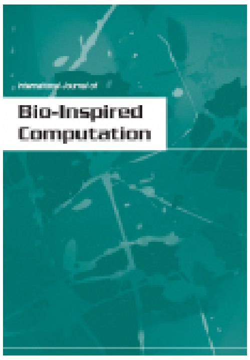


International Journal of Bio-Inspired Computation

ISSN online: 1758-0374 - ISSN print: 1758-0366
https://www.inderscience.com/ijbic

Alligator optimisation algorithm for solving unconstrained optimisation problems

Weng-Hooi Tan, Junita Mohamad-Saleh

DOI: 10.1504/IJBIC.2022.10049859

Article History:

Received:	29 November 2021
Accepted:	09 July 2022
Published online:	04 April 2023

Alligator optimisation algorithm for solving unconstrainted optimisation problems

Weng-Hooi Tan and Junita Mohamad-Saleh*

School of Electrical and Electronic Engineering,
Universiti Sains Malaysia,
Nibong Tebal, 11430, Penang, Malaysia
Email: tanwenghooi@hotmail.com
Email: jms@usm.my
*Corresponding author

Abstract: Inspired by cooperative hunting skills and movement patterns of alligators in nature, this research paper proposes a novel bio-inspired meta-heuristic algorithm, named alligator optimisation (AgtrO) algorithm. Upon mathematical modelling, AgtrO emphasises two main phases: the hunting phase that mimics fishing, purse seining and catching mechanisms, and the relocating phase that mimics travelling and homing instinct mechanisms. The hunting phase discovers any promising global optimal area, towards tracking the true global optimal solution. Meanwhile, the relocating phase avoids local optima (traps) through local exploration and conducts in-depth investigations through local exploitation. The proposed AgtrO was tested on 23 classical optimisation benchmark functions and ten modern CEC-C06-2019 benchmark functions, in comparison with eight recently proposed state-of-the-art algorithms. Upon evaluation, AgtrO has been proved to outperform other algorithms in terms of global-best achievement, while being very competitive in terms of convergence speed.

Keywords: bio-inspired; metaheuristic; optimisation; classical benchmark; CEC benchmark.

Reference to this paper should be made as follows: Tan, W-H. and Mohamad-Saleh, J. (2023) 'Alligator optimisation algorithm for solving unconstrainted optimisation problems', *Int. J. Bio-Inspired Computation*, Vol. 21, No. 1, pp.11–25.

Biographical notes: Weng-Hooi Tan received his BSc in Engineering and MSc in Research from the Department of Electrical and Electronic Engineering, Universiti Sains Malaysia (USM) in 2017 and 2020, respectively. He is currently pursuing his PhD at the School of Electrical and Electronic Engineering, Universiti Sains Malaysia (USM). His research interests are in the areas of distributed computing, soft computing system and applications.

Junita Mohamad-Saleh received her BSc in Computer Engineering from the Case Western Reserve University, USA in 1994, MSc degree from the University of Sheffield, UK in 1996 and PhD degree from the University of Leeds, UK in 2002. She is currently an Associate Professor in the School of Electrical and Electronic Engineering, Universiti Sains Malaysia. Her research interests include computational intelligence, tomographic and medical imaging and soft computing.

1 Introduction

There is an increasing demand for more sophisticated optimisation techniques and algorithms, as state-of-the-art technologies require more effective optimisation for robust application. Through in-depth research and survey, potential application areas of near-term optimisation techniques include optimised plans for system, optimal system operation, optimal design of equipment, optimised equipment diagnostics and decision making that consider uncertainty. The system optimisation planning enhances system functionality and integration of subsystem elements to keep all components operating at or above user expectations. Tremendous efforts have been put into optimal construction, optimal facility expansion, optimal material procurement, optimal equipment layout, economic efficiency, and quality assurance. Recently, the application of system optimisation planning has begun to focus on broader fields, such as academia (Deng and Lv, 2020), engineering (Elekidis et al., 2018; Leiber et al., 2022), economics (Elekidis et al., 2018; Triska et al., 2021), medicine (Kodama et al., 2021) and politics (Burdova and Tikhonova, 2021). As per definition, operational optimisation is the process of ensuring that operations are performed as efficiently and effectively as possible, typically to minimise load and maximise operational capacity. Optimal design of equipment can be simply defined as a class of equipment designs that are optimal based on some statistical criteria. It mainly benefits from optimal construction, minimum volume and uniform properties (Li and Deng, 2020; Agushaka and Ezugwu, 2021; Guo, 2021; Lee et al., 2021). Equipment diagnostics is a subfield of control engineering that focuses on monitoring systems, identifying when faults occur, and pinpointing the type of fault and its location. The optimisation technique establishes a fault classification model, shortens the training time, improves the accuracy of fault diagnosis, and enables the equipment to recover from the fault through the recovery of the faulty components. This application is widely used in the medical field for the early detection of diseases such as breast cancer (Bensaoucha, 2021; Dou and Meng, 2021) and lung tumours (Luo et al., 2021), as these diseases must be cured at an early stage. Automatic computer-aided detection is a great way to reduce human error and improve detection accuracy. A final potential application, decision making, is primarily to maximise average profit and minimise risk hedging. Decisions must be based on real options, profitability assessments and operational efficiency. It is mainly employed in the financial field to maximise the average profit of a product while considering risk and quality assurance (Qin et al., 2022), or any other field where risk hedging needs to be considered (Lainas et al., 2020; Li et al., 2021a, 2021b; Rana and Varshney, 2021).

To sum up, as technology develops, stronger optimisation techniques are required to obtain better solutions. This has prompted researchers to further deepen the improvement of various optimisation algorithms, pushing the technology to new heights. Optimisation is a process of searching for the most effective global minimum or maximum. Complex optimisation problems usually require more sophisticated resolution such as intelligent approach. It is often referred to as a class of population-based algorithms because it assigns a large number of search agents for execution according to programmatic regulations inspired by nature. These algorithms basically fall into three broad categories: evolutionary, physics-based, and bio-inspired.

An evolutionary algorithm is a probabilistic search method that simulates the process of natural selection based on biological evolution, such as reproduction, mutation, recombination, and selection, which enable the population to share and inherit the best-optimised information throughout generations (iterations). Some well-known evolutionary algorithms include evolutionary programming (EP) (Eiben and Smith, 2003), genetic programming (GP) (Vanneschi and Poli, 2012), differential evolution (DE) (Storn and Price, 1997), genetic algorithm (GA) (Holland, 1975) and evolutionary strategies (ES) (Beyer and Schwefel, 2002).

Physics-based algorithms mimic the rules of physics in the universe and are usually suitable for hybridising with other optimisation techniques. The comparatively popular physics-based algorithms include gravitational search algorithm (GSA) (Rashedi et al., 2009), space gravitational algorithm (SGA) (Shah-Hosseini, 2011) and electromagnetism-like algorithm (EMA) (Birbil and Fang, 2003).

On the contrary, bio-inspired algorithms mimic the social behaviours of a group of creatures. They are primarily the outcome of interspecies and intraspecies interaction in nature, where the essence of interaction can be either cooperative or competitive. Hence, bio-inspired algorithms are mostly regarded as swarm intelligence algorithms. Among popular bio-inspired algorithms are ant colony optimisation (ACO) (Colomi et al., 1991), artificial fish swarm algorithm (AFSA) (Li et al., 2002), artificial bee colony (ABC) (Basturk and Karaboga, 2007) and particle swarm optimisation (PSO) (Eberhrt and Kennedy, 1995). This group of optimisation techniques continues to be a prevalent theme of study. Just recently, in 2020, black widow optimisation (BWO) algorithm was proposed (Hayyolalam and Kazem, 2020). In 2019, emperor penguins colony (EPC), Harris Hawks optimisation (HHO) algorithm (Heidari et al., 2019), artificial coronary circulation system (ACCS) (Kaveh and Kooshkebaghi, 2019), blue monkey (BM) algorithm (Mahmood and Al-Khateeb, 2019) and sunflower optimisation (SFO) algorithm (Gomes et al., 2019) were proposed. In 2018, emperor penguin optimiser (EPO) (Dhiman and Kumar, 2018), pity beetle algorithm (PBA) (Kallioras et al., 2018), lion pride optimisation algorithm (LPOA) (Kaveh and Mahjoubi, 2018) and coyote optimisation algorithm (COA) (Pierezan and Coelho, 2018) were presented. Therefore, it is well-proved that the development of bio-inspired algorithms is still a hot topic.

The continued popularity of biomimetic algorithms has motivated researchers in the field to further investigate any possible developments in this research area. However, while all recently proposed bio-inspired algorithms have the unique ability to support current technological demands, they also have limitations in their selective search capabilities. Therefore, the development of novel algorithm is highly valued. This research work proposes a novel bio-inspired algorithm, referred to as alligator optimisation (AgtrO) algorithm. It mimics the living behaviours of alligators in nature. AgtrO involves two main phases: hunting phase and relocating phase. It is also a meta-heuristic algorithm, aiming to find, generate, or select a heuristic (partial search algorithm), which can provide sufficiently good solutions for optimisation problems, especially in the case of incomplete information or limited computing capability. AgtrO is expected to be able to deal with different optimisation issues, as the hunting phase emphasises global search (including global exploration and global exploitation), and the relocating phase emphasises local search (including local exploration and local exploitation).

This paper is outlined as follows: Section 1 introduces the research work, Section 2 describes the inspiration of this research work, Section 3 explains the methodologies of the proposed AgtrO algorithm, Section 4 describes the simulation setups, analyses the results and discussion, and Section 5 conclude the research work.

2 Inspiration

In 2015, a study conducted by Dinets (2015) from the University of Tennessee, Knoxville has confirmed that alligators often work as teams when hunting. This is indeed a good inspiration for the swarm-based architectures in the algorithm. After more than 3,000 hours of personal observation, Dinets (2015) obtained credible samples about the coordination and cooperation of alligators in nature. An interesting fishing behaviour of American alligators was discovered from his observations. Dinets (2015) observed that three larger alligators drove the fish from the depth of the lake to the shallows, while four smaller and agile alligators blocked their escape and fed on them. The four smaller alligators were waiting for the fish to reach their position and were ready to chase and hunt down the fish when they were in the fishing range. Subsequently, a role exchange was observed. 1 of the 3 larger alligators that once played the role of driving has joined the role of fishing. This is an impressive hunting pattern with a strong emphasis on cooperation, as it reflects the fact that in addition to being just a contributor, a larger alligator can also be a beneficiary from the contributions of other allies. In addition to observing himself, Dinets (2015) also collects reliable samples from another researcher: Chip Campbell stated that American alligators adopt a purse seining behaviour, where a group of alligators eased themselves into a loose semicircle and then approaches inward (close in), pushing schools of fish to entrap them in a shrinking pool. As observed, the alligators execute catching behaviour after purse seining. In this behaviour, they hunt in an orderly manner, taking turns to catch their prey to avoid collisions during foraging. As Campbell also mentioned in the statement, at any given time, about 1/2 of the alligators were resting and waiting for their turn. When an alligator captures a fish, it would swim out and join the resting alligators. At the same time, one of the previously resting alligators would slide out and join the active hunting group. All the available evidence makes alligators social species.

