# Grey Wolf Optimization

[1] Sarath Kumar Reddy, [2] Yaswanth Reddy, Damodar Reddy,Monish Kumar Reddy [1]
[1] Karunya Institute of Technology and Sciences, Coimbatore, India
[2] Department of Computer Science and Engineering, Faculty of Computer Engineering, Karunya University

***Abstract*:** A new meta-heuristic work named Grey Wolf Optimizer (GWO) inspired by grey wolves (Canis lupus) is suggested here. The GWO algorithm mimics the structure of leadership and the system for killing grey wolves in nature. Four forms of grey wolves are used to represent the structure of leadership, such as alpha , beta , delta and omega. Moreover, the three main phases of hunting, in search of The prey, the prey encircling, and the prey attacking are executed. The algorithm is then benchmarked on 29 well-known test functions and the results are checked through a comparative analysis with Particle Swarm Optimization ( PSO), GSA, and Evolution Strategy (ES). The results show that the GWO algorithm is able to produce very competitive results in comparison with these well-known meta-heuristics. The paper also aims to solve three classic engineering problems and presents a real application of the proposed method in the field of optical engineeriing. The results of the classical engineering design problems and the actual implementation show that the proposed algorithm is applicable to the task of uncertain search spaces.

Keywords: optimisation, methods of optimisation, heuristic algorithm, metaheuristics, constricted optimization, GWO

## 1. Introduction

Meta-heuristic optimization techniques became very fashionable over the last 20 years . Surprisingly, variety of them like Genetic Algorithm (GA) [1], Ant Colony Optimization (ACO) [2], and Particle Swarm Optimization (PSO) [3] are fairly well-known among not only computer scientists but also scientists from different fields. additionally to the massive number of theoretical works, such optimization techniques are applied in various fields of study. there's a problem here on why meta-heuristics became remarkably common. the answer to this question are often summarized into four main reasons: simplicity, flexibility, derivation-free mechanism, and native optima avoidance.

First, meta-heuristics are fairly simple. they have been mostly inspired by very simple concepts. The inspirations are typically related to physical phenomena, animals' behaviour's, or evolutionary concepts. The simplicity allows computer scientists to simulate different natural concepts, propose new meta-heuristics, hybridize two or more meta-heuristics, or improve this meta-heuristics. Moreover, the simplicity assists other scientists to seek out out meta-heuristics quickly and apply them to their problems.

Second, flexibility refers to the applicability of meta-heuristics to different problems with none special changes within the structure of the algorithm. Meta-heuristics are readily applicable to different problems since they mostly assume problems as black boxes. In other words, only the input(s) and output(s) of a system are important for a meta-heuristic. So, all a designer needs is to know the thanks to represent his/her problem for meta- heuristics.

Third, the majority of meta-heuristics have derivation-free mechanisms. In contrast to gradient-based optimization approaches, meta-heuristics optimize problems stochastically. The optimization process starts with random solution(s), and there is no need to calculate the derivative of search spaces to hunt out the optimum. This makes meta-heuristics highly suitable for real problems with expensive or unknown derivative information.

Finally, meta-heuristics have superior abilities to avoid local optima compared to plain optimization techniques. this is often often because of the stochastic nature of meta-heuristics which enable them to avoid stagnation in local solutions and search the entire search space extensively. The search space of real problems is usually unknown and really complex with an enormous number of local optima, so meta-heuristics are good options for optimizing these challenging real problems.

The No gift (NFL) theorem [4] is worth mentioning here. This theorem has logically proved that there is no meta-heuristic best fitted to solving all optimization problems. In other words, a selected meta- heuristic may show very promising results on a gaggle of problems, but the same algorithm may show poor performance on a special set of problems. Obviously, NFL makes this field of study highly active which finishes up in enhancing current approaches and proposing new meta-heuristics once a year . This also motivates our attempts to develop a replacement meta-heuristic inspirationally from grey wolves.

