Available online at www.ijournalse.org

Emerging Science Journal

(ISSN: 2610-9182)

Vol. 8, No. 4, August, 2024

A Hybrid Ant Colony and Grey Wolf Optimization Algorithm for Exploitation-Exploration Balance

Joan Angelina Widians [1, 2], Retantyo Wardoyo [1*], Sri Hartati [1]

[1] *Department of Computer Science and Electronics, Faculty of Mathematics and Natural Science, Gadjah Mada University, Yogyakarta 55281, Indonesia.*

[2] *Department of Informatics, Faculty of Engineering, Mulawarman University, Samarinda 75123 , Indonesia.*

Abstract

The Ant Colony Optimization (ACO) and Grey Wolf Optimizer (GWO) are well-known nature-inspired algorithms. ACO is a metaheuristic search algorithm that takes inspiration from the behavior of real ants. In contrast, GWO is a grey wolf population-based heuristic algorithm. The important procedure in optimization is exploration and exploitation. ACO has excellent global and local search capabilities, and the exploration process is performed better than the exploitation process. In the case of regular, GWO is a greatly competitive algorithm compared to other common meta-heuristic algorithms, as it has super performance in the exploitation phase. This study proposed hybrid ACO and GWO algorithms. This hybridization is to acquire the balance between exploitation and exploration in optimization Swarm Intelligence algorithm—comprehensive examination using CEC 2014 benchmark functions. Detail investigations indicate that ACO-GWO could find solutions to unimodal, multi-modal, and hybrid problems in evaluation functions. The results show that the ACO-GWO algorithm outperforms its predecessors in several benchmark function cases. In addition, the proposed ACO-GWO algorithm could achieve an exploitation-exploration balance. Even though ACO-GWO has one disadvantage: since ACO-GWO directly combines two algorithms (ACO and GWO) with two different agents, it has superior demands on computational complexity.

Keywords:

Ant Colony Optimization;
Grey Wolf Optimizer;
Swarm Intelligence;
Exploitation-Exploration Balance;
Optimization.

Article History:

Received:	01	April	2024
Revised:	27	June	2024
Accepted:	09	July	2024
Published:	01	August	2024

1- Introduction

Swarm Intelligence (SI) has developed very quickly since the 2010s. Thousands of scientific papers on theory and application have been published, increasing significantly. In terms of terms, a swarm is a group, collection, herd, colony, troupe, or anything numerous, collective, and massive. SI could be interpreted as collective intelligence or group intelligence resulting from the behavior of groups or agents. SI is a scientific discipline that deals with natural and artificial systems composed of many individuals who coordinate using decentralized control and can organize themselves [1]. In SI, collective intelligence only emerges from simple agents or individuals, and this intelligence will not arise if one great agent dominates simple agents. For example, an ant does not have extraordinary intelligence. However, thousands of ants in a colony, who communicate and coordinate well, could quickly find the shortest path between their food source and their nest. This balanced cooperation produces extraordinary intelligence. What would happen if, in the colony, there was an ant that was much smarter than all of them? The existence of domination will result in collective intelligence decreasing or never appearing. The SI system could carry out actions in a coordinated way without an external controller. The behavior of each individual is described probabilistically; each individual has behavior that depends on local perceptions of the individuals who are their neighbors.

* **CONTACT**: rw@ugm.ac.id

DOI: http://dx.doi.org/10.28991/ESJ-2024-08-04-023

Swarm builds a powerful communication system to help the system achieve various types of complex goals, such as laying webs, finding food, traveling from one place to another, and living safely in dangerous environments. Therefore, a structure that is managed effectively and can be understood is needed. To maintain a solid coordination system, swarm uses two phenomena: exploration and exploitation. Exploration is the process of gathering new information, while exploitation is the process of using existing information to improve coordination. Exploration and exploitation are two prominent concepts in SI techniques for solving optimization problems. Exploration (diversification) refers to the ability of an algorithm to search for promising areas across the search space at large. In contrast, exploitation (intensification) refers to the ability of an algorithm to perform focused local searches on promising regions discovered in the exploration process. These processes are used in optimization to target global and local solutions. Flashy, a balance between exploration and exploitation is essential to achieve optimal solutions to optimization-related problems.

In SI, several parameters can be adjusted to produce a high level of adaptively. Good parameter settings can balance the exploitation and exploration behavior of individual herds. Exploitation is the behavior of a group of individuals to find solutions in a particular area. Thus, exploitation focuses on the direction of search. On the other side, exploration is the behavior of a group of individuals to search for solutions randomly in various locations. In short, exploration is the spread of search direction. In principle, exploration and exploitation must be balanced to avoid local optimums and ensure that SI always reaches the global optimum. When should SI carry out exploitation and exploration? For example, in the SI algorithm, certain measures can be used to measure the variance or standard deviation of the fitness values of all individuals. However, it can also be done by increasing the speed parameters of all particles so that they spread (exploration) into distant solution spaces (increasing variance) in the hope of finding better new solutions.