The movement patterns of alligators have also caught research attention. Long ago, in 1984, research has been conducted on the movement of alligators that often live in remote areas through very high-frequency radio telemetry technology (Rodda, 1984). 19 juvenile alligators were displaced to a diameter in the range of 1–10 outside their home range and were tracked by radio telemetry. At least ten of them headed directly homeward and completed their homing journey, but in other cases, habitat obstacles seemed to deviate the alligators' homing path. This research reveals the true navigational capabilities of alligators, thus proving that they have a strong homing instinct. In 2011, alligator movement patterns were quantified with passive acoustic telemetry (Rosenblatt and Heithaus, 2011). In this study, 16 American alligators (all males) were tracked for at least six consecutive months. As a result, three different broad classes of alligator movements were detected among these 16 alligators. The first group (two alligators) remained in the middle of the estuary during the entire detection period. The second group (nine alligators) regularly moves between the middle and downstream areas of the estuary, and occasionally enters the upstream area. The third group (five alligators) often moved between the middle and upstream areas of the estuary and never used the downstream areas. Some of these have shown that alligators have relatively sedentary characteristics (Fujisaki et al., 2014), whereas others have shown that they have the ability to travel long distances (with dispersal into different systems) (Lance et al., 2011). These individual behavioural patterns of alligators: the homing instinct behaviour and travelling behaviour inspired an in-depth idea of novel architecture, that is, the strategy of getting rid of local traps by constantly switching between travelling and homing instinct, repeatedly verifying, and escaping from local optima.

3 Mathematical model of proposed AgtrO algorithm

Inspired by the ecology of alligators in nature mentioned beforehand, we hereby formulate a novel optimisation algorithm, named 'AgtrO algorithm'. AgtrO mimics the living behaviours of alligators by emphasising their cooperative hunting skills and relocating patterns through mathematical expressions. The algorithm proposes a total of two phases (generally including all the behaviours mentioned above), which are:

1 hunting phase

2 relocating phase.

Each phase proposes a main equation, which will be explained and analysed in the following subsections.

3.1 Hunting phase

This is the main phase where the agents imitate the cooperative hunting behaviours of alligators. Agents act systematically according to the concept of hunting, where they are interdependent of each other during simulation. The mechanism in the hunting phase can be expressed mathematically as follows:

$$\boldsymbol{X}_i^{t+1} = \boldsymbol{X}_i^t + \vec{\boldsymbol{W}} \bullet \overrightarrow{\boldsymbol{V}_{F_i}}^t + \left(1 - \vec{\boldsymbol{W}}\right) \bullet \left[\overrightarrow{\boldsymbol{V}_{PS_i}}^t + F_i \overrightarrow{\boldsymbol{V}_{C_i}}^t\right] \quad (1)$$

where $\bullet$ denotes the element-by-element multiplication, $t \in \{1, \ldots, t_{\max}\}$ denotes the index of iteration, $i \in \{1, \ldots, N_{Pop}\}$ denotes the index of population, and $\vec{\boldsymbol{W}}$ represents the weight vector. Note that $t_{\max}$ is the maximum number of iterations and N_{Pop} is the maximum population number. F represents the flag, so F_i is the flag for the i^{th} agent. $\boldsymbol{X}$ represents the position point, hence $\boldsymbol{X}_i^t$ is the position point of the i^{th} agent at t^{th} iteration, and $\boldsymbol{X}_i^{t+1}$ is the position point of the i^{th} agent at $(t + 1)^{\text{th}}$ iteration. According to equation (1), the hunting mechanism is modelled as a combination of three position vectors: $\overrightarrow{\boldsymbol{V}_{F_i}}^t$, $\overrightarrow{\boldsymbol{V}_{PS_i}}^t$ and $\overrightarrow{\boldsymbol{V}_{C_i}}^t$, where each position vector represents each hunting element in nature.

$\overrightarrow{V_{F_i}}^t$ is a position vector imitating fishing behaviour, $\overrightarrow{V_{PS_i}}^t$ is a position vector imitating purse seining behaviour, while $\overrightarrow{V_{C_i}}^t$ is a position vector imitating catching behaviour.

3.1.1 Fishing stage

Inspired by the fishing mechanism in nature, $\overrightarrow{V_{F_i}}^t$ is formulated to operate as a global exploration vector, where $\overrightarrow{V_{F_i}}^t$ can be expressed as follows:

$$\overrightarrow{V_{F_i}}^t = (1+r_1)\left(\boldsymbol{X}_{prey} - \boldsymbol{X}_i^t\right) \tag{2}$$

where r_1 is a random number within 0 and 1, and $\boldsymbol{X}_{prey}$ is the position of prey driven by other hunting agents. To ensure that the prey is driven to a position closer to or in front of the i^{th} agent, $\boldsymbol{X}_{prey}$ is expressed as follows:

$$\boldsymbol{X}_{prey} = 0.9\boldsymbol{X}_i^t + 0.1\frac{\sum_{j=1}^{n}\boldsymbol{X}_{ldr,j}}{n} \tag{3}$$

where $\boldsymbol{X}_{ldr}$ is the leader solution, and n is a random integer within 1 and 3. The i^{th} agent in execution can select at least 1 and at most 3 agents that currently have better performing objective fitness values than the i^{th} agent during the iterative process. We refer these selected agents as leaders and their current positions as the leader solutions, $\boldsymbol{X}_{ldr}$. Note that the leaders are playing the role of the larger alligators, driving the prey to the front of the i^{th} agent. The stochastic selection of n leaders is conducive to global exploration, making the search more decentralised and effective, thereby accelerating the convergence speed towards optimised solution achievement.

During the fishing mechanism, the i^{th} agent moves in a promising direction towards the highly probable global optimal regions that these leader agents have explored in advance. This improves the efficiency of exploration, thereby ensuring a promising global exploratory search for the i^{th} agent. In theory, it is expected to further speed up the convergence speed of the optimisation.

Equation (2) can be executed normally by every agent at each iteration, but there are exceptional cases, where (in terms of objective fitness):

- The 1st ranked agent will not execute the fishing mechanism [equation (2)] because it is currently the best performing agent (that is, in this iteration, there is currently no other agent with better objective fitness value than it). In other words, no one can be selected for help.
- The 2nd ranked agent can only select the 1st ranked agent for help because there is no other selection with better an objective fitness value.
- The 3rd ranked agent can only select the 1st ranked or 2nd ranked agent, or both, for help.

3.1.2 Purse seining stage

Inspired by the purse seining behaviour of alligators, $\overrightarrow{V_{PS_i}}^t$ is formulated to operate as a global exploitation vector, where $\overrightarrow{V_{PS_i}}^t$ is expressed as follows:

$$\overrightarrow{V_{PS_i}}^t = (1-A)\left(\boldsymbol{X}_{gbest}^t - \boldsymbol{X}_i^t\right) \tag{4}$$

where $\boldsymbol{X}_{gbest}^t$ is the global best solution at t^{th} iteration, which is defined as the best solution among all the positions that the population have visited so far throughout the iteration, and A is the coefficient value, which can be expressed as:

$$A = -\alpha + 2\alpha r_2 \tag{5}$$

where r_2 is a random number within 0 and 1, used to assign different coefficient values of A to respective agents to achieve a variable and flexible hunting range, and α is an adaptive parameter determined using the following expression:

$$\alpha = \left(1 - \frac{t}{t_{\max}}\right)^2 \tag{6}$$

Equation (5) well reflects that the coefficient A is a random variable in the range from $-\alpha$ to α. The parameter α gradually declines from 1 to 0 at a slight decreasing rate, as to arrange for earlier access to detailed exploitation to improve the global-best achievement in AgtrO.

Agents refer to the global best solution, $\boldsymbol{X}_{gbest}^t$ as the location of prey to be purse seined. During execution, the i^{th} agent updates its position point by bringing itself closer to $\boldsymbol{X}_{gbest}^t$. Each agent has the natural characteristics of an alligator, which can accurately recognise the exact location of the prey and other agents. This is well reflected in Equation (4), that is, the i^{th} agent can accurately locate $\boldsymbol{X}_{gbest}^t$ without bias.

From mathematical perspective, a group of agents work together to form a loose semicircle to surround the prey. During execution, the agent updates its position along the line pointing to the global best solution, $\boldsymbol{X}_{gbest}^t$, depending on the value of A. The smaller the value of $|A|$, the closer it will be to $\boldsymbol{X}_{gbest}^t$. If A has a negative value, the agent will choose to attack from the opposite direction. In theory, this is to ensure that the targeted prey can be surrounded from two opposite directions without leaving any escape routes.

3.1.3 Catching stage

Inspired by the systematic catching behaviour, we formulated $\overrightarrow{V_{C_i}}^t$ as follows:

$$\overrightarrow{V_{C_i}}^t = (1+M)\left(\boldsymbol{X}_{gbest}^t - \left(\boldsymbol{X}_i^t + \overrightarrow{V_{PS_i}}^t\right)\right) \tag{7}$$

where M represents catching momentum, which is expressed as follows:

$$M = e\left(\frac{f\left(X_{gbest}^t\right) - f\left(X_i^t\right)}{\max\left(f\left(X^t\right)\right) - \min\left(f\left(X^t\right)\right)}\right) \quad (8)$$

where $f(\ )$ is the objective function that returns the objective fitness value, $e(\ )$ returns the exponential value, $\max(f(X^t))$ returns the maximum element of f $f(X^t)$ array, and $\min(f(X^t))$ returns the minimum element of $f(X^t)$ array. The multiplication with $(1 + M)$ can be referred to as the propulsion exerted by the i^{th} agent to catch the prey in X_{gbest}^t from X_i^t, where 1 ensures that the i^{th} agent reaches X_{gbest}^t, and M provides the tendency to stop the i^{th} agent within an acceptable distance behind X_{gbest}^t. Note that the final position of the agent in execution will certainly not break through the semi-circular purse seining net that theoretically surrounds and entraps the prey.