Generally speaking, meta-heuristics are often divided into two main classes: single-solution-based and population-based. within the previous class (Simulated Annealing [5] for instance) the search process starts with one candidate solution. This single candidate solution is then improved over the course of iterations. Population- based meta-heuristics, however, perform the optimization employing a group of solutions (population). during this case the search process starts with a random initial population (multiple solutions), and this population is enhanced over the course of iterations. Population-based meta-heuristics have some advantages compared to single solution- based algorithms:

• Multiple candidate solutions share information about the search space which finishes up in sudden jumps toward the promising a neighborhood of search space
• Multiple candidate solutions assist each other to avoid locally optimal solutions
• Population-based meta-heuristics generally have greater exploration compared to single solution-based algorithms

One of the interesting branches of the population-based meta-heuristics is Swarm Intelligence (SI). The concepts of SI was first proposed in 1993 [6]. according to Bonabeau et al. [1], SI is “The emergent collective intelligence of groups of straightforward agents”. The inspirations of SI techniques originate mostly from natural colonies, flock, herds, and schools. variety of the foremost popular SI techniques are ACO [2], PSO [3], and Artificial Bee Colony (ABC) [7]. A comprehensive literature review of the SI algorithms is provided within subsequent section. variety of the advantages of SI algorithms are:

• SI algorithms preserve information about the search space over the course of iteration, whereas Evolutionary Algorithms (EA) discard the knowledge of the previous generations
• SI algorithms often utilize memory many |to avoid wasting"> to save lots of lots of the only solution obtained so far
• SI algorithms usually have fewer parameters to manage
• SI algorithms have less operators compared to evolutionary approaches (crossover, mutation, elitism, then on)
• SI algorithms are easy to implement

Regardless of the differences between the meta-heuristics, a typical feature is that the division of the search process into two phases: exploration and exploitation [8-12]. The exploration phase refers to the tactic of investigating the promising area(s) of the search space as broadly as possible. An algorithm must have stochastic operators to randomly and globally search the search space so on support this phase. However, exploitation refers to the local search capability around the promising regions obtained within the exploration phase. Finding an accurate balance between these two phases is taken under consideration a challenging task because of the stochastic nature of meta-heuristics. This work proposes a replacement SI technique inspirationally from the social hierarchy and hunting behavior of grey wolf packs. the rest of the paper is organized as follows:
Section 2 presents a literature review of SI techniques. Section 3 outlines the proposed GWO algorithm. The results and discussion of benchmark functions, semi-real problems, and a real application are presented in Section 4, Section 5, and Section 6, respectively. Finally, Section 7 concludes the work and suggests some directions for future studies.

## 2. Literature review

Meta-heuristics could also be classified into three main classes: evolutionary, physics-based, and SI algorithms. EAs are usually inspired by the concepts of evolution in nature. the foremost popular algorithm during this branch is GA. This algorithm was proposed by Holland in 1992 [13] and simulates Darwnian evolution concepts. The engineering applications of GA were extensively investigated by Goldberg [14]. Generally speaking, the optimization is completed by evolving an initial random solution in EAs. Each new population is made by the mixture and mutation of the individuals within the previous generation. Since the simplest individuals have higher probability of participating in generating the new population, the new population is probably going to be better than the previous generation(s). this will guarantee that the initial random population is optimized over the course of generations. a number of the EAs are Differential Evolution (DE) [15], Evolutionary Programing (EP) [16, 17], and

As an example, the BBO algorithm was first proposed by Simon in 2008 [21]. the essential idea of this algorithm has been inspired by biogeography which refers to the study of biological organisms in terms of geographical distribution (over time and space). The case studies might include different islands, lands, or maybe continents over decades, centuries, or millennia. during this field of study different ecosystems (habitats or territories) are investigated for locating the relations between different species (habitants) in terms of immigration, emigration, and mutation. The evolution of ecosystems (considering different sorts of species like predator and prey) over migration and mutation to succeed in a stable situation was the most inspiration of the BBO algorithm.