A quality search algorithm is one that has a balance between exploration and exploitation, which is critical [2]. Hybrid strategies use two or more algorithms to balance exploitation and exploration. It motivates us to suggest combining two algorithms, namely, *ant colony optimization* (ACO) and *grey wolf optimization* (GWO). In this study, we propose an algorithm that combines ACO and GWO. ACO algorithms are meta-heuristics that aim to find approximate solutions to optimization problems. The ACO was first introduced by Marco Dorigo in 1991 in his PhD thesis, later published under Ant System. The behavior of ant colonies inspires this algorithm. This animal is simple, even weak-looking. However, if many ants work together in a large colony, they have extraordinary abilities. Ant colonies could find the shortest path from the nest to the food source. This concept is used in the Traveling Salesman Problem (TSP). In the real world, TSP sends packages, designs flight routes, and more [3].

In contrast, Mirjalili et al. proposed the GWO algorithm in 2014 [4]. Develop the GWO algorithm based on inspiration from the behavior of the Grey Wolf (*Canis lupus*), particularly its hunting techniques and social hierarchies. Grey wolves are considered apex predators, meaning they are at the top of the food chain. Grey wolves mostly prefer to live in packs. The average group size is five to twelve individuals [5]. The process of predation is a process of finding the optimal solution. From an optimization concept, GWO is more relied on for exploitation [6]. It motivates us to combine ACO and GWO to solve optimization problems more efficiently. The proposed hybrid ACO-GWO combines the best characteristics of the ACO and GWO. The hybridization of these two optimization processes is expected to increase the rate of exploration and exploitation, avoiding premature convergence of algorithmic processes. This paper is organized into several sections. The related works in Section 2 and the proposed approach are described in Section 3. The experimental setup is presented in Section 4. Finally, the conclusion is summarized in Section 5.

2- Related Works

The SI algorithm is built on population-based algorithms initially found in nature. Some systems studied in SI are ant colonies, bees, fireflies, bats, and more. SI delves into the collective social behavior of various animal groups, such as birds flying in the sky in formations or bird flocking, groups of land animals that move in specific rules or herds of land animals, hawks hunting, or schools of fish swimming in groups or fish schooling. Most SI lies in understanding the behavior and lifestyle of these groups, as well as the interactions among their members, particularly in locating food sources. Natural mechanisms suggest various techniques for addressing optimization issues. These algorithms prioritize studying swarm members' dynamics and interactions to optimize tasks [7].

ACO is included in the SI algorithm, one type of algorithm used to solve optimization problems. Within ACO, individual ants within the colony move about, depositing pheromones along their paths. The pheromone is a signal to fellow ants. These pheromones serve as communication signals among the ants. Routes with the shortest distances attract a more potent pheromone signal. As more ants traverse a particular route, the pheromone signal along that path intensifies. This concept is used in TSP, which seeks the shortest path to a solution [8]. One of the notable things in ACO is that when the system has found an optimal solution, ACO will be able to adapt quickly to changes that occur around it. This adaptation is based on pheromones. A solution with an optimal path will possess a stronger pheromone.

Moreover, ACO has been successfully utilized in many types of research to select the optimization. Most ACO methods represent features as graph nodes with edges denoting components. The ants explore these points to find a suitable path [9, 10]. ACO was also merged with Cuckoo Search for FS in the digital mammographic dataset [11]. Hybrid ACO and PSO for feature selection performance have achieved the best result in terms of classification performance [8].

The GWO algorithm replicates the behaviors observed in grey wolves within the natural world. Within this algorithm, the population is categorized into four groups: alpha, beta, delta, and omega. These groups are organized according to their respective fitness levels, arranged in ascending order [12]. The top three individuals and the remaining individuals are referred to as omega. A simple approach to thinking about a solution involves approximating the optimal solution's location using the top-performing members of the population [5]. GWO has the potential for enhancement through the implementation of a dimension learning-oriented hunting strategy, which uses a different approach to construct a neighborhood for each wolf to enhance the balance of local and global searches while preserving diversity-attempts to do this by combining swarm intelligence algorithms with GWO [13]. Hybrid GWO and Whale optimization algorithm (WOA) for solving pressure vessel design [14] This hybridization leverages the strengths of both algorithms to enhance the optimization process, leading to improved designs and potentially significant cost savings in engineering applications. Hybrid GWO and Grasshopper Optimization Algorithm (GOA) have also been proposed for solving the feature selection and clustering problem in machine learning [15].

