B. Cheng, "Lion pride optimizer: An optimization algorithm inspired by lion pride behavior", *Sci. China Inf. Sci.*, Vol. 55, No. 10, pp. 2369–2389, 2012.
- [46]. A.H. Gandomi, A.H. Alavi, "Krill herd: A new bio-inspired optimization algorithm", *Commun. Nonlinear Sci.*, Vol. 17, No. 12, pp. 4831–4845, 2012.
- [47]. S. Mirjalili, S.M. Mirjalili, A. Lewis, "Grey wolf optimizer", *Adv. Eng. Softw.*, Vol. 69, pp. 46–61, 2014.
- [48]. A.H. Gandomi, X.S. Yang, A.H. Alavi, "Cuckoo search algorithm: A metaheuristic approach to solve structural optimization problems", *Eng. Comput.*, Vol. 29, No. 1, pp. 17–35, 2013.
- [49]. S. Mirjalili, "Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems", *Neural Comput. Appl.*, Vol. 27, No. 4, pp. 1053–1073, 2016.
- [50]. S. Mirjalili, "A. Lewis, The whale optimization algorithm", *Adv. Eng. Softw.*, Vol. 95, pp. 51–67, 2016.
- [51]. S. Mirjalili, A.H. Gandomi, S.Z. Mirjalili, S. Saremi, H. Faris, S.M. Mirjalili, "Salp swarm algorithm: A bio-inspired optimizer for engineering design problems", *Adv. Eng. Softw.*, Vol. 114, pp.163–191, 2017.

- [52]. A.A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, "Harris hawks optimization: Algorithm and applications", *Future Gener. Comput. Syst.*, Vol. 97 pp. 849–872, 2019.
- [53]. G. Azizyan, F. Miarnaemi, M. Rashki, N. Shabakhty, "Flying squirrel optimizer (FSO): A novel SI-based optimization algorithm for engineering problems", *Iran. J. Optim.*, Vol. 11, No. 2, pp.177–205, 2019.
- [54]. N. Moosavian, B.K. Roodsari, "Soccer league competition algorithm: A novel meta-heuristic algorithm for optimal design of water distribution networks", *Swarm Evol. Comput.*, Vol. 17, pp. 14–24, 2014.
- [55]. A.A. Volk, R.W. Epps, D.T. Yonemoto, S. B.Masters, F. N. Castellano, K. G. Reyes, M. Abolhasani, "AlphaFlow: autonomous discovery and optimization of multi-step chemistry using a self-driven fluidic lab guided by reinforcement learning", *Nat Commun*, Vol. 14, 2023.
- [56]. A.M.K. Nambiar, C. P. Breen, T. Hart, T. Kulesza, T. F. Jamison, K. F. Jensen, "Bayesian optimization of computer-proposed multistep synthetic routes on an automated robotic flow platform", *ACS Cent. Sci.* Vol. 8, pp. 825–836, 2022
- [57]. Y. Jiang, D. Salley, A. Sharma, G. Keenan, M. Mullin, L. Cronin, "An artificial intelligence enabled chemical synthesis robot for exploration and optimization of nanomaterials", *Sci. Adv.*, Vol. 8, 2022.
- [58]. D. Karan, G. Chen, N. Jose, J. Bai, P. McDaid, A.A. Lapkin, "A machine learning-enabled process optimization of ultra-fast flow chemistry with multiple reaction metrics", *Reaction Chemistry & Engineering*, vol. 9, pp. 619-629, 2024.
- [59]. G.-N. Ahn, J.H. Kang, H.J. Lee, B.E. Park, M. Kwon, G.S. Na, H. Kim, D.H. Seo, D.P. Kim., "Exploring ultrafast flow chemistry by autonomous self-optimizing platform", *Chem. Eng. J.*, Vol. 453, 2023.
- [60]. M. Gholami, S.M. Muyeen, S. Lin, "Optimizing microgrid efficiency: Coordinating commercial and residential demand patterns with shared battery energy storage", *Journal of Energy Storage*, Volume 88, 2024.
- [61]. D. Borkowski, P. Oramus, M. Brzezinka, "Battery energy storage system for grid-connected photovoltaic farm – energy management strategy and sizing optimization algorithm", *J. Energy Storage*, Vol. 72, 2023
- [62]. K. Ullah, J. Quanyuan, G. Geng, R.A. Khan, S. Aslam, W. Khan, "Optimization of demand response and power-sharing in microgrids for cost and power losses", *Energies*, Vol. 15, 2022.
