Weighted Distance Grey Wolf Optimization with Immigration Operation for Global Optimization Problems

Duangjai Jitkongchuen, Warattha Sukpongthai
College of Innovative Technology and Engineering
Dhurakij Pundit University
Bangkok, Thailand
{duangjai.jit,warattha.suk}@dpu.ac.th

Arit Thammano
Faculty of Information Technology
King Mongkut's Institute of Technology Ladkrabang
Bangkok, Thailand
arit@it.kmitl.ac.th

Abstract—**The proposed algorithm presents a solution to improve the grey wolf optimizer performance using weighted distance and immigration operation. The weight distance is used for the omega wolves movement is defined from fitness value of each leader (alpha, beta and delta). The traditional grey wolf algorithm has only one pack and has opportunity to trap in local optimum so the wolves in our proposed algorithm have more pack and have migrated between them. When the amount of pack has more than to predefine some pack will be eliminated. The experimental results are evaluated by a comparative with the traditional grey wolf optimizer (GWO) algorithm, particle swarm optimization (PSO) and differential evolution (DE) algorithm on 9 well-known benchmark functions. The experimental results showed that the proposed algorithm is capable of efficiently to solving complex optimization problems.**

Keywords— Metaheuristic algorithm; Grey wolf optimizer algorithm; Weighted distance; Immigration operation;

I. INTRODUCTION

Improvements of computer hardware through the decade help scientists and researchers experiment and proof their hypothesis more easily and more quickly than earlier. Many problems, which required advanced mathematics and science knowledge for solving, can be implemented into a model and find its solution in a short time. From all feasible solutions, finding the best solution is necessary and challenging. This is a definition of optimization problem.

Swarm Intelligence (SI) is a field of computer science which gains a huge interesting from researchers for the past few decades. It is a population-based metaheuristic that can help to solve an optimization problem. According to Bonabeau et al. [2], SI definition extended to "Include any attempt to design algorithms or distributed problem-solving devices inspired by the collective behavior of social insect colonies and other animal societies".

There are many well-known nature-inspired metaheuristic algorithms. For example, Ant Colony Optimization (ACO) [1] that uses artificial ant colony for solving the travelling salesman problem (TSP) by collecting the information of pheromone trail. Many application applied ACO to solve their problem such as Green vehicle traffic routing system [10], which is not only reduce traveling time and improve traffic flow but also reduce fuel consumption and CO2 emissions by using Ant-based Vehicle Congestion Avoidance System (AVCAS) together with the Signalized Intersection Design and Research Aid (SIDRA) fuel consumption and emission model. Another one is an ACO for RFID reader in theme parks [3]. The RFID tracking system help people to find lost visitors location by using an Ant Colony System (ACS)-based reader and the ACS with incremental location setting (ACS-ILS) algorithm to minimize frequency collision of readers and find the best locations for install them. Bat Algorithm (BA) [4] is inspired by echolocation of microbats to probe their environment, find their home, and detect their prey in the darkness. A usage of BA in an oil wells location optimization [19] gives better result than particle swarm optimization and genetic algorithm. Cuckoo Optimization Algorithm (COA) [5] is inspired by their behaviors about egg laying, breeding, and reproduction. Cuckoo is a brood parasite; they remove an egg from the host nest, lay their mimic eggs instead and rely on the host to raise their baby. Many researches make use of COA in various fields, for instance, solving graph coloring problem by using a discrete COA [8], and pressure tracking in gas distribution network [9] that use COA for training an Artificial Neural Network. Firefly Algorithm (FA) [6] is a stochastic algorithm, which inspired by the bioluminescence of fireflies when they want to mate or to warn their predators. The light color can be green, red, or yellow and the flashing rhythm can vary. Firefly algorithm with neighborhood attraction (NaFA) [20] is one of an adaptive FA method, which is more efficient than standard FA and other adaptive FA in various ways such as the reducing of computational time. Particle Swarm Optimization (PSO) [14], another well-known algorithm of SI, inspired by bird flocking or fish schooling. Each particle has its own best solution (velocity and distance value from the centroid), which will be iteratively updated to improve its solution during movement around the search space. The result leads to the best value of global solution. An example of applied PSO is the accelerated particle swarm optimization (APSO) technique [12], which used to find a better battery capacity and charging time of plug-in hybrid electrical vehicles for better power management. Ant Lion Optimizer (ALO) [11] imitates five steps of hunting prey of antlions (or doodlebugs). The ALO algorithm helps in finding an optimal location and sizing of renewable distributed generations [18] and finding short-term wind integrated hydrothermal power generation scheduling [15]. Lion Optimization Algorithm (LOA) [17] was inspired by their social organization, nomads and residents. Resident is a group of lion. A male lion in mature state will be drove out from its resident group and become a nomad. Their behaviors, such as prey hunting and mating, were also used in the work.