One popular hybrid approach combines GWO and Particle Swarm Optimization (PSO). This study proposed that the GWOPSO acquire the balance between exploration and exploitation. GWO is due to the high exploration and perfect exploitation achieved by PSO. This hybrid is for feature selection with seventeen UCI machine learning repository datasets [2]. Another hybridization involves GWO and CS (Cuckoo Search). An augmentation of GWO, named Augmented GWO (AGWO), was recently proposed and possesses more excellent exploration abilities. The CS (Cuckoo Search) algorithm is a nature-inspired optimizing technique miming cuckoo birds' and levy-flights' unique nesting strategies. Both algorithms possess powerful searching capabilities. The proposed algorithm amalgamates the exploring abilities of the AGWO with the exploiting abilities of the CS [16]. Additionally, reasonably balancing exploration and exploitation will improve the search algorithm's performance. The other study merged a binary hybrid GWO with Harris Hawks Optimization (HHO) to achieve a good balance. They used GWO for exploration and HHO for exploitation [17].

The hybrid algorithms utilized GWO and Elephant Herding Optimization (EHO). The main idea is to integrate the strength of GWO in exploitation and the ability of EHO in exploration to avoid getting trapped in local optima. In this algorithm, the clan life of elephants is used to group wolves so that the wolf population is divided into a certain number of clans [18]. A recently developed metaheuristic hybrid PSO and GWO is implemented for optimization. This algorithm is a powerful fusion of PSO's exploitation and GWO's exploration properties [19].

GWO is a developed stochastic meta-heuristic technique motivated by nature. It replicates grey wolf hunting behavior and social hierarchy, exploring the solution space similar to their natural process. GWO efficiently explores and converges to the optimal solution and has limited exploitation capability. To address this, the GWO-Employed-Onlooker model suggests incorporating the onlooker and scout bee operators from the artificial bee colony algorithm (ABC) during the position-changing stage of the grey wolves. This algorithm enhances exploitation capability, improving local convergence rates and solution quality [20].

In conclusion, combining GWO with different optimization techniques has great promise in improving GWO performance. Hybridization is often used to overcome the limitations of individual algorithms, such as the balance between exploration and exploitation. Apart from that, it also avoids the problem of premature convergence and being trapped in local optima.

3- Research Method

3-1- Ant Colony Optimization Algorithm

Ant colonies inspire the ACO algorithm. Ants have no eyesight. How could they find the shortest path between a food source and their nest? This paradigm is based on ethologists' observations about the media ants use to communicate information about the shortest path to food via pheromone pathways. When looking for food, the ants randomly wander around the nest area. Once they know there is food, the ants will analyze the quantity and quality of food and take some of it back to their nest. On their journey, they leave behind pheromones that will guide their friends to find food sources [21]. The amount of pheromone left behind depends on the amount of food found. The more food the ants get, the more pheromones they leave behind, so the more ants that pass through a path, the stronger the pheromone trail that collects on that path [22]. The ACO algorithm has excellent global and local search capabilities and various versions, most of which could be applied to optimization. More importantly, the process of exploration in the ACO algorithm is better than the exploitation process. The ACO algorithm could be applied to optimization problems based on five aspects, namely: appropriate problem representation, desirability heuristic, solution construction mechanism, pheromone update rule, and probabilistic transition rule [23].

ACO transition and pheromone update regulations could be implemented based on this graph representation restructuring. The pheromone and heuristic attributes are not linked to the connection in this scenario. Instead, each attribute possesses its own pheromone and heuristic values. Several artificial ants are employed to construct a solution progressively for optimization dilemmas. During each cycle, an ant will allocate an amount of pheromone corresponding to the solution's effectiveness. At every stage, each Ant calculates a series of feasible expansions to a partial solution and chooses one based on two factors: local heuristics and previous experience [24].

Figure 1 illustrates the step of ACO representation in optimization. Any node could be chosen as the next option because nodes are fully connected [24]. Graphical representation in ACO uses transitions and pheromone update rules [9]. The ants will traverse the nodes, aiming to find the optimal feature for the subset, until the termination condition is met. The fundamental components of the ACO algorithm involve constructive heuristics for constructing solutions probabilistically. These constructive heuristics formulate a solution by arranging elements from a limited set of solution components. The process of solution construction commences with an initially empty partial solution [25]. Artificial ants follow the rules called probabilistic transition rules that determine the probability of a feature k ants selecting i feature to be the solution at time t. The probabilistic transition rule is to establish two parameters, namely heuristic information and pheromone level, as shown in Equation 1 [8].

$$p_{ij}^{k}(t) = \begin{cases} \frac{[\tau_i(t)]^A[\eta_i]^B}{\sum_{j\in J^k}[\tau_j(t)]^A[\eta_j]^B}, & if\ i \in J^k \\ 0 & otherwise \end{cases} \tag{1}$$

J^k is the set of the ant-k's unvisited features, ηi is the heuristic desirability of element i. Where τ_i (t) is the pheromone value at feature-i. While η_j is the heuristic value of element-j, and τ_j (t) is the pheromone value of feature-*j*. Simultaneously, parameters A and B play crucial roles in defining the significance of both pheromone levels and heuristic information. Once all the ants have completed the solution, the pheromones produced during the tour should be administered. In pheromone management, there are two primary processes: pheromone evaporation and storage. These processes revolve around a central procedure known as pheromone renewal. Evaporation of pheromones serves the purpose of preventing Ants from following identical paths, thus diversifying solutions. Moreover, Ants can adjust pheromone levels at locations they visit. The most successful Ant, having found the optimal solution, earns the privilege of storing more pheromones than others [26].