For example, the BBBC algorithm was inspired by the large bang and large crunch theories. The search agents of BBBC are scattered from some extent in random directions during a search space consistent with the principles of the large bang theory. They search randomly then take in a final point (the best point obtained so far) consistent with the principles of the large crunch theory. GSA is another physics-based algorithm. the essential physical theory from which GSA is inspired is Newton’s law of universal gravitation. The GSA algorithm performs search by employing a set of agents that have masses proportional to the worth of a fitness function. During iteration, the masses are interested in one another by the gravitational forces between them. The heavier the mass, the larger the attraction . Therefore, the heaviest mass, which is possibly on the brink of the worldwide optimum, attracts the opposite masses in proportion to their distanc

The third subclass of meta-heuristics is that the SI methods. These algorithms mostly mimic the social behavior of swarms, herds, flocks, or schools of creatures in nature. The mechanism is nearly almost like physics-based algorithm, but the search agents navigate using the simulated collective and social intelligence of creatures. the foremost popular SI technique is PSO. The PSO algorithm was proposed by Kennedy and Eberhart [3] and inspired from the social behavior of birds flocking. The PSO algorithm employs multiple particles that chase the position of the simplest particle and their own best positions obtained thus far . In other words, a particle is moved considering its own best solution also because the best solution the swarm has obtained.

Another popular SI algorithm is ACO, proposed by Dorigo et al. in 2006 [2]. This algorithm was inspired by the social behavior of ants in an ant colony. In fact, the social intelligence of ants find the shortest path between the nest and a source of food is that the main inspiration of ACO. A pheromone matrix is evolved over the course of iteration by the candidate solutions. The ABC is another popular algorithm, mimicking the collective behavior of

bees find food sources. There are three sorts of bees in ABS: scout, onlooker, and employed bees. The scout bees are liable for exploring the search space, whereas onlooker and employed bees exploit the promising solutions found by scout bees. Finally, the Bat-inspired Algorithm (BA), inspired by the echolocation behavior of bats, has been proposed recently [33]. There are many sorts of bats within the nature. they're different in terms of size and weight, but all of them have quite similar behaviors when navigating and hunting. Bats utilize natural sonar so as to try to to this. the 2 main characteristics of bats when finding prey are adopted in designing the BA algorithm. Bats tend to decrease the loudness and increase the speed of emitted ultrasonic sound once they chase prey. This behavior has been mathematically modeled for the BA algorithm. the remainder of the SI techniques proposed thus far are as follows:

- Marriage in Honey Bees Optimization Algorithm (MBO) in 2001 [34]
- Artificial Fish-Swarm Algorithm (AFSA) in 2003 [35]
- Termite Algorithm in 2005 [36]
- Wasp Swarm Algorithm in 2007 [37]
- Monkey Search in 2007 [38]
- Bee Collecting Pollen Algorithm (BCPA) in 2008 [39]
- Cuckoo Search (CS) in 2009 [40]
- Dolphin Partner Optimization (DPO) in 2009 [41]
- Firefly Algorithm (FA) in 2010 [42]
- Bird Mating Optimizer (BMO) in 2012 [43]
- Krill Herd (KH) in 2012 [44]
- Fruit fly Optimization Algorithm (FOA) in 2012 [45]

This list shows that there are many SI techniques proposed thus far , many of them inspired by hunting and search behaviors. To the simplest of our knowledge, however, there's no SI technique within the literature mimicking the leadership hierarchy of grey wolves, documented for his or her pack hunting. This motivated our plan to mathematically model the social behavior of grey wolves, propose a replacement SI algorithm inspired by grey wolves, and investigate its abilities in solving benchmark and real problems.

## 3. Grey Wolf Optimizer (GWO)

In this section the inspiration of the proposed method is first discussed. Then, the mathematical model is provided.

### 3.1. Inspiration

Grey wolf (Canis lupus) belongs to Canidae family. Grey wolves are considered as apex predators, meaning that they're at the highest of the organic phenomenon . Grey wolves mostly like better to sleep in a pack. The group size is 5-12 on the average . Of particular interest is that they need a really strict social dominant hierarchy as shown in Fig. 1.