The new metaheuristic has been focused on this paper called the grey wolf optimizer. The grey wolf optimizer algorithm (GWO) was proposed in [7]. The concept of GWO algorithm is to simulate the grey wolf behavior to live in a pack. Our algorithm proposed a weight distance method for grey wolf optimizer to updating location vector of the pack. And integrate the immigration operation to improve the exploration and exploitation.

The rest of the paper is organized as follows: the next section explains some backgrounds about the original grey wolf optimizer algorithm. The proposed algorithm is presented in Section 3. The computational results from selected benchmark functions is shown in Section 4. Finally, the conclusions will be discussed in Section 5.

II. GREY WOLF OPTIMIZER ALGORITHM

Grey Wolf Optimizer (GWO) was proposed in [7] uses the leadership hierarchy and hunting mechanism to create an algorithm for solving problems with unknown search spaces. Examples include Economic load dispatch problems [16], Wide-area power system stabilizer design [13], and Optimum spectrum mask based medical image fusion [21].

The algorithm is to simulate the grey wolf behavior to live in a pack. They have a serious social dominant hierarchy. The top level is the leaders, called alpha. The alpha is responsible for making decisions in the pack. The second level is the subordinate wolves, called beta. The operation of beta is to help the alpha in decision making or other activities. The next level is the subordinate wolves, called delta. The members in this category consist of scouts, sentinels, elders, hunters and caretakers. Scouts are responsible for observation the boundaries of region and warning the pack in case of any danger. Sentinels protect and guarantee the safety of the pack. Elders are the expertise wolves who used to be alpha or beta. Hunters help the alphas and betas when hunting prey and providing food for the pack and the caretakers are responsible for caring for the weak, ill, and wounded wolves in the pack. Omega is the lowest level. The omega wolves have to comply with all the other dominant wolves. In some cases the omega is also the babysitters in the pack.

An ability of Grey wolves has to recognize the positions of prey and to encircle them. The alpha performs the leader in the hunt. To simulate the hunting behavior of the grey wolves to the mathematically model, the best solution is assumed to be alpha (α). The beta (β) and delta (δ) is similar to the second and the third optimal solutions, respectively. The omega (ω) is the rest of the candidate solutions. The alpha, beta and delta are the leader to guide the hunting while the omega wolves should update their positions by considering the positions of these three best solutions.

Grey wolves encircle prey during the hunt. The mathematically of model encircling behavior as follows

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D} \tag{1}$$

$$\vec{D} = \left| \vec{C} \cdot \vec{X}_p(t) - \vec{X}(t) \right| \tag{2}$$

where t is the iteration number, $\vec{X}$ is the grey wolf position and $\vec{X}_p$ is the prey position. The vectors $\vec{A}$ and $\vec{C}$ that are coefficients are calculated as follows

$$\vec{A} = 2a \cdot \vec{r}_1 - a \tag{3}$$

$$C = 2\vec{r}_2 \tag{4}$$

The value of a is decreased linearly from 2 to 0 over the course of iterations and $\vec{r}_1$, $\vec{r}_2$ are random vectors in the range [0, 1]. The vector $\vec{C}$ is used to furnish a random weight to mention attractiveness of prey is a random value in the range [0, 2].