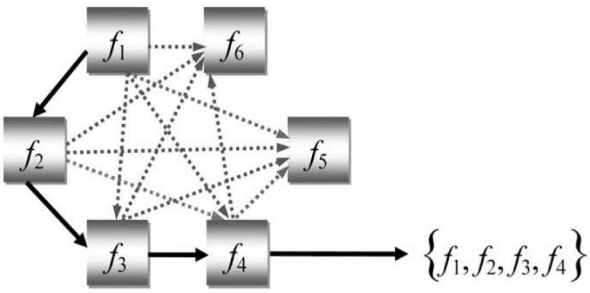


Figure 1. ACO Representation

3-2- Grey Wolf Optimizer Algorithm

The GWO algorithm draws its inspiration primarily from the intelligence, leadership, and hunting instincts observed in the grey wolf species in their natural habitat. Grey wolves, also known as timber or western wolf, always live in form packs consisting of approximately 5 to 11 individuals. Notably, they exhibit a well-defined social hierarchy, as illustrated in Figure 2. The wolf social hierarchy consists of four groups: alpha wolf, beta wolf, delta wolf, and omega wolf. The update of the wolf position relies mainly on learning from leader individuals. Alpha wolf (α) is a decision-maker and commander. Beta (β) assists the alpha in collective leadership. Delta (δ) follows the instructions of alpha and beta. The remaining wolves are omega (ω) and monitor the other wolves' movements [27].

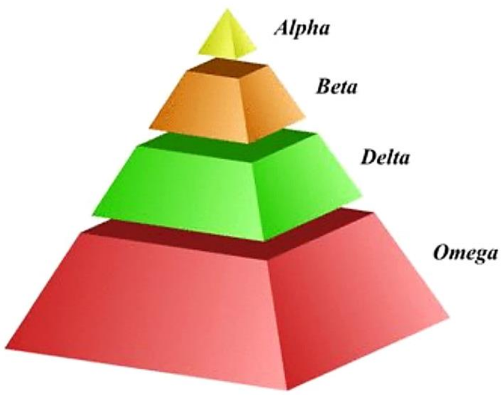


Figure 2. Social Hierarchy of Grey Wolves

The second level in the hierarchy is Beta. Beta Wolves assist Alphas in decision-making and group activities, reinforcing their commands and offering feedback. The lowest level in the grey wolf group is Omega, who acts as a victim. Wolves at this level are obliged to obey the orders of other wolves who are higher in the hierarchy, and they are not allowed to eat food until other wolves in other groups have eaten. Despite appearing insignificant, Omegas play a crucial role in identifying internal conflicts and other issues within the pack. They bear the responsibility of exposing cruelty and discontent among wolves, thus maintaining group cohesion and order. Omega keeps the other wolves satisfied and also preserves the grey wolf's central organization. Occasionally, omega wolves serve as babysitters in the pack [28, 29].

Delta Wolves must comply with Alpha and Beta, but they dominate Omega. Omega wolves include scouts, hunters and guards. Scouts monitor territory boundaries and warn the group of any danger. Hunters assist Alpha when hunting prey and preparing food for the group. Ultimately, the keepers care for the pack’s weak, sick and injured wolves [30].

Figure 2 illustrates the social structure of grey wolves. Within this hierarchy, the top-ranking position is denoted by alpha (α), followed by beta (β) and delta (δ) as the second and third-best solutions, respectively. Another solution is designated as omega (ω). In the GWO algorithm, hunting or optimization is led by the alpha, beta, and delta wolves, while omega wolves trail behind them [31].

The second source of inspiration for GWO is how the grey wolf hunts. Pursuit, encirclement, harassment, and assault are used to conclude the hunting process. The GWO begins with an initial population generated at random and is modified during iterations. Exploration is investigating the solution space for novel and potentially superior alternatives. It entails performing random or diverse actions to identify new regions of the solution space [4].