$\overrightarrow{V_{C_i}}^t$ becomes a supportive vector for $\overrightarrow{V_{PS_i}}^t$, which is also a secondary global exploitation vector in AgtrO. During execution, $\overrightarrow{V_{C_i}}^t$ sharpens the global exploitation search by contrasting the difference between $f(X_{gbest}^t)$ and $f(X_i^t)$ to estimate the appropriate momentum M in equation (8). The greater the difference, the smaller the momentum M of the i^{th} agent. As the huge difference in objective fitness values indicates that the i^{th} agent has not fully converged, hence it provides less intensive momentum to bring the final position point of the i^{th} agent closer to the opposite of X_{gbest}^t. This is beneficial for global exploitation, as it can accelerate the search tempo by fine-tuning the agent to the desired position as per determined by the current fitness.

By definition, global exploration is a search operation to discover the global optimal area, and global exploitation is a search operation to track the true global optimal solution that is likely to exist somewhere within the global optimal area. Therefore, emphasising global exploration in the early iterations and global exploitation in the later iterations is an ideal strategy to achieve good solutions in optimisation. Equation (1) follows this concept by having the weight vector $\vec{W}$ varies from 0 to 1 to distribute the ratio of global exploration and global exploitation, where $\vec{W}$ is formulated as follows:

$$\vec{W} = \beta r_3 \quad (9)$$

where r_3 is a random array within 0 and 1, which allocate variable vector values of $\vec{W}$ to respective agents, and β is the parameter that determines the maximum allowable vector value in each iteration loop, which is adapted using the following expression:

$$\beta = \left(1 - \frac{t}{t_{\max}}\right)^{\frac{2t}{t_{\max}}} \quad (10)$$

The parameter β is halved from its maximum value when it reaches 50% of the entire iteration. This is where the major proportions begin to shift from global exploration to global exploitation. The idea is to allocate the first half iteration to a higher proportion of global exploration and the second half iteration to a higher proportion of global exploitation to balance the contradiction between the two search operations. In this way, the agents have been given enough iterations to explore a global optimal area and then continues to exploit the true global optimal solution within this area.

In equation (1), F_i is inserted into $\overrightarrow{V_{C_i}}^t$ to turn on or off the execution of $\overrightarrow{V_{C_i}}^t$. This allows the i^{th} agent to mimic the nature of alligators to 'take turns' in executing the catching behaviour. F_i can be expressed as follows:

$$F_i = \begin{cases} 1 & i\%3 = t \\ 0 & i\%3 \neq t \end{cases} \quad (11)$$

where % denotes the modulus operation. Whenever $F_i = 1$, $\overrightarrow{V_{C_i}}^t$ works normally for the i^{th} agent at that iteration, and whenever $F_i = 0$, $\overrightarrow{V_{C_i}}^t$ is temporarily terminated for the i^{th} agent at the iteration. The number of modulo 3 in equation (11) brings another implication, that is, only one out of three agents is allowed to join the catching stage at a time.

3.2 Relocating phase

In the non-hunting phase, alligators choose to rest or relocate. Since the resting phase is not conducive for optimisation, we mainly imitate the relocating patterns of the alligators. In mathematical terms, this is the stage where the agents mimic the instinct of wild alligators to explore new habitats or return home. The mechanism of the relocating phase can be mathematically expressed as follows:

$$X_i^{t+1} = X_i^t + \overrightarrow{V_{T/H_i}}^t + \overrightarrow{V_{W_i}}^t \quad (12)$$

According to Equation (12), the relocating mechanism is modelled as a combination of two position vectors: $\overrightarrow{V_{T/H_i}}^t$ and $\overrightarrow{V_{W_i}}^t$, where $\overrightarrow{V_{T/H_i}}^t$ is a position vector imitating both travelling or homing instinct behaviour, while $\overrightarrow{V_{W_i}}^t$ is a position vector that simulates the external causes of uncontrollable waves.

3.2.1 Travelling or homing instinct stage

With reference to the travelling and homing instinct behaviours of alligators in nature, we could formulate $\overrightarrow{V_{T/H_i}}^t$ as follows:

$$\overrightarrow{V_{T/H_i}}^t = (1-|C|)\left(2r_4 X^t_{i,lbest} - X^t_i\right) \tag{13}$$

where | | denotes the absolute value, r_4 is a random number within 0 and 1, and $X^t_{i,lbest}$ is the local best solution of the i^{th} agent. The i^{th} agent regards $X^t_{i,lbest}$ as its home-territory, where $X^t_{i,lbest}$ can also be defined as the best solution among all the positions that the i^{th} agent has visited so far throughout the iteration. If a better solution is found by the i^{th} agent at t^{th} iteration, the i^{th} agent will replace $X^{t+1}_{i,lbest}$ with this solution, indicating that the agent can imitate the nature of creatures to migrate to a more comfortable living space. As the memory is fuzzy and unreliable, r_4 is used as a random number to offset the exact position of $X^t_{i,lbest}$, which imitate that an alligator vaguely remembering the direction towards the home area. C is the coefficient value, which can be expressed as follows:

$$C = -\delta + 2r_5\delta \tag{14}$$

where r_5 is a random number within 0 and 1, used to assign different coefficient values of C to respective agents to achieve a variable and flexible relocating range, and δ is the parameter adapted using the following expression:

$$\delta = 20\left(1-\left(\frac{t}{t_{\max}}\right)^{5\left(1-\frac{t}{t_{\max}}\right)}\right)^5 \tag{15}$$

Equation (14) well reflects that the coefficient C is a random variable in the range from $-\delta$ to δ, where the parameter δ declines from 20 at an increasing rate at the first half of the iteration and then declines to 0 at a decreasing rate until the end of the iteration. The agent either enters the travelling mechanism or the homing instinct mechanism depending on the coefficient value of C. When $|C| \geq 1$, the equation (13) operates as the travelling vector $\overrightarrow{V_{T_i}}^t$, where the i^{th} agent moves away from $X^t_{i,lbest}$ as the intention to imitate the long-distance travelling behaviour. When $|C| < 1$, Equation (13) operates as the homing instinct vector $\overrightarrow{V_{H_i}}^t$ where the i^{th} agent moves toward $X^t_{i,lbest}$ as the intention to imitate the homing instinct behaviour. In fact, δ is distributed at a ratio of 19:1, representing the available step range allocated to the travelling and homing instinct mechanisms, respectively.

Through their relocating patterns, we know that $\overrightarrow{V_{T_i}}^t$ serves as a local exploration vector to help the i^{th} agent escape the most likely local trap in $X^t_{i,lbest}$, and $\overrightarrow{V_{H_i}}^t$ serves as a local exploitation vector to allow the i^{th} agent to return to the vicinity of $X^t_{i,lbest}$ to find any possible better solution. As we expand the analysis, 64% of the entire iteration becomes the turning point for the transition from local exploration to local exploitation stage, exactly when $\delta \approx 1$. Hence, it is roughly estimated that there are 64% local exploration and 36% local exploitation throughout the total iterations.

3.2.2 Environmental Wave

By considering the environmental effects on relocating mechanism, $\overrightarrow{V_{W_i}}^t$ is formulated as follows:

$$\overrightarrow{V_{W_i}}^t = (1\pm\omega)\cdot\left(X^t_{i,lbest} - X^t_i\right) \tag{16}$$

where the coefficient ω can be expressed as follows:

$$\omega = e^B\left(L\left(\cos(2\pi B)\right) + (1-L)\left(\sin(2\pi B)\right)\right) \tag{17}$$

where L is a binary logical array, used to select sine or cosine element for a specific dimension in equation (17), and B is the coefficient value, which can be expressed as follows:

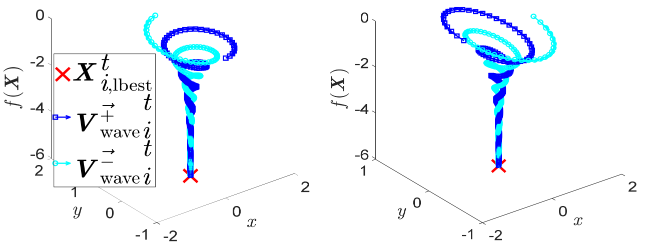
$$B = \gamma r_6 \tag{18}$$

where r_6 is a random number within 0 and 1, mainly used to assign different coefficient values of B to respective agents, and γ is the parameter adapted using the following expression:

$$\gamma = -50\left(1-\frac{t}{t_{\max}}\right)^5 \tag{19}$$

$\overrightarrow{V_{W_i}}^t$ is neither an exploration vector nor an exploitation vector for the i^{th} agent. Instead, it represents an additional local displacement vector that has a strong influence on the relocating mechanisms of the i^{th} agent. $\overrightarrow{V_{W_i}}^t$ mimics the wave vector of vortex in nature, where the centre point of the vortex imposed on the i^{th} agent is defined as $X^t_{i,lbest}$. The combination of sine and cosine elements in equation (17) constitutes the helical shape of the vortex, as shown in Figure 1. As we expand the analysis, $\overrightarrow{V_{W_i}}^t$ serves as a clockwise or anti-clockwise vortex vector, pulling the agent toward the centre of the vortex by force, as the intention of trapping the i^{th} agent in $X^t_{i,lbest}$. $\overrightarrow{V_{W_i}}^t$ plays its true value through combination with $\overrightarrow{V_{T_i}}^t$ and $\overrightarrow{V_{H_i}}^t$, where it acts as a supportive vector to $\overrightarrow{V_{T/H_i}}^t$. It first displaces the initial X^t_i of the i^{th} agent to the either (±) side of the tail of the vortex, thereby imposing an inward circular position vector, which neither significantly promote nor demote $\overrightarrow{V_{T/H_i}}^t$, but shifts the vector direction of $\overrightarrow{V_{T/H_i}}^t$ for more diverse and credible local search in the entire relocating mechanism.