To mathematically simulate the hunting behavior of grey wolves, the alpha, beta and delta are assumed to have better knowledge about the possible location of prey. The first three best solutions get so far and force the other search agents (including the omegas) to update their positions according to the position of the three best search agents. The wolves' positions are updated as follow

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \tag{5}$$

where $\vec{X}_1$, $\vec{X}_2$, $\vec{X}_3$ are determined as in Eq. (6)-(8), respectively.

$$\vec{X}_1 = \left| \vec{X}_\alpha - \vec{A}_1 \cdot \vec{D}_\alpha \right| \tag{6}$$

$$\vec{X}_2 = \left| \vec{X}_\beta - \vec{A}_2 \cdot \vec{D}_\beta \right| \tag{7}$$

$$\vec{X}_3 = \left| \vec{X}_\delta - \vec{A}_3 \cdot \vec{D}_\delta \right| \tag{8}$$

where $\vec{X}_\alpha$, $\vec{X}_\beta$, $\vec{X}_\delta$ are the first three best solutions at a given iteration t, $\vec{A}_1$, $\vec{A}_2$, $\vec{A}_3$ are determined as in Eq. (3), and $\vec{D}_\alpha$, $\vec{D}_\beta$, $\vec{D}_\delta$ are determined as in Eq. (9)-(11), respectively.

$$\vec{D}_\alpha = \left| \vec{C}_1 \cdot \vec{X}_\alpha - \vec{X} \right| \tag{9}$$

$$\vec{D}_\beta = \left| \vec{C}_2 \cdot \vec{X}_\beta - \vec{X} \right| \tag{10}$$

$$\vec{D}_\delta = \left| \vec{C}_3 \cdot \vec{X}_\delta - \vec{X} \right| \tag{11}$$

where $\vec{C}_1$, $\vec{C}_2$, $\vec{C}_3$ are determined as in Eq. (4). The parameter a that controls the tradeoff between exploration and exploitation is linearly updated to range from 2 to 0 in each iteration as shown in Eq. (12)

$$a = 2 - t\frac{2}{MaxIter} \tag{12}$$

where t is the iteration number and MaxIter is the total number of iteration.

III. THE PROPOSED ALGORITHM

The process of traditional grey wolf algorithm has only one pack and has opportunity to trap in local optimum. The proposed algorithm presents grey wolf algorithm has more than one pack and used immigration operation between pack to reduce local optimum ratio and distribute value to expand the search space. Then the omega wolves in each pack used different weight value to move for convergence rate increasing.

The omega wolves in traditional grey wolf algorithm adjust its position by verified from the position of three leaders, alpha, beta and delta wolf. The omega wolves in proposed algorithm used different weight value of three leaders to adjust its position. The weight value is defined from fitness value of each leader. The alpha wolf that is the best solution has the best weight value and the beta and delta are the second and the third weight value, respectively. The omega wolves positions are updated as follow

$$\vec{X}(t+1) = \frac{W_1\vec{X}_1 + W_2\vec{X}_2 + W_3\vec{X}_3}{3} \tag{13}$$

where W_1 is the weight value of alpha wolf. W_2 is the weight value of beta wolf and W_3 is the weight value of delta wolf. The weight value of each leader are calculated as follows

$$W_1 = \frac{f_\alpha}{f_\alpha + f_\beta + f_\delta} \tag{14}$$

$$W_2 = \frac{f_\beta}{f_\alpha + f_\beta + f_\delta} \tag{15}$$

$$W_3 = \frac{f_\delta}{f_\alpha + f_\beta + f_\delta} \tag{16}$$

where f_α, f_β, f_δ is the fitness value of alpha, beta and delta wolf, respectively.