The process of exploration and exploitation represents two contrasting approaches that an algorithm may adopt in problem optimization. Exploration involves the algorithm's attempt to uncover new facets of the problem's search space by implementing sudden alterations to the solution. Its objective is to identify promising areas within the search landscape and prevent solutions from becoming entrenched in local optima. Conversely, exploitation involves thoroughly examining the current best solution or the most promising regions of solution space to find the optimal solution. Its goal is to enhance the already best-available solution and further refine it. Subsequently, the fitness function is evaluated, with Equation 2 utilized to characterize the behavior of wolves during hunting [12]. The main goal of exploitation involves enhancing the assessment of solutions achieved through exploration by uncovering the surrounding area of each solution. Incremental adjustments to the solution are necessary for reaching the overall best outcome. The key difficulty lies in managing the tension between exploitation and exploration. Hence, an algorithm needs to navigate and reconcile these conflicting behaviors during optimization to estimate the global optimum for a specific problem precisely [2].

Grey wolves typically scan their surroundings for potential prey by tracking the positions of α, β, and δ. They disperse to explore areas where prey might be located, then regroup to launch coordinated attacks. In this scenario, let's consider $A^{\rightarrow}$ as a random vector ranging from -1 to 1, guiding the search agents to move away from the prey, thus emphasizing the global search aspect in GWO. Additionally, this algorithm incorporates an extra component ($C^{\rightarrow}$) to facilitate the generation of new solutions [32].

$$\begin{aligned} \vec{D} &= |\vec{C}\cdot\vec{X}_p(t)-\vec{X}(t)| \\ \vec{X}(t+1) &= \vec{X}_p(t)-\vec{A}\cdot\vec{D} \end{aligned} \tag{2}$$

where t is current literacy, and vector coefficient, is the preposition, and position of grey wolf is vectors are calculated using Equation 3.

$$\begin{aligned} \vec{A} &= 2\vec{a}\cdot\vec{r}_1-\vec{a} \\ \vec{C} &= 2\cdot\vec{r}_2 \end{aligned} \tag{3}$$

where $\vec{a}$ is the vector coefficient, $\vec{r}_1$, $\vec{r}_2$ are the random number [0,1]. It is known that the value decreases linearly from 2 to 0 during the iteration.

Based on Equation 3, the component in GWO that supports exploration is vector C. The GWO algorithm could explore more of the search space randomly, thus allowing search agents to avoid getting trapped in local optima during the optimization process. Meanwhile, the decrease in vector C is nonlinear. The vector C values are assigned randomly during iteration to improve the global search in the decision space and prevent the search space from moving far within the local optimum [2].

The primary factor governing exploration in GWO is the variable C. This parameter consistently generates a random value within the range of [0, 2]. It alters the influence of the prey in determining the subsequent position, with a stronger effect when C is greater than 1, causing the solution to gravitate more towards the prey. Regardless of the iteration number, C provides random values, thereby prioritizing exploration during optimization, particularly in instances of stagnation in local optima. Another influential parameter in encouraging exploration is A. Its value is determined based on a, which linearly decreases from 2 to 0. Due to its random nature, the A fluctuates within the interval to [-2, 2]. Exploration is facilitated when A exceeds 1 or falls below -1, while emphasis is placed on exploitation when -1 < A < 1.

In a similar vein, grey wolves possess the ability to manoeuvre within an n-dimensional decision space, akin to nodes of a hypercube, in close proximity to the optimal solution, which represents the prey's location. They demonstrate an ability to discern the prey's location amidst others and strategically encircle it. Typically, the hunting process is guided by alpha (α) and beta (β), with delta (δ) providing support to alpha. Thus, in emulating the stalking behavior of grey wolves, it is assumed that alpha, beta, and delta are more aware of the probable directions of the prey. Consequently, GWO retains the three most favourable solutions attained, necessitating omega wolves to adjust their positions to attain the optimal location in the decision space. This approach enables the package to converge towards the prey by updating positions based on the best alpha, beta, and delta locations as per Equation 2 [2].

$$
\begin{aligned}
\vec{D}_\alpha &= \left|\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}\right| \\
\vec{D}_\beta &= \vec{C}_2 \cdot \vec{X}_\beta - \vec{X}| \\
\vec{D}_\delta &= |\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}| \\
\vec{X}_1 &= \vec{X}_\alpha - \vec{A}_1 \cdot (\vec{D}_\alpha) \\
\vec{X}_2 &= \vec{X}_\beta - \vec{A}_2 \cdot (\vec{D}_\beta) \\
\vec{X}_3 &= \vec{X}_\delta - \vec{A}_3 \cdot (\vec{D}_\delta)
\end{aligned}
\tag{4}
$$

The position of grey wolf alpha is $\vec{X}_\alpha$, $\vec{X}_\beta$ is the position of beta and $\vec{X}_\delta$ is the position of delta wolf [5].