Figure 1 3D display of wave vector in AgtrO, (a) anti-clockwise wave vector (b) clockwise wave vector (see online version for colours)



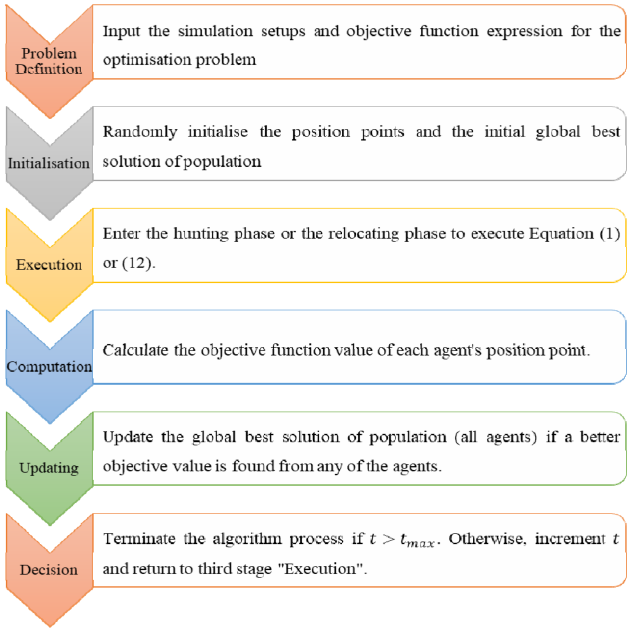
(a) (b)

3.3 *Complete mechanism*

The proposed AgtrO is a meta-heuristic algorithm. There are two main executable phases: the hunting phase and the relocating phase. Both phases have their own advantages, and only one phase may not be able to solve specific problems. Each of them addresses a given type of optimisation problem from their own perspective, and hence, the combination of the two phases in AgtrO is expected to solve all the included problems. Overall speaking, the hunting phase concerns on large-scale search, clustering, global optimal approach, and cooperation, which is conducive to deal with the unimodal objective functions. On the other hand, the relocating phase concerns on getting rid of local optima (traps), proving the rationality of the local optimal approaching, making personal decisions, and staying away from the cluster, which is conducive to deal with the multimodal objective functions.

The agents (alligators) are independent of one other and will decide which phase to enter during each iteration based on random selection, without using unnecessary decision coefficients. This is to ensure that agents are not restricted by any fixed decision-making patterns and to prevent the execution time of the algorithm from being prolonged due to excessive patterns of selection. Each agent has half the chance to enter hunting phase or relocating phase, deciding on whether to execute equation (1) or equation (12), respectively. Both mathematically simulate the real behaviours of alligators in nature, while maintaining good performance for optimisation. Since the displaceable distance of the agent is varied indirectly through random numbers or random array, it brings to a more flexible change of pace, so that each iterative update of position is not regular. In addition, agents can share and update the same information on $\boldsymbol{X}_{gbest}^{t}$ even if they enter different phases in the iteration. The interaction among the entire population is considered to produce better performance because it can share the benefits from both phases through the exchange of information during each iterative update of $\boldsymbol{X}_{gbest}^{t}$ solution. For ease of understanding, a vector diagram of the AgtrO mechanism is presented in Figure 2.

Figure 2 Vector diagram of the proposed AgtrO mechanism (see online version for colours)



4 Results and discussion

AgtrO was evaluated by comparing its performance with other optimisation algorithms. aquila optimiser (AO) algorithm (Abualigah et al., 2021b), arithmetic optimisation algorithm (AOA) (Abualigah et al., 2021a), equilibrium optimiser (EO) (Faramarzi et al., 2020), social ski driver (SSD) algorithm (Tharwat and Gabel, 2020), atom search optimisation (ASO) (Zhao et al., 2019), seagull optimisation algorithm (SOA) (Dhiman and Kumar, 2019), sunflower optimisation (SFO) algorithm (Gomes et al., 2019) and grasshopper optimisation algorithm (GOA) (Saremi et al., 2017) were used as comparative algorithms. These eight comparative algorithms are amongst recently proposed novel algorithms and published in high-impact journals.

Table 1 displays the parameter settings of AgtrO and other comparative algorithms. In AgtrO, all parameters are variable, they are not fixed, their values change with iterations.

The proposed AgtrO and the other algorithms were simulated on 23 classical benchmark problems (Yao et al., 1999; Digalakis and Margaritis, 2002; Yang, 2010) and ten modern CEC-C06-2019 benchmark problems (Price et al., 2018; Abdullah and Rashid, 2019). Among these classical functions, the 1st to 7th are unimodal functions, the 8th to 13th are multimodal functions, and the rest are fixed-dimensional multimodal functions. On the other hand, CEC-C06-2019 benchmark functions were improved for a single objective optimisation problem. All test functions from CEC-C06-2019 are scalable, while only functions CEC04 to CEC10 are shifted and rotated.

Table 1 Parameter settings of AgtrO and other comparative algorithms

Parameter	*Range or value*
AgtrO	
α	[0, 1]
β	[0, 1]
δ	[0, 20]
γ	[–50, 0]
AO	
α	0.1
δ	0.1
u	0.0265
r_0	10
ω	0.005
φ_0	3π/2
AOA	
$MOP_{\max}$	1
$MOP_{\min}$	0.2
α	5
μ	0.499
EO	
α_1	2
α_2	1
GP	0.5
SSD	
α	[0, 2]
ASO	
α	50
β	0.2
SOA	
FC	[0, 2]
b	1
SFO	
ρ	0.05
m	0.1
s	$1 - \rho + m$
GOA	
$c_{\max}$	1
$c_{\min}$	0.00004
f	0.5
ℓ	1.5

Through preliminary tests, we concluded that the problem definition of $N_{Pop} = 50$ and $t_{\max} = 500$ is sufficient for the simulation on 23 classical benchmark functions, and the problem definition of $N_{Pop} = 100$ and $t_{\max} = 100$ is suitable for the simulation on 10 CEC-C06-2019 benchmark functions. For an equitable comparison, these problem definitions were employed equally to all comparative algorithms, including AgtrO.

The following subsections contain:

1 classical benchmark evaluation

2 CEC-C06-2019 benchmark evaluation

3 comprehensive result analysis and discussion.

4.1 Classical benchmark

Table 2 collects the statistical result data [in terms of mean and standard deviation (SD)] of AgtrO and eight state-of-the-art comparative algorithms on 23 classical benchmark functions. The superiority of the mean value guarantees the accuracy of the global-best achievement, and the superiority of the SD value secures the precision and robustness of the algorithm. For in-depth analysis, we rank these algorithms based on their mean and SD values on each function (f_{1-23}). On a total of 23 classical optimisation benchmark functions, AgtrO ranked 1st on the greatest number of benchmark functions (i.e., 10 benchmark functions), ranked at least 3rd out of 20 benchmark functions, and has never ranked below 5th place among all compared algorithms. On the classical benchmark functions, AgtrO is absolutely the best in terms of performance ranking. Therefore, we can claim with certainty that among all the comparative algorithms, AgtrO is the algorithm with the best performance in terms of global-best achievement.

It is worth mentioning that the multimodal optimisation functions theoretically have higher possibilities to trap agents in local optima, which will completely immobilise the agent in execution. This terminates the update of better solutions in the iterative process and ultimately deteriorates the optimisation performance. However, as can be verified from Table 2, AgtrO has achieved superior mean and SD results regarding the global best objective fitness values on f_{9-11} and f_{14-23}. This supports the statement that AgtrO implements a high escaping skill during the local exploration stage. This was well-articulated in the explanation of equation (13) on how it contributes to get rid of local optima as a travelling vector.

In addition, we inspect the convergence speed of AgtrO on f_{1-23} as compared with other algorithms. Figure 3 displays the convergence curves of all algorithms on each classical benchmark function. Among them, AgtrO is claimed to have a satisfactory convergence speed, as most AgtrO curves could converge to their minimal fitness values in about 5%–50% of the entire iteration. Although the convergence speed of AgtrO was slightly slower than that of AO, EO and SFO on a few benchmark functions, AgtrO could converge to a more minimal global best objective fitness value as compared with these three algorithms, except for f_6 and f_{12-13}. Due to these facts, we can claim that AgtrO performs well in terms of convergence speed.

Table 2 Statistical results of AgtrO and eight state-of-art algorithms on 23 classical benchmark functions (see online version for colours)