Fig. 1. Hierarchy of grey wolf (dominance decreases from top down)

The leaders are a male and a female, called alphas. The alpha is usually liable for making decisions about hunting, sleeping place, time to wake, and so on. The alpha's decisions are dictated to the pack. However, some quite democratic behavior has also been observed, during which an alpha follows the opposite wolves within the pack. In gatherings, the whole pack acknowledges the alpha by holding their tails down. The alpha wolf is additionally called the dominant wolf since his/her orders should be followed by the pack [46]. The alpha wolves are only allowed to mate within the pack. Interestingly, the alpha isn't necessarily the strongest member of the pack but the simplest in terms of managing the pack. This shows that the organization and discipline of a pack is far more important than its strength.

The second level within the hierarchy of grey wolves is beta. The betas are subordinate wolves that help the alpha in decision-making or other pack activities. The beta wolf are often either male or female, and he/she is perhaps the simplest candidate to be the alpha just in case one among the alpha wolves passes away or becomes very old. The beta wolf should respect the alpha, but commands the opposite lower-level wolves also . It plays the role of an advisor to the alpha and discipliner for the pack. The beta reinforces the alpha's commands throughout the pack and provides feedback to the alpha.

The lowest ranking grey wolf is omega. The omega plays the role of scapegoat. Omega wolves always need to undergo all the opposite dominant wolves. they're the last wolves that are allowed to eat. it's going to seem the omega isn't a crucial individual within the pack, but it's been observed that the entire pack face internal fighting and problems just in case of losing the omega. this is often thanks to the venting of violence and frustration of all wolves by the omega(s). This assists satisfying the whole pack and maintaining the dominance structure. In some cases the omega is additionally the babysitters within the pack.

If a wolf isn't an alpha, beta, or omega, he/she is named subordinate (or delta in some references). Delta wolves need to undergo alphas and betas, but they dominate the omega. Scouts, sentinels, elders, hunters, and

caretakers belong to the present category. Scouts are liable for watching the boundaries of the territory and warning the pack just in case of any danger. Sentinels protect and guarantee the security of the pack. Elders are the experienced wolves who wont to be alpha or beta. Hunters help the alphas and betas when hunting prey and providing food for the pack. Finally, the caretakers are liable for caring for the weak, ill, and wounded wolves within the pack.

Fig. 2. *Hunting behaviour of grey wolves: (A) chasing, approaching, and tracking prey (B-D) pursuiting, harassing, and encircling (E) stationary situation and attack [47]*

*In this work this hunting technique and therefore the social hierarchy of grey wolves are mathematically modeled so as to style GWO and perform optimization.*

*3.2. Mathematical model and algorithm*
*In this subsection the mathematical models of the social hierarchy, tracking, encircling, and attacking prey are provided. Then the GWO algorithm is printed .*

*3.2.1. Social hierarchy:*
*In order to mathematically model the social hierarchy of wolves when designing GWO, we consider the fittest solution because the alpha ( ). Consequently, the second and third best solutions are named beta ( ) and delta ( ) respectively. the remainder of the candidate solutions are assumed to be omega ( ). within the GWO algorithm the hunting (optimization) is guided by , , and . The wolves follow these three wolves.*

*3.2.2. Encircling prey:*
*As mentioned above, grey wolves encircle prey during the hunt. so as to mathematically model encircling behavior the subsequent equations are proposed:*

$\vec{\ }\ \vec{\ }\ \vec{\ \ }$ *(3.1)*

$\vec{\ }\ \vec{\ \ }\ \vec{\ }$ *(3.2)*

*where t indicates the present iteration, and are coefficient vectors ,* $\vec{\ }\ \vec{\ \ }$ *is that the position vector of the prey, and indicates the position vector of a gray wolf.*
*The vectors and are calculated as follows:*

*= 2 .* $\vec{\ }$ *- (3.3)*

$\vec{\ }$ *(3.4)*

*where components of are linearly decreased from 2 to 0 over the course of iterations and , are random vectors in [0,1].*
*To see the consequences of equations (3.1) and (3.2), a two-dimensional position vector and a few of the possible neighbors are illustrated in Fig. 3 (a). As are often seen during this figure, a gray wolf within the position of (X,Y) can update its position consistent with the position of the prey (X*,Y*). Different places round the best agent are often*
*reached with reference to the present position by adjusting the worth of and vectors. as an example , (X*-X,Y*) are often reached by setting and therefore the possible updated positions of a gray wolf in 3D space*are depicted in Fig. 3 (b). Note that the random vectors and allow wolves to reach any position between the points illustrated in Fig. 3. So a grey wolf can update its position inside the space around the prey in any random location by using equations (3.1) and (3.2).