- [63]. S. Sakina, Zaidi, S.S. Haider Zaidi, B.M. Khan, L. Moin, "Optimal designing of grid-connected microgrid systems for residential and commercial applications in Pakistan", *Heliyon*, Vol. 9, 2023.
- [64]. R. Asri, H. Aki, D. Kodaira, "Optimal operation of shared energy storage on islanded microgrid for remote communities", *Sustain. Energy, Grids Networks*, Vol. 35, 2023.
- [65]. Q. Huang, H. Ding, N. Razmjoooy, "Oral cancer detection using convolutional neural network optimized by combined seagull optimization algorithm", *Biomedical Signal Processing and Control*, Vol. 87, Part B, 2024.
- [66]. M. M. Emam, E. H. Houssein, N. A. Samee, M. A. Alohal, M. E. Hosney, "Breast cancer diagnosis using optimized deep convolutional neural network based on transfer learning technique and improved Coati optimization algorithm", *Expert Systems with Applications*, Vol. 255, Part B, 2024.
- [67]. S. Almutairi, S. Manimurugan, B. G. Kim, M.M. Aborokbah, C. Narmatha, "Breast cancer classification using Deep Q Learning (DQL) and gorilla troops optimization (GTO)", *Applied Soft Computing*, Vol. 142, 2023
- [68]. M. M. Emam, N. A. Samee, M. M. Jamjoom, E. H. Houssein, "Optimized deep learning architecture for brain tumor classification using improved Hunger Games Search Algorithm", *Computers in Biology and Medicine*, Vol. 160, 2023.
- [69]. W. Zou, X. Luo, M. Gao, C. Yu, X. Wan, S. Yu, Y. Wu, A. Wang, W. Fenical, Z. Wei, Y. Zhao, Y. Lu, "Optimization of cancer immunotherapy on the basis of programmed death ligand-1 distribution and function", Vol. 181, Themed Issue: Cancer Microenvironment and Pharmacological Interventions, pp. 257-272, 2024.
- [70]. J. Palmer, G. Sagar, "Agropyron repens (L.) Beauv. (Triticum repens L.; Elytrigia repens (L.) Nevski)", *J. Ecol.*, Vol. 51, pp. 783–794, 1963.
- [71]. P.A. Werner, R. Rioux, "The biology of Canadian weeds. 24. Agropyron repens (L.) Beauv. Can." *J. Plant Sci.*, Vol. 57, pp. 905–919, 1977.
- [72]. L.G. Holm, D.L. Plucknett, J.V. Pancho, J.P. Herberger, *The World's Worst Weeds*, University Press: Honolulu, HI, USA, 1977.
- [73]. C. Andreasen, I.M. Skovgaard, "Crop and soil factors of importance for the distribution of plant species on arable fields in Denmark", *Agric. Ecosyst. Environ.*, Vol. 133, pp. 61–67, 2009.
- [74]. J. Salonen, T. Hyvönen, H.A. Jalli, "Composition of weed flora in spring cereals in Finland—A fourth survey", *Agric. Food Sci.*, Vol. 20, 2011.
- [75]. P. A. Werner, R. Rioux, "The Biology of Canadian Weeds. 24. Agropyron Repens (L.) Beauv", *Canadian Journal of Plant Science*, Vol. 57, pp. 905-919.
- [76]. K.M. Ibrahim, P.M. Peterson, *Grasses of Washington*, D.C., Published by Smithsonian Institution Scholarly Press, Washington D.C., 2014.
- [77]. X. Yao, Y. Liu, G. Lin, "Evolutionary Programming Made Faster", *IEEE Transactions on Evolutionary Computation*, Vol. 3, No. 2, pp. 82-102, 1999.
- [78]. E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, "GSA: A Gravitational Search Algorithm", *Information Sciences*, Vol. 179, pp. 2232–2248, 2009.
- [79]. X. Yang, "Firefly algorithms for multimodal optimization", *International conference on stochastic algorithms foundations and applications*, pp.169–178, 2009.
- [80]. Y. Li, Y. Zhao, Y. Shang, J. Liu "An improved firefly algorithm with dynamic self-adaptive adjustment", *PLoS ONE*, Vol. 16, 2021.
- [81]. D. Wang, D. Tan, L. Liu, "Particle swarm optimization algorithm: an overview", *Soft Comput.*, Vol. 22, pp. 387–408, 2018.