FUNC.	*IND*	*Algorithm*								
		AgtrO (proposed)	*AO (2021)*	*AOA (2021)*	*EO (2020)*	*SSD (2020)*	*ASO (2019)*	*SOA (2019)*	*SFO (2018)*	*GOA (2017)*
f_{1}	Mean	*1.43E-229*	4.00E-109	1.63E-13	1.13E-48	1.11E+03	7.31E-19	5.96E-13	2.67E+01	3.28E+00
	SD	*0.00E+00*	4.00E-108	1.63E-12	2.85E-48	4.24E+02	8.45E-19	1.36E-12	2.49E+01	1.77E+00
f_{2}	Mean	3.78E-118	2.15E-65	*0.00E+00*	3.89E-28	1.57E+01	1.48E-06	5.16E-09	8.68E-01	6.00E+00
	SD	2.18E-117	1.70E-64	*0.00E+00*	5.36E-28	2.53E+00	1.44E-05	5.66E-09	5.94E-01	1.45E+01
f_{3}	Mean	*4.60E-223*	1.00E-107	3.46E-03	1.82E-12	6.50E+03	1.36E+03	1.61E-05	1.25E+03	1.65E+03
	SD	*0.00E+00*	8.80E-107	7.97E-03	6.60E-12	2.30E+03	5.50E+02	1.14E-04	1.14E+03	1.09E+03
f_{4}	Mean	*1.02E-115*	3.48E-55	2.05E-02	2.14E-12	1.32E+01	3.12E-07	2.10E-03	1.64E+00	8.50E+00
	SD	*9.69E-115*	3.48E-54	2.01E-02	3.41E-12	2.67E+00	3.95E-07	6.31E-03	9.03E-01	3.08E+00
f_{5}	Mean	2.73E+01	*2.00E-03*	2.83E+01	2.48E+01	6.96E+04	4.23E+01	2.81E+01	9.43E+01	8.36E+02
	SD	5.55E-01	*3.65E-03*	3.42E-01	1.83E-01	5.43E+04	4.36E+01	5.97E-01	2.27E+02	7.49E+02
f_{6}	Mean	1.60E-01	6.81E-05	2.89E+00	1.52E-07	1.01E+03	*6.36E-19*	2.82E+00	2.83E+01	3.09E+00
	SD	1.80E-01	3.20E-04	2.69E-01	2.45E-07	2.93E+02	*9.11E-19*	4.18E-01	2.70E+01	1.90E+00
f_{7}	Mean	4.72E-04	5.79E-05	*4.02E-05*	7.74E-04	7.10E-02	4.37E-02	1.77E-03	7.03E-03	2.74E-02
	SD	4.56E-04	5.39E-05	*3.91E-05*	4.36E-04	3.54E-02	2.01E-02	1.50E-03	8.35E-03	1.22E-02
f_{8}	Mean	–9.51E+03	–8.30E+03	–5.60E+03	–9.02E+03	–3.15E+03	–7.27E+03	–5.27E+03	*–1.25E+04*	–7.57E+03
	SD	1.36E+03	3.73E+03	3.61E+02	6.44E+02	4.09E+02	6.14E+02	6.47E+02	*6.19E+01*	6.64E+02
f_{9}	Mean	*0.00E+00*	0.00E+00	0.00E+00	0.00E+00	9.77E+01	2.72E+01	2.08E+00	4.33E+00	8.42E+01
	SD	*0.00E+00*	0.00E+00	0.00E+00	0.00E+00	1.73E+01	6.72E+00	4.60E+00	2.83E+00	3.54E+01
f_{10}	Mean	*8.88E-16*	8.88E-16	8.88E-16	7.50E-15	8.23E+00	4.98E-10	2.00E+01	1.87E+00	3.35E+00
	SD	*0.00E+00*	0.00E+00	0.00E+00	1.24E-15	8.30E-01	2.59E-10	1.64E-03	1.04E+00	7.66E-01
f_{11}	Mean	*0.00E+00*	0.00E+00	1.18E-01	1.23E-04	1.10E+01	1.04E-03	1.47E-02	1.23E+00	6.27E-01
	SD	*0.00E+00*	0.00E+00	9.86E-02	1.23E-03	3.48E+00	3.27E-03	3.60E-02	3.55E-01	1.82E-01
f_{12}	Mean	1.71E-02	1.30E-06	4.33E-01	*5.13E-09*	1.01E+01	7.38E-03	2.39E-01	1.23E-01	5.22E+00
	SD	1.18E-02	2.09E-06	4.71E-02	*1.03E-08*	3.70E+00	2.58E-02	1.17E-01	1.28E-01	2.31E+00
f_{13}	Mean	1.72E+00	*1.20E-05*	2.83E+00	1.40E-02	2.97E+03	1.13E-03	1.89E+00	6.47E-01	1.38E+01
	SD	9.13E-01	*2.17E-05*	1.15E-01	3.21E-02	1.62E+04	3.23E-03	1.83E-01	8.03E-01	1.75E+01
f_{14}	Mean	9.98E-01	1.66E+00	9.47E+00	*9.98E-01*	1.34E+00	1.21E+00	1.65E+00	1.48E+00	9.98E-01
	SD	2.19E-15	1.51E+00	3.95E+00	*1.16E-16*	6.03E-01	4.44E-01	1.55E+00	8.54E-01	6.69E-16
f_{15}	Mean	6.30E-04	*4.51E-04*	9.91E-03	2.37E-03	4.54E-03	1.02E-03	1.19E-03	8.45E-04	1.05E-02
	SD	3.86E-04	*1.24E-04*	1.88E-02	6.03E-03	6.46E-03	2.66E-04	2.07E-04	5.94E-04	1.84E-02
f_{16}	Mean	*–1.03E+00*	–1.03E+00	–1.03E+00	–1.03E+00	–1.03E+00	–1.03E+00	–1.03E+00	–1.03E+00	–1.03E+00
	SD	*1.46E-15*	2.65E-04	8.76E-08	1.51E-15	8.07E-07	1.54E-15	9.41E-07	1.86E-07	5.80E-13
f_{17}	Mean	3.98E-01	3.98E-01	3.98E-01	*3.98E-01*	4.18E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01
	SD	1.07E-15	1.39E-04	4.50E-08	*1.06E-15*	9.62E-02	1.06E-15	5.74E-05	6.87E-08	3.05E-13
f_{18}	Mean	3.00E+00	3.02E+00	5.97E+00	*3.00E+00*	3.01E+00	3.00E+00	3.00E+00	3.00E+00	4.62E+00
	SD	3.87E-07	1.54E-02	8.49E+00	*9.00E-16*	7.14E-02	2.21E-15	3.86E-05	6.07E-06	1.15E+01
f_{19}	Mean	–3.86E+00	–3.86E+00	–3.85E+00	–3.86E+00	–3.81E+00	*–3.86E+00*	–3.85E+00	–3.86E+00	–3.85E+00
	SD	1.42E-06	4.00E-03	3.07E-03	7.88E-04	4.61E-02	*6.31E-15*	9.97E-04	2.75E-07	1.09E-01
f_{20}	Mean	–3.28E+00	–3.19E+00	–3.09E+00	–3.26E+00	–2.74E+00	*–3.32E+00*	–3.01E+00	–3.26E+00	–3.29E+00
	SD	5.72E-02	8.17E-02	7.89E-02	6.19E-02	3.43E-01	*2.23E-15*	2.56E-01	7.36E-02	5.47E-02
f_{21}	Mean	*–1.02E+01*	–1.01E+01	–3.97E+00	–9.14E+00	–5.14E+00	–7.95E+00	–4.70E+00	–9.85E+00	–6.89E+00
	SD	*4.72E-11*	6.74E-03	1.15E+00	2.13E+00	3.37E+00	3.26E+00	4.29E+00	7.09E-01	3.40E+00
f_{22}	Mean	*–1.04E+01*	–1.04E+01	–3.90E+00	–9.93E+00	–6.14E+00	–1.03E+01	–7.75E+00	–1.01E+01	–6.66E+00
	SD	*2.21E-11*	9.24E-03	1.60E+00	1.64E+00	3.25E+00	8.43E-01	3.78E+00	7.96E-01	3.60E+00
f_{23}	Mean	*–1.05E+01*	–1.05E+01	–4.08E+00	–1.00E+01	–6.75E+00	–1.02E+01	–7.65E+00	–1.04E+01	–7.75E+00
	SD	*3.01E-11*	7.31E-03	1.61E+00	1.91E+00	3.36E+00	1.38E+00	3.79E+00	5.35E-01	3.67E+00

Note: **Highlighted and italic data reveal the best performing results among the compared algorithms.

Figure 3 Sample of convergence curves of AgtrO and other algorithms on f_{1-23} (see online version for colours)

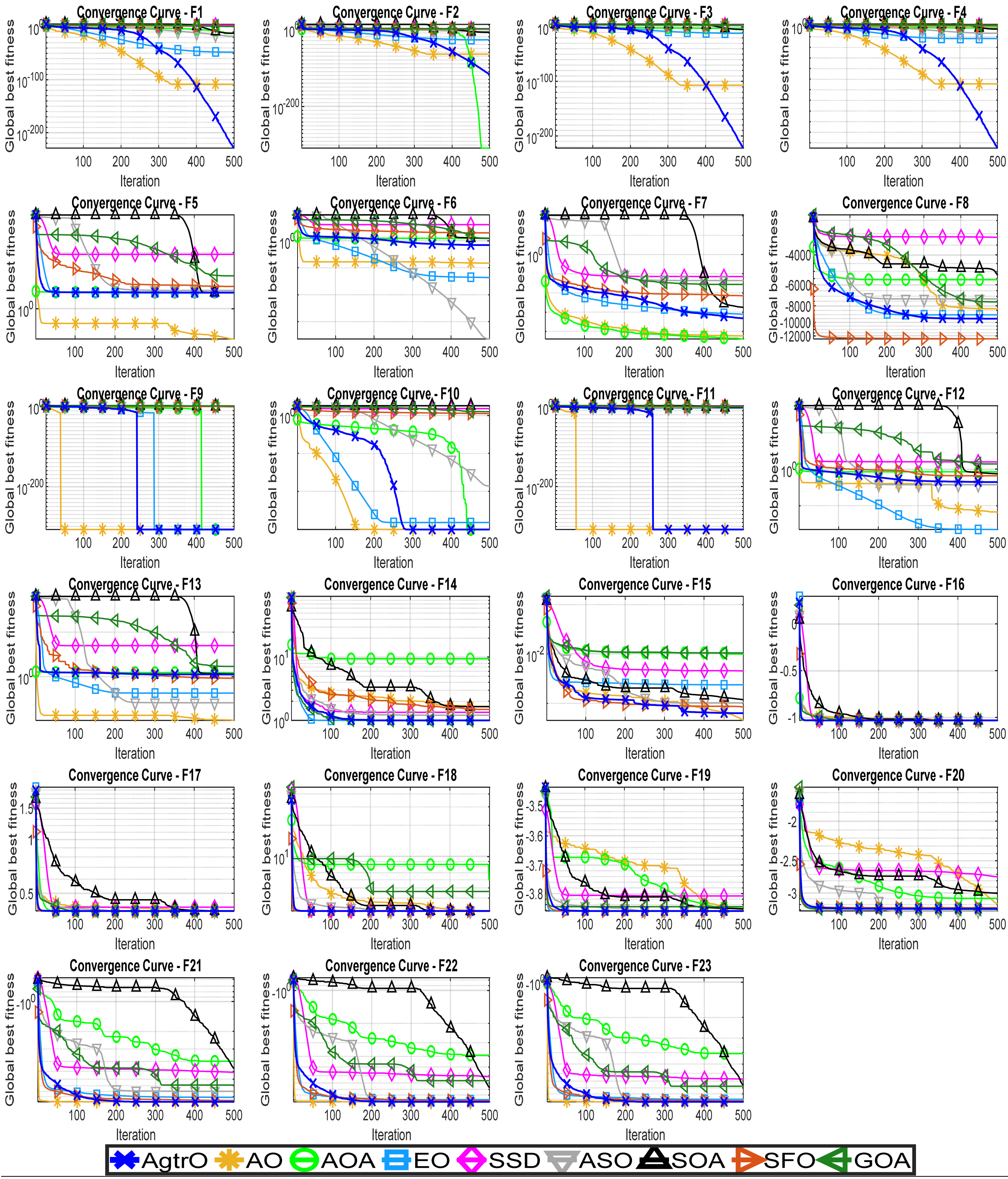
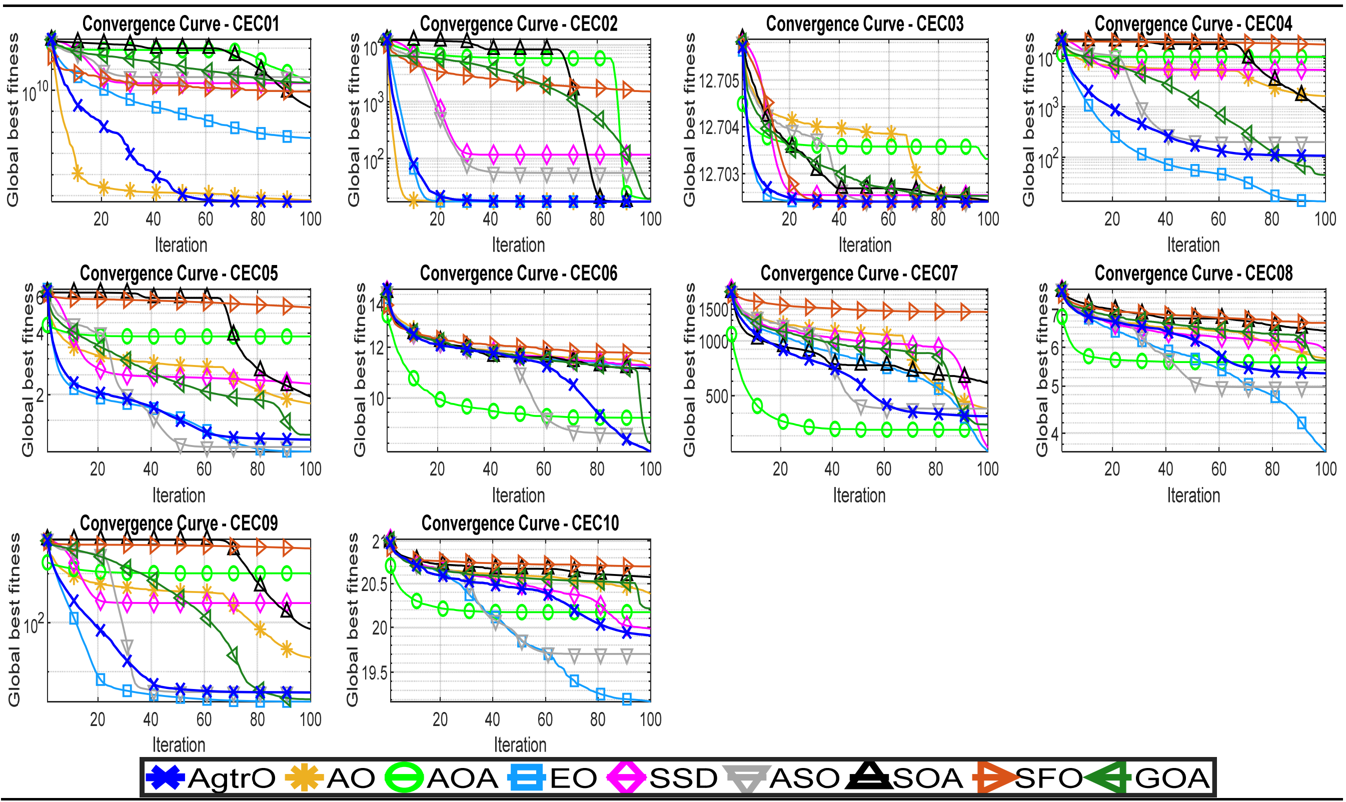