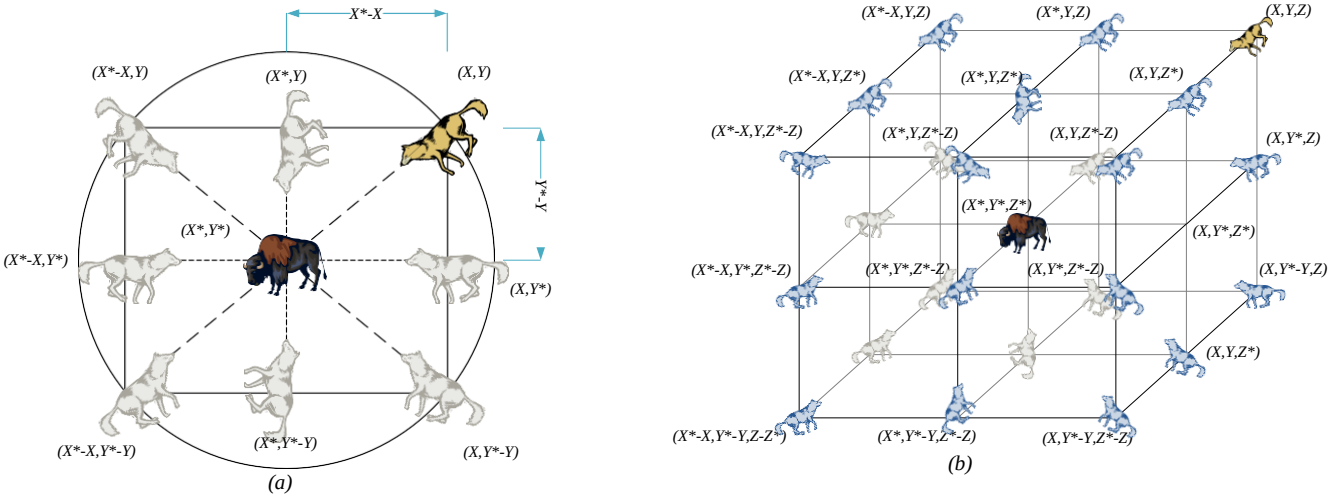


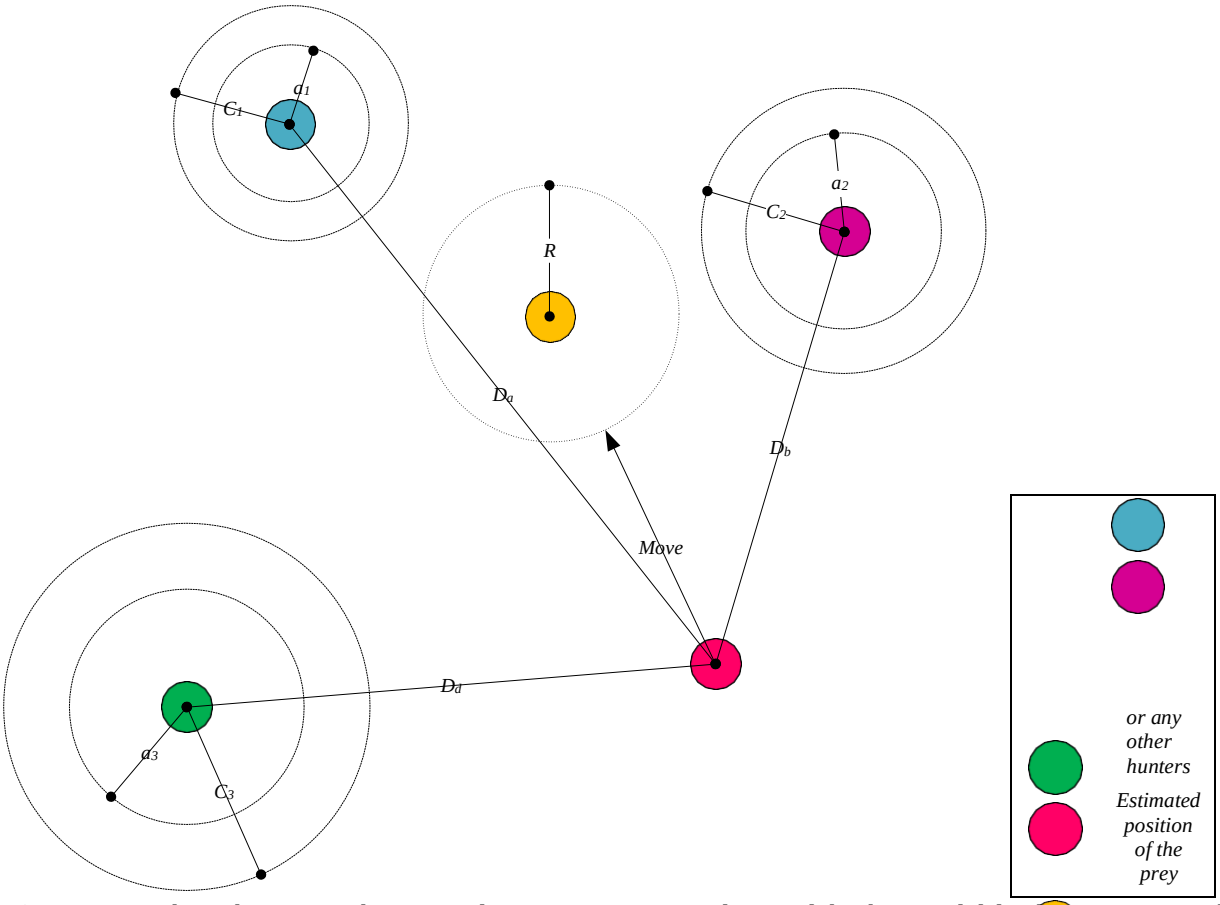

Fig. 3. *2D and 3D position vectors and their possible next locations*

The same concept can be extended to a search space with *n* dimensions, and the grey wolves will move in hyper-cubes (or hyper-spheres) around the best solution obtained so far.

3.2.1. Hunting:

3.2.2. Grey wolves have the power to acknowledge the situation of prey and encircle them. The hunt is typically guided by the alpha. The beta and delta may additionally participate in hunting occasionally. However, in an abstract search space we've no idea about the situation of the optimum (prey). so as to mathematically simulate the hunting behavior of grey wolves, we suppose that the alpha (best candidate solution) beta, and delta have better knowledge