After that, the immigration operation is used to distribute value to expand the search space. The wolves from each pack that have a few fitness values have emigrated to combine in new pack. The old pack that lose wolves member will generate new wolves member to fill up. The new wolves members are generated by considering the center value of the pack in order to maintain the boundary of the pack.

The amount of pack is continually increasing in each iteration. When the amount of pack has more than to predefine some pack will be eliminated. The procedure for selecting the eliminative pack starts with each pack are calculated amount of wolves that have fitness value less than the average fitness value of pack. The pack that have maximum infirm member is eliminated. The eliminative operation is determined as in Eq. (3)

$$P_e = P,\ max(F^P) \tag{17}$$

$$F^P = count(X_i^P),\ f(X_i^P) < \frac{1}{NP}\sum_{k=1}^{NP} f(X_k^P) \tag{18}$$

where P_e is eliminative pack. P is pack. F^P is amount of wolves that have fitness value less than the average fitness value of pack. $f(X_i^P)$ is the fitness value associated to the individual and NP is population size.

Finally, the alpha, beta and delta positions are updated to reduce local optimum ratio by random new value. The overall and details of weighted distance grey wolf optimization with immigration operation procedure are shown below in Fig. 1.

```
Initialize the grey wolf population X_i (i = 1, 2, ..., n)
Initialize a, A and C
Calculate the fitness of each search agent
X_α = the best search agent
X_β = the second best search agent
X_δ = the third best search agent
while (t < Max number of iterations)
    for (X_i in each pack)
        Calculate weight value by Eq (14)-(16)
        Update current wolves position using weight value by Eq (13)
        Generate new position for leader wolves by random value
        Update a, A and C
        Calculate the fitness of all search agents
        Update X_α, X_β, X_δ
    end for
    for (X_i in each pack)
        Select i individuals that have less fitness than average fitness of pack
        Move i selected individuals into a new pack
        Fill up member in old pack using the center value of the pack
    end for
    if (amount of pack more than to predefine)
        Calculated amount of wolves that have fitness value less than average fitness value of pack
        Remove pack that have maximum infirm member
    end if
    t = t - 1
end while
```

Fig. 1. Pseudo code of Weighted Distance Grey Wolf Optimization with Immigration Operation

IV. EXPERIMENTAL RESULTS

The proposed algorithm have been evaluated performance by using nine commonly-used benchmark functions [22]. Table I are shown the details of benchmark functions where S is the boundary of search space. The function f_1 is Sphere function. The functions f_2 to f_4 is Schwefel's Problem 2.22, 1.2 and 2.21, respectively. The function f_5 is Generalized Rosenbrock function. The function f_6 is Step function. The function f_7 is Quartic function. The function f_8 is Generalized Schwefel's Problem 2.26 function and the function f_9 is Generalized Rastrigin function. The functions f_1 to f_7 are continuous unimodal functions and the functions f_8 and f_9 are multimodal functions. The control parameters used in the experiments are shown in Table II. The experiments are controlled in each test function, each algorithm is run 10 times with different initial populations each time and the control parameters are set to the same values.

The performances of the proposed algorithm have been evaluated with the grey wolf optimizer (GWO) algorithm, particle swarm optimization (PSO) algorithm and differential evolution (DE) algorithm. The experimental results are shown in Table III and the results of GWO, PSO and DE were taken from the results reported in [7]. The italic figures indicate the best results among the four algorithms. The experimental results showed that the proposed algorithm achieved the minimum of zero for the functions f_1, f_2, f_3 and f_9. Both the proposed algorithm and DE were prosperous to find the optimum solution for the functions f_4 and f_6. While only the DE could solve the optimum in the function f_5. For the functions f_7 and f_8, the results gained from the proposed algorithm are still the best among those of the compared algorithm.