3-3- ACO-GWO Algorithm

The hybrid ACO-GWO algorithm represents a sophisticated approach that harnesses the strengths of two powerful optimization techniques: Ant Colony Optimization (ACO) and Grey Wolf Optimizer (GWO). By integrating these methods, the algorithm aims to achieve a balance between exploration and exploitation, thereby enhancing its ability to efficiently search for optimal solutions in complex problem spaces.

In the initial phase, the ACO component takes charge of exploration. This phase involves the construction of a diverse set of candidate solutions by leveraging the concept of pheromone trails, mimicking the foraging behavior of ants. Each ant, representing a potential solution, probabilistically selects solution components based on the intensity of pheromone trails associated with those components. These selected components are then combined to form candidate solutions, enabling the algorithm to explore a wide range of possibilities within the search space.

Following the exploration phase, the GWO algorithm steps in to refine the candidate solutions and converge towards the optimal solution. Here, the candidate solutions generated by ACO are evaluated using an objective function, which quantifies their fitness. The GWO algorithm then adjusts the positions of the alpha, beta, and delta wolves—key agents in the optimization process—based on the evaluation results. This adjustment mechanism facilitates the exploration of promising regions in the solution space while gradually converging towards the most favorable solution.

The iterative nature of the algorithm ensures continuous improvement, with each iteration aiming to enhance the quality of solutions. This iterative process continues until predefined stopping criteria are met, such as reaching the maximum number of generations specified for the algorithm. Throughout this process, the algorithm dynamically adapts its exploration and exploitation strategies, guided by the interplay between ACO's exploration capabilities and GWO's exploitation prowess.

The flowchart in Figure 3 depicting the ACO-GWO hybrid algorithm provides a visual representation of its sequential execution, illustrating the interplay between the ACO and GWO components in the pursuit of optimal solutions. By leveraging the complementary strengths of these two optimization techniques, the hybrid ACO-GWO algorithm offers a robust and versatile approach for addressing complex optimization problems across various domains.

ACO-GWO algorithm is designed to leverage the strengths of both algorithms and provide a more effective and efficient optimization approach. The use of ACO for candidate solution construction enables the algorithm to explore the search space and generate diverse solutions. ACO is used as the exploration agent due to its effective mechanism of discovering new paths based on pheromone trails and heuristic information. The pheromone update step is also important for adapting the search behavior of ants and influencing the overall search process. Without sufficient exploration, an algorithm might quickly converge on a solution that appears optimal within a limited region of the search space but is actually suboptimal in the global context. The exploratory nature of ACO helps in broadening the search scope, increasing the chances of identifying the true global optimum.

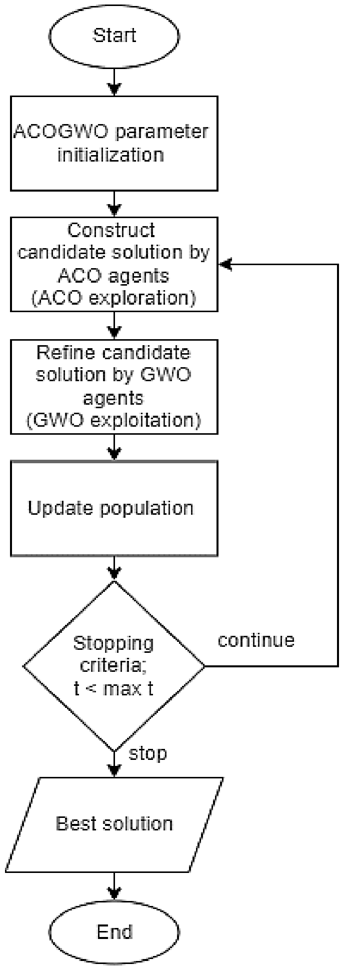


Figure 3. Flowchart of ACO-GWO

While exploration is crucial, too much of it can lead to inefficiency, as the algorithm might spend excessive time evaluating suboptimal regions. This is where GWO's exploitation capabilities become essential, allowing the algorithm to intensify the search around the best solutions and converge more rapidly and accurately. The use of GWO for solution refinement enables the algorithm to exploit promising regions of the search space and find the optimal solution. GWO is used as the exploitation agent, leveraging its leadership hierarchy and social behaviors to refine and intensify the search around the best solutions found. This structured approach allows the algorithm to fine-tune solutions with high precision, maximizing the likelihood of finding the global optimum. The ACO-GWO hybrid effectively combines these complementary strengths to achieve a superior balance between exploration; searching new areas within the solution space and exploitation; focusing the search around the best-found solutions. This balance is vital in optimization as it prevents the algorithm from getting trapped in local optima while ensuring thorough examination of the solution space.