about the potential location of prey. Therefore, we save the primary three best solutions obtained thus far and oblige the opposite search agents (including the omegas) to update their positions consistent with the position of the simplest search agent. the subsequent formulas are proposed during this regard.

3.2.3.

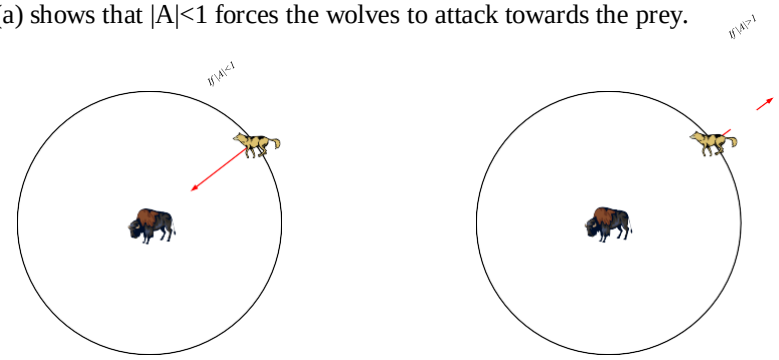



3.2.4. Fig. 4 shows how a search agent updates its position according to alpha, beta, and delta during a 2D search space. It are often observed that the last word position would be during a random place within a circle which is defined by the positions of alpha, beta, and delta within the search space. In other words alpha, beta, and delta estimate the position of the prey, and other wolves updates their positions randomly around the prey.