The proposed algorithm presents two new concepts: weight distance and the immigration operation. The two concepts are independently applied to the traditional GWO algorithm. The weight distance significantly improves the performance of convergence ratio and the immigration operation helps to escape GWO from the local minima. The better results gained from the proposed algorithm confirmed that both concepts complement each other when used together.

V. CONCLUSIONS

The proposed algorithm presents a solution to improve the grey wolf optimizer performance using weight distance and immigration operation. The weight distance is used for the omega wolves movement is defined from fitness value of each leader. In addition, the wolves in our proposed algorithm have more pack and have migrated between them. The immigration operation is used to reduce local optimum ratio and distribute value to expand the search space. The proposed algorithm was evaluated performance with GWO, PSO and DE algorithms. The results showed that the proposed algorithm is capable of efficiently to solving optimization problems.

TABLE I. THE BENCHMARK FUNCTIONS

Functions	S
$f_1(x)=\sum_{i=1}^{N} x_i^2$	[-100, 100]
$f_2(x)=\sum_{i=1}^{N} \lvert x_i\rvert+\prod_{i=1}^{N}\lvert x_i\rvert$	[-10, 10]
$f_3(x)=\sum_{i=1}^{N}(\sum_{j=1}^{i} x_j)^2$	[-100, 100]
$f_4(x)=max_i\{\lvert x_i\rvert, 1\le i\le N\}$	[-100, 100]
$f_5(x)=\sum_{i=1}^{N-1}\left[100(x_{i+1}-x_i^2)^2+(x_i-1)^2\right]$	[-30, 30]
$f_6(x)=\sum_{i=1}^{N}(\lfloor x_i+0.5\rfloor)^2$	[-100, 100]
$f_7(x)=\sum_{i=1}^{N} ix_i^4+random\,[0,1)$	[-1.28, 1.28]
$f_8(x)=\sum_{i=1}^{N}-x_i\, sin(\sqrt{\lvert x_i\rvert})$	[-500, 500]
$f_9(x)=\sum_{i=1}^{N}\left[x_i^2-10cos(2\pi x_i)+10\right]$	[-5.12, 5.12]

TABLE II. THE CONTROL PARAMETERS USED IN THE EXPERIMENTS

Parameters	Values
Population size (NP)	100
Maximum pack	5
Maximum number of iterations	
Function f_1, f_6, f_{10}	1500
Function f_2, f_{11}	2000
Function f_7	3000
Function f_3, f_4, f_9	5000
Function f_8	9000
Function f_5	20000

TABLE III. THE EXPERIMENTAL RESULTS

Functions	Proposed	GWO	PSO	DE
f_1	*0*	6.59E-28	1.36E-03	8.20E-14
f_2	*0*	7.18E-17	4.21E-02	1.50E-09
f_3	*0*	3.29E-06	70.1256	6.80E-11
f_4	*0*	5.61E-07	1.0865	*0*
f_5	23.80734	26.81258	96.7183	*0*
f_6	*0*	0.816579	1.02E-04	*0*
f_7	*1.03E-06*	0.002213	1.23E-01	4.63E-03
f_8	*-11,876.1*	-6123.1	-4841.29	-11,080.1
f_9	*0*	0.310521	46.7042	69.20

REFERENCES

[1] M. Dorigo, L.M. Gambardella, "Ant colonies for the travelling salesman problem," Biosystems, vol.43, issue 2, 1997, p.73-81.

[2] E. Bonabeau, M. Dorigo and G. Theraulaz, "Swarm intelligence: from natural to artificial systems," OUP USA; 1999.

[3] Chieh-Yuan Tsai, Hui-Ting Chang, and Ren Jieh Kuo, "An ant colony based optimization for RFID reader deployment in theme parks under service level consideration," Tourism Management, vol.58, Feb 2017, p.1-14.

[4] X.S. Yang, "A New Metaheuristic Bat-Inspired Algorithm," Nature Inspired Cooperative Strategies for Optimization (NISCO 2010), Studies in Computational Intelligence, Springer Berlin, 284, Springer, p.65–74.