Table 3 Statistical results of AgtrO and eight other comparative algorithms on 10 CEC-C06-2019 benchmark functions (see online version for colours)

FUNC.	*IND*	*Algorithm*								
		AgtrO (proposed)	*AO (2021)*	*AOA (2021)*	*EO (2020)*	*SSD (2020)*	*ASO (2019)*	*SOA (2019)*	*SFO (2018)*	*GOA (2017)*
f_{CEC01}	Mean	*5.18E+04*	6.14E+04	2.03E+10	5.39E+07	2.14E+10	4.19E+10	1.52E+09	8.52E+09	2.32E+10
	SD	*6.39E+03*	1.58E+04	4.68E+10	1.01E+08	2.13E+10	4.16E+10	2.26E+09	2.48E+10	2.82E+10
f_{CEC02}	Mean	1.74E+01	1.74E+01	1.94E+01	*1.73E+01*	1.16E+02	5.56E+01	1.74E+01	1.50E+03	1.94E+01
	SD	5.56E-02	1.89E-02	3.47E-01	*1.96E-08*	1.64E+02	2.38E+01	9.91E-02	9.03E+02	9.27E+00
f_{CEC03}	Mean	1.27E+01	1.27E+01	1.27E+01	*1.27E+01*	1.27E+01	1.27E+01	1.27E+01	1.27E+01	1.27E+01
	SD	1.24E-07	1.64E-05	9.27E-04	*4.04E-08*	2.00E-04	1.13E-05	1.18E-05	2.57E-05	4.65E-04
f_{CEC04}	Mean	1.08E+02	1.63E+03	9.63E+03	*1.36E+01*	5.36E+03	2.01E+02	7.85E+02	1.70E+04	4.53E+01
	SD	5.17E+01	1.06E+03	3.33E+03	*6.69E+00*	1.99E+03	1.34E+02	9.50E+02	4.57E+03	1.79E+01
f_{CEC05}	Mean	1.21E+00	1.81E+00	3.85E+00	*1.05E+00*	2.26E+00	1.11E+00	1.95E+00	5.33E+00	1.27E+00
	SD	1.14E-01	2.13E-01	7.64E-01	*4.41E-02*	4.55E-01	1.70E-01	1.48E-01	8.89E-01	1.28E-01
f_{CEC06}	Mean	*8.25E+00*	1.12E+01	9.32E+00	1.11E+01	1.12E+01	8.81E+00	1.11E+01	1.17E+01	8.55E+00
	SD	*1.28E-01*	8.57E-01	9.47E-01	7.36E-01	8.31E-01	2.59E+00	7.06E-01	9.06E-01	1.09E+00
f_{CEC07}	Mean	3.85E+02	4.24E+02	3.24E+02	*2.45E+02*	2.58E+02	4.24E+02	5.84E+02	1.45E+03	3.47E+02
	SD	2.61E+02	2.30E+02	*1.60E+02*	1.95E+02	1.67E+02	3.12E+02	1.56E+02	1.92E+02	2.38E+02
f_{CEC08}	Mean	5.31E+00	5.69E+00	5.59E+00	*3.67E+00*	5.89E+00	4.97E+00	6.48E+00	6.73E+00	5.64E+00
	SD	7.60E-01	5.06E-01	5.78E-01	1.13E+00	4.29E-01	1.03E+00	4.68E-01	*4.08E-01*	6.15E-01
f_{CEC09}	Mean	3.74E+00	1.91E+01	1.01E+03	*2.44E+00*	2.51E+02	3.81E+00	7.35E+01	3.26E+03	2.74E+00
	SD	6.10E-01	2.89E+01	5.55E+02	*5.31E-02*	1.77E+02	5.67E-01	1.86E+02	7.50E+02	2.30E-01
f_{CEC10}	Mean	1.99E+01	2.04E+01	2.02E+01	*1.92E+01*	2.00E+01	1.97E+01	2.06E+01	2.07E+01	2.02E+01
	SD	1.71E+00	6.70E-01	*7.93E-02*	4.86E+00	2.28E+00	3.49E+00	9.34E-02	1.19E-01	9.52E-02

Note: **Highlighted and italic data reveal the best performing results among the compared algorithms.

Figure 4 Sample of convergence curves of AgtrO and other algorithms on $f_{CEC01–CEC10}$ (see online version for colours)



4.2 CEC-C06-2019 benchmark

Table 3 collects the statistical results of AgtrO and eight state-of-the-art comparative algorithms on ten modern CEC-C06-2019 benchmark functions. On a total of ten modern CEC-C06-2019 optimisation benchmark functions, AgtrO ranked 1st on f_{CEC01} and f_{CEC06}, ranked 2nd on $f_{CEC02–CEC03}$, and ranked 3rd on $f_{CEC04–CEC05}$ and $f_{CEC08–CEC10}$. In overall, AgtrO could rank at least 3rd out of 9 benchmark functions and has never ranked below 5th place among all compared algorithms. Despite such an excellent performance ranking, AgtrO could only be the second best among all algorithms. Combining the fact that AgtrO could rank first only on very few benchmark functions, while EO was slightly superior to AgtrO on majority functions, it has been recognised that EO has a better solution on the CEC-C06-2019 benchmark. Through an in-depth analysis of the EO mechanism, we found that EO has the following physics-based properties: non-selective execution, refined computation, and specialised adaptive methods that can convincingly address these benchmark functions. AgtrO, on the other hand, adopts selective rather than non-selective execution, where the algorithm selects either equation for computation, resulting in AgtrO being less competitive than EO on the CEC-C06-2019 benchmark. Although AgtrO has better clustering properties, it does not have much impact on this benchmark type. Anyway, upon analytical discussion, we still claim that AgtrO performs well on CEC-C06-2019 benchmark functions, as AgtrO has achieved outstanding mean and SD results compared to other algorithms.

We also inspect the convergence speed of AgtrO on $f_{CEC01–CEC10}$ as compared with other algorithms. Figure 4 displays the convergence curves of all algorithms on each CEC-C06-2019 benchmark function. By observation, the convergence graph of AgtrO could significantly curve downward in early iterations. AgtrO did not achieve the fastest convergence on any single function, but in general, it has a highly competitive convergence speed, as it has achieved at least the 3rd fastest convergence speed on every benchmark function, except for f_{CEC10}.

4.3 Results analysis and discussion

For an in-depth discussion, AgtrO has been rationally analysed to determine its practical significance, advantages, and shortcomings. The easiest method to inspect the practical significance is through hypothesis tests. From the excellent statistical results, it can be pointed out that AgtrO is practically significant against other compared algorithms, as the differences between the means are large enough to show absolute superiority. The most convincing evidence can be found in the results on $f_{1–4}$, $f_{9–11}$, f_{14}, $f_{16–17}$, $f_{21–23}$, $f_{CEC01–CEC03}$ and f_{CEC06}.