Fig. 4. *Position updading in GWO*

3.2.5. Attacking prey (exploitation):

3.2.6. As mentioned above the grey wolves finish the hunt by attacking the prey when it stops moving. so as to mathematically model approaching the prey we decrease the worth of . Note that the fluctuation range of is also decreased by . In other words may be a random value within the interval [-a,a] where a is decreased from 2 to 0 over the course of iterations. When random values of are in [-1,1], the next position of a search agent can be in any position between its current position and the position of the prey. Fig. 5 (a) shows that |A|<1 forces the wolves to attack towards the prey.



*(a)* *(b)*

Fig. 5. *Attacking prey versus searching for prey*

With the operators proposed thus far , the GWO algorithm allows its search agents to update their position supported the situation of the alpha, beta, and delta; and attack towards the prey. However, the GWO algorithm is susceptible to stagnation in local solutions with these operators. it's true that the encircling mechanism proposed shows exploration to some extent, but GWO needs more operators to stress exploration.

3.2.7. look for prey (exploration):

Grey wolves mostly search consistent with the position of the alpha, beta, and delta. They diverge from one another to look for prey and converge to attack prey. so as to mathematically model divergence, we utilize with random values greater than 1 or but -1 to oblige the search agent to diverge from the prey. This
emphasizes exploration and allows the GWO algorithm to look globally. Fig. 5(b) also shows that |A|>1 forces the grey wolves to diverge from the prey to hopefully find a fitter prey. Another component of GWO that favors exploration is . As could also be seen in Equation (3.4), the vector contains random values in [0, 2]. This component provides random weights for prey so as to stochastically emphasize (C>1) or deemphasize (C1 and converge towards the prey when |

```
Initialize the grey wolf population Xi (i = 1, 2, ..., n)
Initialize a, A, and C
Calculate the fitness of each search agent
Xα=the best search agent
Xβ=the second best search agent
Xδ=the third best search agent
while (t < Max number of iterations)
    for each search agent
            Update the position of the current search agent by equation (3.7)
    end for
    Update a, A, and C
    Calculate the fitness of all search agents
    Update Xα, Xβ, and Xδ
    t=t+1
end while
return Xα
```

|

Fig. 6. *Pseudo code of the GWO algorithm*

To see how GWO is theoretically able to solve optimization problems, some points may be noted:

- The proposed social hierarchy assists GWO to save the best solutions obtained so far over the course of iteration
- The proposed encircling mechanism defines a circle-shaped neighborhood around the solutions which can be extended to higher dimensions as a hyper-sphere
- The random parameters *A* and *C* assist candidate solutions to have hyper-spheres with different random radii

- The proposed hunting method allows candidate solutions to locate the probable position of the prey
- Exploration and exploitation are guaranteed by the adaptive values of $a$ and $A$
- The adaptive values of parameters $a$ and $A$ allow GWO to smoothly transition between exploration and exploitation
- With decreasing $A$, half of the iterations are devoted to exploration ($|A| \geq 1$) and the other half are dedicated to exploitation ($|A| < 1$)
- The GWO has only two main parameters to be adjusted ($a$ and $C$)

## 4. Results and discussion

In this section the GWO algorithm is benchmarked on 29 benchmark functions. the primary 23 benchmark functions are the classical functions utilized by many researchers [16, 48-51]. Despite the simplicity, we've chosen these test functions to be ready to compare our results to those of the present meta-heuristics. These benchmark functions are listed in Table 1, Table 2, and Table 3 where Dim indicates dimension of the function, Range is that the boundary of the function's search space, and fmin is that the optimum. the opposite test beds that we've chosen are six composite benchmark functions from a CEC 2005 session [52]. These benchmark functions are the shifted, rotated, expanded, and combined variants of the classical functions which supply the best complexity among the present benchmark functions [53]. Tables 4 lists the CEC 2005 test functions, where Dim indicates dimension of the function, Range is that the boundary of the function's search space, and fmin is that the optimum. Fig. 7, Fig. 8, Fig. 9, and Fig. 10 illustrate the 2D versions of the benchmark functions used.