4- Results and Discussion

This section examines the performance of ACO, GWO, Random Walk GWO (RW-GWO), and the proposed ACO-GWO using the criteria outlined in the CEC2014 benchmark functions. These benchmarks encompass 30 unconstrained optimization problems categorized into unimodal (F1-F3), multimodal (F4-F16), hybrid (F17-F22), and composite (F23-F30) types. The detailed benchmark functions can be seen in Table 1. All experiments are conducted in Python 3.11. Each test function is run for 30 dimensions to evaluate the performance of both algorithms. The search space for each variable range from -100 to 100. *Termination criteria* are defined as the maximum function evaluations (104 * 30). Testing is carried out across all algorithms, with each run spanning 3000 epochs and ten independent processes to obtain the average objective function value solution. The parameters for ACO-GWO are consistent with ACO, with Q set to 0.5 and pheromone decay set to 0.5. The population parameter for all algorithms is fixed at 30 and adjusted according to the number of dimensions.

Table 2 shows the results of the ACO-GWO algorithm and competitors. Numbers in bold indicate the best results among all competitors. The unimodal problem is suitable for evaluating the exploitability of all search algorithms.

Functions F_1-F_3 are unimodal in the CEC 2014 benchmark problem set. Thus, regarding exploiting the search regions around the explored search regions, ACO-GWO is better than others in Functions F_2-F_3. Functions F_4-F_{16} are multimodal; the multi-modal test problem typically assesses the exploration strength and local optima avoidance capability. In functions F_4, F_7, F_8, F_9, F_{10}, and F_{15} ACO-GWO could outperform others. Hybrid and composite issues serve to assess the efficacy of overcoming stagnation issues amidst a plethora of local optima while also gauging the capacity to strike a balance between exploration and exploitation within meta-heuristic algorithms. The problems F_{17}-F_{22} are hybrid functions, and F_{23}-F_{30} problems are composite functions. ACO-GWO performs better in functions F_{17}, F_{22}, F_{23}, F_{24}, and F_{26}. In other functions, ACO-GWO could get the minimum values, such as in functions F_6, F_{11}, F_{18}, F_{19}, F_{28}, and F_{29}. Proves the power of exploitation and exploration in ACO-GWO if several experiments are carried out to find the best. So overall, ACO-GWO could overcome unimodal, multimodal, hybrid and composite problems (see Table 2).

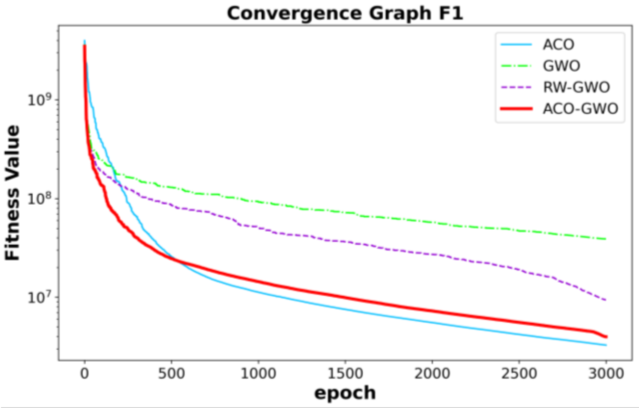
Table 1. Summary of the CEC2014 Benchmark Functions

Problem Type	No.	Functions	$F_i^* = F_i(x^*)$
Unimodal	F1	Rotated high conditioned elliptic function	100
	F2	Rotated Bent Cigar Function	200
	F3	Rotated Discus Function	300
Simple Multimodal	F4	Shifted and Rotated Rosenbrock's Function	400
	F5	Shifted and Rotated Ackley's Function	500
	F6	Shifted and Rotated Weierstrass Function	600
	F7	Shifted and Rotated Griewank's Function	700
	F8	Shifted Rastrigin's Function	800
	F9	Shifted and Rotated Rastrigin's Function	900
	F10	Shifted Schwefel's Function	1000
	F11	Shifted and Rotated Schwefel's Function	1100
	F12	Shifted and Rotated Katsuura Function	1200
	F13	Shifted and Rotated HappyCat Function	1300
	F14	Shifted and Rotated HGBat Function	1400
	F15	Shifted and Rotated Expanded Griewank's plus Rosenbrock's Function	1500
	F16	Shifted and Rotated Expanded Scaffer's F6 Function	1600
Hybrid Function	F17	Hybrid Function 1 (N=3)	1700
	F18	Hybrid Function 2 (N=3)	1800
	F19	Hybrid Function 3 (N=4)	1900
	F20	Hybrid Function 4 (N=4)	2000
	F21	Hybrid Function 5 (N=5)	2100
	F22	Hybrid Function 6 (N=5)	2200
Composition	F23	Composition Function 1 (N=5)	2300
	F24	Composition Function 2 (N=3)	2400
	F25	Composition Function 3 (N=3)	2500
	F26	Composition Function 4 (N=5)	2600
	F27	Composition Function 5 (N=5)	2700
	F28	Composition Function 6 (N=5)	2800
	F29	Composition Function 7 (N=3)	2900
	F30	Composition Function 8 (N=3)	3000
Search Range: $[-100, 100]^{30}$			