[5] R. Rajabioun, "Cuckoo Optimization Algorithm," Applied Soft Computing, vol.11, issue 8, December 2011, p.5508-5518.

[6] I. Fister, I. Fister Jr., X.S. Yang and J. Brest, "A comprehensive review of firefly algorithms," Information Sciences, Swarm and Evolutionary Computation, vol.13, December 2013, p.34-46.

[7] S. Mirjalili, S.M. Mirjalili and A. Lewis, "Grey Wolf Optimizer," Advances in Engineering Software, vol.69, 2014, p.46-61.

[8] S. Mahmoudi and S. Lotfi, "Modified cuckoo optimization algorithm (MCOA) to solve graph coloring problem," Applied Soft Computing, vol.33, August 2015, p.48-64.

[9] M. Hadian, N. AliAkbari and M. Karami, "Using artificial neural network predictive controller optimized with Cuckoo Algorithm for pressure tracking in gas distribution," Journal of Natural Gas Science and Engineering, vol.27, part 3, November 2015, p.1446-1454.

[10] M.R. Jabbarpour, R. Md Noor and R.H. Khokhar, "Green vehicle traffic routing system using ant-based algorithm," Journal of Network and Computer Applications, vol.58, December 2015, p.294-308.

[11] S. Mirjalili, "The Ant Lion Optimizer," Advances in Engineering Software, vol.83, 2015, p.80-98.

[12] I. Rahmana, P.M. Vasanta, B.S. Mahinder Singha and M. Abdullah-Al-Wadudb, "On the performance of accelerated particle swarm optimization for charging plug-in hybrid electric vehicles," Alexandria Engineering Journal, vol.55, issue 1, March 2016, p.419–426.

[13] M.R. Shakarami and I.F. Davoudkhani, "Wide-area power system stabilizer design," Electric Power Systems Research, vol.133, April 2016, p.149-159.

[14] X.S. Yang, S. Deb, S. Fong, X. He and Y.X. Zhao, "From Swarm Intelligence to Metaheuristics: Nature-Inspired Optimization Algorithms," Computer, vol.49, issue 9, September 2016, p.52-59.

[15] H.M. Dubeya, M. Pandita and B.K. Panigrahib, "Ant lion optimization for short-term wind integrated hydrothermal power generation scheduling," International Journal of Electrical Power & Energy Systems, vol.83, December 2016, p.158-174.

[16] M. Pradhan, P.K. Roy and T. Pal, "Grey wolf optimization applied to economic load dispatch problems," International Journal of Electrical Power & Energy Systems, vol.83, December 2016, p.325-334.

[17] M. Yazdani and F. Jolai, "Lion Optimization Algorithm (LOA): A nature-inspired metaheuristic algorithm," Journal of Computational Design and Engineering 3, 2016, p.24-36.

[18] E.S. Ali, S.M. Abd Elazim and A.Y. Abdelaziz, "Ant Lion Optimization Algorithm for optimal location and sizing of renewable distributed generations," Renewable Energy, vol.101, February 2017, p.1311-1324.

[19] M. Naderi and E. Khamehchi, "Well placement optimization using metaheuristic bat algorithm," Journal of Petroleum Science and Engineering, vol.150, February 2017, p.348–354.

[20] H. Wang, W. Wang, X. Zhou, H. Sun, J. Zhao, X. Yu and Z. Cui, "Firefly algorithm with neighborhood attraction," Information Sciences, vol.382–383, March 2017, p.374-387.

[21] E. Daniel, J. Anitha, K.K. Kamaleshwaran and I. Rani, "Optimum spectrum mask based medical image fusion using Gray Wolf Optimization," Biomedical Signal Processing and Control, vol.34, April 2017, p.36-43.

[22] X. Yao, Y. Liu and L. Gao, "Evolutionary programming made faster," IEEE Transactions on Evolutionary Computation, vol. 3, pp. 82-102, 1999.