As a novel bio-inspired meta-heuristic optimisation algorithm, AgtrO has been shown able to solve a variety of unconstrained benchmark problems. The distinctive strength of AgtrO is that it assigns two operational equations that refer to two different target points: equation (1) mainly refers to the global best solution during operation, playing a major role in dealing with the global search type. In contrast, equation (12) mainly refers to the local best solution during the operation, playing a major role in dealing with the local search type. Both equations have their own purpose. Thus, AgtrO can provide explicit commands to search agents through either phase of execution. Interleaved execution between the two phases [i.e., equations (1) and (12)] takes advantage of both search types.

However, AgtrO has a minor weakness. AgtrO converges slower than certain comparative algorithms. The search agent has a chance to select either equation (1) or (12) to execute, and each equation serves to approach a problem only from its local perspective. Two equations are formulated to specialise different purposes for good performance, of which only Equation (1) is responsible for accelerating convergence. This halves the convergence ability of the algorithm in exchange for better global-best achievement.

5 Conclusions

This research work proposed a novel bio-inspired meta-heuristic algorithm, called AgtrO algorithm. Inspired by the natural ecology of alligators, AgtrO was formulated to execute two main phases: hunting phase and relocating phase. The hunting phase mimics fishing, purse seining and catching behaviours in mathematical terms, and the relocating phase mimics travelling and homing instinct behaviours in mathematical terms. A comprehensive assessment was conducted on 23 classical benchmark functions and ten modern CEC-C06-2019 benchmark functions. When comparing statistical results with eight state-of-the-art comparative algorithms, AgtrO achieved best performance on classical benchmark functions and superior performance on modern CEC-06-2019 benchmark functions in terms of global-best achievement. Upon evaluation, AgtrO obtained satisfactory convergence speed on every benchmark. Overall, AgtrO outperformed other algorithms on a majority of classical and CEC10-2019 benchmark functions. Therefore, it can be concluded that AgtrO is a well-performed optimisation algorithm that strikes a proper balance between exploration and exploitation.

In recent application trends, optimising efficiency and processing speed should be the focus. AgtrO needs to face the future challenges to further minimise the convergence efficiency and execution time. In this research work, it was decided to utilise only the hunting and relocating phases when building a basic AgtrO, but alligators have other operations that are worth implementing in modelling, such as breeding and infant feeding mechanisms. The further implementation of evolutionary mechanisms will be the subject of the future.

References

Abdullah, J.M. and Rashid, T.A. (2019) 'Fitness dependent optimizer: inspired by the bee swarming reproductive process', *IEEE Access*, Vol. 7, pp.43473–43486, DOI: 10.1109/ACCESS.2019.2907012.

Abualigah, L., Diabat, A. et al. (2021a) 'The arithmetic optimization algorithm', *Computer Methods in Applied Mechanics and Engineering*, Vol. 376, p.113609, DOI: 10.1016/j.cma.2020.113609.

Abualigah, L., Yousri, D. et al. (2021b) 'Aquila optimizer: a novel meta-heuristic optimization algorithm', *Computers & Industrial Engineering*, Vol. 157, No. 11, p.107250, DOI: 10.1016/j.cie.2021.107250.

Agushaka, J.O. and Ezugwu, A.E. (2021) 'Advanced arithmetic optimization algorithm for solving mechanical engineering design problems', *PLoS One*, Vol. 16, No. 8, p.e0255703, DOI: 10.1371/journal.pone.0255703. eCollection 2021.

Basturk, B. and Karaboga, D. (2007) 'A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm', *Journal of Global Optimization*, Vol. 39, No. 3, pp.459–471, DOI: 10.1007/s10898-007-9149-x.

Bensaoucha, S. (2021) 'Breast cancer diagnosis using optimized machine learning algorithms', in *2021 International Conference on Recent Advances in Mathematics and Informatics (ICRAMI)*, IEEE, Tebessa, Algeria, DOI: 10.1109/ICRAMI52622.2021.9585977.

Beyer, H-G. and Schwefel, H-P. (2002) 'Evolution strategies – a comprehensive introduction', *Natural Computing*, Vol. 1, pp.3–52, DOI: 10.1023/A:1015059928466.

Birbil, Ş.İ. and Fang, S-C. (2003) 'An electromagnetism-like mechanism for global optimization', *Journal of Global Optimization*, Vol. 25, pp.263–282, DOI: 10.1023/A: 1022452626305.

Burdova, D.V. and Tikhonova, K.B. (2021) 'Optimization of the territorial planning system based on the formation of integrated information systems with a single geospace', *IOP Conf. Series: Earth and Environmental Science*, IOP Publishing Ltd., Vol. 937, No. 042069, pp.1–6, DOI: 10.1088/1755-1315/937/4/042069.

Colomi, A., Dorigo, M. and Maniezzo, V. (1991) 'Distributed optimization by ant colonies', in *Proceedings of the first European Conference on Artificial Life*, Paris, France, pp.134–142.

Deng, X. and Lv, T. (2020) 'Power system planning with increasing variable renewable energy: a review of optimization models', *Journal of Cleaner Production*, Vol. 246, p.118962, DOI: 10.1016/j.jclepro.2019.118962.

Dhiman, G. and Kumar, V. (2018) 'Emperor penguin optimizer: a bio-inspired algorithm for engineering problems', *Knowledge-Based Systems*, Vol. 159, pp.20–50, DOI: 10.1016/j.knosys.2018.06.001.

Dhiman, G. and Kumar, V. (2019) 'Seagull optimization algorithm: theory and its applications for large-scale industrial engineering problems', *Knowledge-Based Systems*, Vol. 165, pp.169–196, DOI: 10.1016/j.knosys.2018.11.024.

Digalakis, J.G. and Margaritis, K.G. (2002) 'An experimental study of benchmarking functions for genetic algorithms', *International Journal of Computer Mathematics*, Vol. 79, No. 4, pp.403–416, DOI: 10.1080/00207160210939.

Dinets, V. (2015) 'Apparent coordination and collaboration in cooperatively hunting crocodilians', *Ethology Ecology & Evolution*, Vol. 27, No. 2, pp.244–250, DOI: 10.1080/ 03949370.2014.915432.

Dou, Y. and Meng, W. (2021) 'An optimization algorithm for computer-aided diagnosis of breast cancer based on support vector machine', *Frontiers in Bioengineering and Biotechnology*, Vol. 9, No. 698390, DOI: 10.3389/fbioe. 2021.698390.

Eberhrt, R. and Kennedy, J. (1995) 'A new optimizer using particle swarm theory', in *Proceeding of the 6th International Symposium on Micro Machine and Human Science*, pp.39–43, DOI: 10.1109/MHS.1995.494215.

Eiben, A.E. and Smith, J.E. (2003) 'Evolutionary programming', in *Introduction to Evolutionary Computing*. Natural Co. Springer, Berlin, Heidelberg, pp.89–99, DOI: 10.1007/978-3-662-05094-1_5.

Elekidis, A.P., Koltsaklis, N.E. and Georgiadis, M.C. (2018) 'An optimization approach for the assessment of the impact of transmission capacity on electricity trade and power systems planning', *Ind. Eng. Chem. Res.*, Vol. 57, No. 30, pp.9766–9778, DOI: doi.org/10.1021/acs.iecr.7b05159.

Faramarzi, A. et al. (2020) 'Equilibrium optimizer: a novel optimization algorithm', *Knowledge-Based Systems*, Vol. 191, pp.1–21, DOI: 10.1016/j.knosys.2019.105190.

Fujisaki, I. et al. (2014) 'Home range and movements of American alligators (Alligator mississippiensis) in an estuary habitat', *Animal Biotelemetry*, Vol. 2, No. 8, DOI: 10.1186/2050-3385-2-8.

Gomes, G.F., Cunha, S.S. and Ancelotti, A. (2019) 'A sunflower optimization (SFO) algorithm applied to damage identification on laminated composite plates', *Engineering with Computers*, Vol. 35, pp.619–626, DOI: 10.1007/s00366-018-0620-8.

Guo, C. (2021) 'Application of computer technology in optimal design of overall structure of special machinery', *Mathematical Problems in Engineering*, Vol. 2021, Article ID 6619485, pp.1–9, DOI: 10.1155/2021/6619485.

Hayyolalam, V. and Kazem, A.A.P. (2020) 'Black widow optimization algorithm: a novel meta-heuristic approach for solving engineering optimization problems', *Engineering Applications of Artificial Intelligence*, Vol. 87, DOI: 10.1016/j.engappai.2019.103249.

Heidari, A.A. et al. (2019) 'Harris hawks optimization: algorithm and applications', *Future Generation Computer Systems*, Vol. 97, pp.849–872, DOI: 10.1016/j.future.2019.02.028.

Holland, J.H. (1975) *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, Illustrate, University of Michigan Press, Ann Arbor, Michigan.

Kallioras, N.A., Lagaros, N.D. and Avtzis, D.N. (2018) 'Pity beetle algorithm – a new metaheuristic inspired by the behavior of bark beetles', *Advances in Engineering Software*, Vol. 121, pp.147–166, DOI: 10.1016/j.advengsoft.2018. 04.007.

Kaveh, A. and Kooshkebaghi, M. (2019) 'Artificial coronary circulation system: a new bio-inspired metaheuristic algorithm', *SCIENTIA IRANICA: International Journal of Science and Technology*, Vol. 26, No. 5, pp.2731–2747, DOI: 10.24200/SCI.2019.21366.

Kaveh, A. and Mahjoubi, S. (2018) 'Lion pride optimization algorithm: a meta-heuristic method for global optimization problems', *SCIENTIA IRANICA: International Journal of Science and Technology*, Vol. 25, No. 6, pp.3113–3132, DOI: 10.24200/SCI.2018.20833.

Kodama, T. et al. (2021) 'Algorithm for an automatic treatment planning system using a single-arc VMAT for prostate cancer', *J. Appl. Clin. Med. Phys.*, Vol. 22, No. 12, pp.27–36, DOI: 10.1002/acm2.13442.

Lainas, G.T. et al. (2020) 'A decision-making algorithm for performing or cancelling embryo transfer in patients at high risk for ovarian hyperstimulation syndrome after triggering final oocyte maturation with hCG', *Human Reproduction Open*, Vol. 2020, No. 3, p.hoaa013, DOI: 10.1093/hropen/hoaa013.

Lance, V.A. et al. (2011) 'Long-distance movement by American Alligators in Southwest Louisiana', *Southeastern Naturalist*, Vol. 10, No. 3, pp.389–398, DOI: 10.1656/058.010.0301.