Generally speaking, the benchmark functions used are minimization functions and may be divided into four groups: unimodal, multimodal, fixed-dimension multimodal, and composite functions. Note that an in depth descriptions of the composite benchmark functions are available within the CEC 2005 technical report [52].

## References


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

[2] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *Computational Intelligence Magazine, IEEE,* vol. 1, pp. 28-39, 2006.

[3] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, 1995, pp. 1942-1948.

[4] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *Evolutionary Computation, IEEE Transactions on,* vol. 1, pp. 67-82, 1997.

[5] S. Kirkpatrick, D. G. Jr., and M. P. Vecchi, "Optimization by simulated annealing," *science,* vol. 220, pp. 671-680, 1983.

[6] G. Beni and J. Wang, "Swarm intelligence in cellular robotic systems," in *Robots and Biological Systems: Towards a New Bionics?*, ed: Springer, 1993, pp. 703-712.

[7] B. Basturk and D. Karaboga, "An artificial bee colony (ABC) algorithm for numeric function optimization," in *IEEE swarm intelligence symposium*, 2006, pp. 12-14.

[8] O. Olorunda and A. P. Engelbrecht, "Measuring exploration/exploitation in particle swarms using swarm diversity," in *Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on*, 2008, pp. 1128-1134.

[9] E. Alba and B. Dorronsoro, "The exploration/exploitation tradeoff in dynamic cellular genetic algorithms," *Evolutionary Computation, IEEE Transactions on,* vol. 9, pp. 126-142, 2005.

[10] L. Lin and M. Gen, "Auto-tuning strategy for evolutionary algorithms: balancing between exploration and exploitation," *Soft Computing,* vol. 13, pp. 157-168, 2009.
S. Mirjalili and S. Z. M. Hashim, "A new

[11] T. Baba, "Slow light in photonic crystals," *Nature Photonics,* vol. 2, pp. 465-473, 2008.

[12] Y. Zhai, H. Tian, and Y. Ji, "Slow light property improvement and optical buffer capability in ring-shape-hole photonic crystal waveguide," *Lightwave Technology, Journal of,* vol. 29, pp. 3083-3090, 2011.

[13] D. Wang, J. Zhang, L. Yuan, J. Lei, S. Chen, J. Han, and S. Hou, "Slow light engineering in polyatomic photonic crystal waveguides based on square lattice," *Optics Communications,* vol. 284, pp. 5829-5832, 2011.

[14] S. M. Mirjalili and S. Mirjalili, "Light property and optical buffer performance enhancement using Particle Swarm Optimization in Oblique Ring-Shape-Hole Photonic Crystal Waveguide," in *Photonics Global Conference (PGC), 2012*, 2012, pp. 1-4.

[15] S. M. Mirjalili, K. Abedi, and S. Mirjalili, "Optical buffer performance enhancement using Particle Swarm Optimization in Ring-Shape-Hole Photonic Crystal Waveguide," *Optik - International Journal for Light and Electron Optics,* vol. 124, pp. 5989-5993, 2013.

[16] S. M. Mirjalili, S. Mirjalili, and A. Lewis, "A Novel Multi-objective Optimization Framework for Designing Photonic Crystal Waveguides," *Photonics Technology Letters, IEEE,* vol. 26, pp. 146-149, 2014.

[17] S. M. Mirjalili, S. Mirjalili, A. Lewis, and K. Abedi, "A tri-objective Particle Swarm Optimizer for designing line defect Photonic Crystal Waveguides," *Photonics and Nanostructures - Fundamentals and Applications*.

[18] J. Wu, Y. Li, C. Peng, and Z. Wang, "Wideband and low dispersion slow light in slotted photonic crystal waveguide," *Optics Communications,* vol. 283, pp. 2815-2819, 2010.