Lee, T-H. et al. (2021) 'Optimal design of a synchronous reluctance motor using a genetic topology algorithm', *Processes*, Vol. 9, No. 10, p.1778, DOI: 10.3390/pr9101778.

Leiber, D. et al. (2022) 'Simulation-based layout optimization for multi-station assembly lines', *Journal of Intelligent Manufacturing*, Vol. 33, pp.537–554, DOI: 10.1007/s10845-021-01853-5.

Li, H. et al. (2021a) 'Cognitive electronic jamming decision-making method based on improved Q-learning algorithm', *International Journal of Aerospace Engineering*, Vol. 2021, Article ID 8647386, pp.1–12, DOI: 10.1155/2021/8647386.

Li, H.C. et al. (2021b) 'A decision optimization method for emergency maintenance of regional rail transit based on genetic algorithm', in *ICCMS '21: 2021 The 13th International Conference on Computer Modeling and Simulation*, pp.185–194, DOI: 10.1145/3474963.3474989.

Li, X.L., Shao, Z.J. and Qian, J.X. (2002) 'An optimizing method based on autonomous animate: fish-swarm algorithm', *Chinese Journal of Systems Engineering-Theory & Practice*, Vol. 22, No. 11, pp.32–38, DOI: 10.12011/1000-6788(2002)11-32.

Li, Y. and Deng, X. (2020) 'An efficient algorithm for Elastic I-optimal design of generalized linear models', *Canadian Journal of Statistics*, Vol. 49, No. 2, pp.438–470, DOI: 10.1002/cjs.11571.

Luo, Y., Zhang, L. and Song, R. (2021) 'Optimized lung tumor diagnosis system using enhanced version of crow search algorithm, Zernike moments, and support vector machine', in *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, DOI: 10.1177/09544119211055870.

Mahmood, M. and Al-Khateeb, B. (2019) 'The blue monkey: a new nature inspired metaheuristic optimization algorithm', *Periodicals of Engineering and Natural Sciences*, Vol. 7, No. 3, pp.1054–1066, DOI: 10.21533/pen.v7i3.621.

Pierezan, J. and Coelho, L.D.S. (2018) 'Coyote optimization algorithm: a new metaheuristic for global optimization problems', in *2018 IEEE Congress on Evolutionary Computation (CEC)*, pp.1–8, DOI: 10.1109/CEC.2018.8477769.

Price, K.V. et al. (2018) *The 100-Digit Challenge: Problem Definitions and Evaluation Criteria for the 100-Digit Challenge Special Session and Competition on Single Objective Numerical Optimization*, Nanyang Technological University, Singapore [online] https://github.com/P-N-Suganthan/CEC2019 (accessed 1 July 2019).

Qin, J., Li, M. and Liang, Y. (2022) 'Minimum cost consensus model for CRP-driven preference optimization analysis in large-scale group decision making using Louvain algorithm', *Information Fusion*, Vol. 80, pp.121–136, DOI: 10.1016/j.inffus.2021.11.001.

Rana, P. and Varshney, L.R. (2021) 'Trustworthy predictive algorithms for complex forest system decision-making', *Frontiers in Forests and Global Change*, Vol. 3, Article 587178, pp.1–15, DOI: 10.3389/ffgc.2020.587178.

Rashedi, E., Nezamabadi-pour, H. and Saryazdi, S. (2009) 'GSA: a gravitational search algorithm', *Information Sciences*, Vol. 179, No. 13, pp.2232–2248, DOI: 10.1016/j.ins.2009.03.004.

Rodda, G.H. (1984) 'Homeward paths of displaced juvenile alligators as determined by radiotelemetry', *Behavioral Ecology and Sociobiology*, Vol. 14, pp.241–246, DOI: 10.1007/BF00299494.

Rosenblatt, A.E. and Heithaus, M.R. (2011) 'Does variation in movement tactics and trophic interactions among American alligators create habitat linkages?', *Journal of Animal Ecology*, Vol. 80, No. 4, pp.786–798, DOI: 10.1111/j.1365-2656.2011.01830.x.

Saremi, S., Mirjalili, S. and Lewis, A. (2017) 'Grasshopper optimisation algorithm: theory and application', *Advances in Engineering Software*, Vol. 105, pp.30–47, DOI: 10.1016/j.advengsoft.2017.01.004.

Shah-Hosseini, H. (2011) 'Principal components analysis by the galaxy-based search algorithm: a novel metaheuristic for continuous optimisation', *International Journal of Computational Science and Engineering*, Vol. 6, Nos. 1–2, DOI: 10.1504/IJCSE.2011.041221.

Storn, R. and Price, K. (1997) 'Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces', *Journal of Global Optimization*, Vol. 11, No. 4, pp.341–359, DOI: 10.1023/A:1008202821328.

Tharwat, A. and Gabel, T. (2020) 'Parameters optimization of support vector machines for imbalanced data using social ski driver algorithm', *Neural Computing and Applications*, Vol. 32, pp.6925–6938, DOI: 10.1007/s00521-019-04159-z.

Triska, Y. et al. (2021) 'Integrated production and maintenance planning method with simulation-based optimization', *IFAC-PapersOnLine*, Vol. 54, No. 1, pp.349–354, DOI: 10.1016/j.ifacol.2021.08.160.

Vanneschi, L. and Poli, R. (2012) 'Genetic programming – introduction, applications, theory and open issues', in Rozenberg, G., Bäck, T. and Kok, J.N. (Eds.): *Handbook of Natural Computing*, Springer, Berlin, Heidelberg, pp.709–739, DOI: 10.1007/978-3-540-92910-9_24.

Yang, X.-S. (2010) 'Firefly algorithm, stochastic test functions and design optimisation', *International Journal of Bio-Inspired Computation*, Vol. 2, No. 2, pp.78–84, DOI: 10.1504/IJBIC.2010.032124.

Yao, X., Liu, Y. and Lin, G. (1999) 'Evolutionary programming made faster', *IEEE Transactions on Evolutionary Computation*, Vol. 3, No. 2, pp.82–102, DOI: 10.1109/4235.771163.

Zhao, W., Wang, L. and Zhang, Z. (2019) 'Atom search optimization and its application to solve a hydrogeologic parameter estimation problem', *Knowledge-Based Systems*, Vol. 163, pp.283–304, DOI: 10.1016/j.knosys.2018.08.030.

Appendix

Figure A1 Phase-by-phase description of the fishing mechanism of i^{th} agent in AgtrO

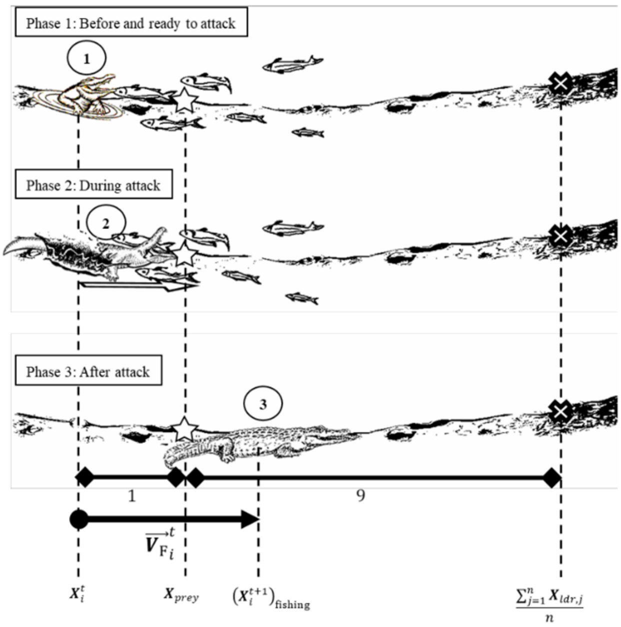


Figure A2 Purse seining mechanism in AgtrO

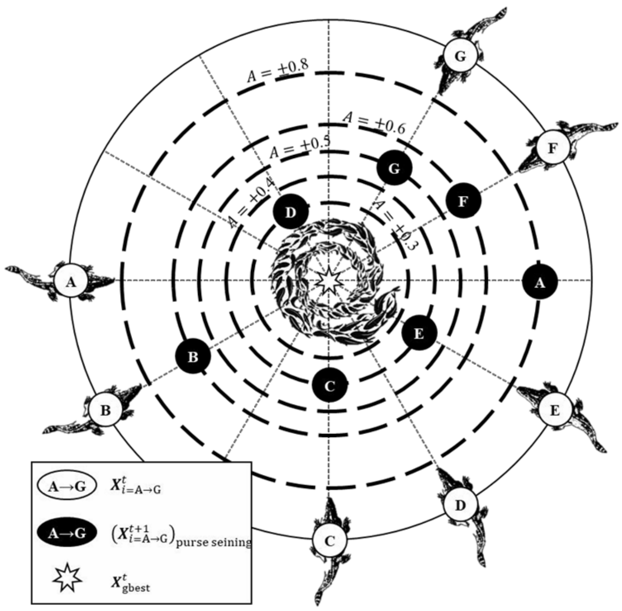


Figure A3 Catching mechanism by agent **A** in AgtrO (see online version for colours)

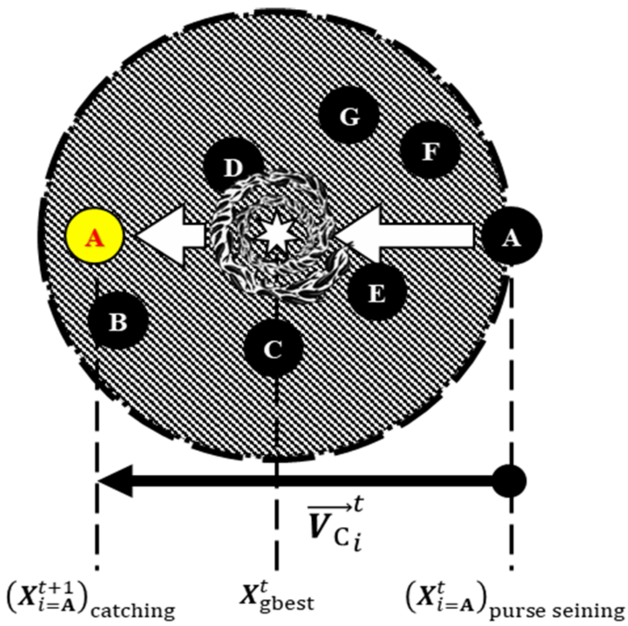


Figure A4 Flow chart of proposed AgtrO algorithm