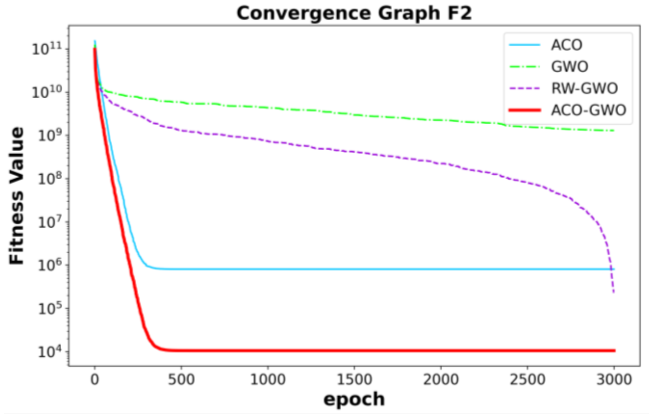
Table 2. Average, Maximum, and Minimum Error Value Obtained by Algorithms for 30-dimensional CEC2014 Benchmark

Function	ACO			GWO			RW-GWO			ACO-GWO		
	mean	max	min	mean	max	min	mean	max	min	mean	max	min
1	3.28E+06	7.12E+06	1.20E+06	3.91E+07	1.12E+08	8.98E+06	9.43E+06	1.54E+07	3.84E+06	3.98E+06	8.60E+06	1.25E+06
2	8.11E+05	7.98E+06	2.89E+03	1.30E+09	5.31E+09	5.07E+07	2.35E+05	4.14E+05	1.23E+05	**1.06E+04**	**2.55E+04**	**6.06E+02**
3	1.33E+04	3.03E+04	9.18E+02	1.36E+04	2.72E+04	6.97E+03	3.80E+03	5.40E+03	2.17E+03	**3.38E+03**	**1.11E+04**	**6.94E+02**
4	4.96E+02	5.38E+02	4.67E+02	6.45E+02	1.05E+03	5.30E+02	5.13E+02	5.70E+02	4.69E+02	**4.82E+02**	**5.37E+02**	**4.03E+02**
5	5.21E+02	5.21E+02	5.21E+02	5.21E+02	5.21E+02	5.21E+02	5.20E+02	5.20E+02	5.20E+02	5.21E+02	5.21E+02	5.21E+02
6	6.02E+02	6.05E+02	6.01E+02	6.11E+02	6.16E+02	6.07E+02	6.15E+02	6.21E+02	6.11E+02	6.03E+02	6.08E+02	**6.01E+02**
7	7.00E+02	7.00E+02	7.00E+02	7.06E+02	7.14E+02	7.02E+02	7.01E+02	7.01E+02	7.00E+02	**7.00E+02**	**7.00E+02**	**7.00E+02**
8	8.11E+02	8.16E+02	8.08E+02	8.58E+02	8.73E+02	8.43E+02	8.54E+02	8.75E+02	8.36E+02	8.15E+02	8.22E+02	8.09E+02
9	1.07E+03	1.08E+03	1.05E+03	1.01E+03	1.12E+03	9.70E+02	1.01E+03	1.04E+03	9.82E+02	**1.01E+03**	1.08E+03	**9.15E+02**
10	2.59E+03	6.77E+03	1.14E+03	3.06E+03	6.67E+03	1.96E+03	2.18E+03	2.63E+03	1.37E+03	**1.44E+03**	**2.40E+03**	**1.03E+03**
11	8.05E+03	8.58E+03	6.80E+03	5.96E+03	8.94E+03	3.32E+03	3.96E+03	4.98E+03	3.24E+03	4.85E+03	8.37E+03	**2.27E+03**
12	1.20E+03	1.20E+03	1.20E+03	1.20E+03	1.20E+03	1.20E+03	1.20E+03	1.20E+03	1.20E+03	1.20E+03	1.20E+03	1.20E+03
13	1.30E+03	1.30E+03	1.30E+03	1.30E+03	1.30E+03	1.30E+03	1.30E+03	1.30E+03	1.30E+03	1.30E+03	1.30E+03	1.30E+03
14	1.40E+03	1.40E+03	1.40E+03	1.40E+03	1.40E+03	1.40E+03	1.40E+03	1.40E+03	1.40E+03	1.40E+03	1.40E+03	1.40E+03
15	1.52E+03	1.52E+03	1.51E+03	1.52E+03	1.56E+03	1.51E+03	1.51E+03	1.52E+03	1.51E+03	**1.51E+03**	**1.52E+03**	**1.50E+03**
16	1.61E+03	1.61E+03	1.61E+03	1.61E+03	1.61E+03	1.61E+03	1.61E+03	1.61E+03	1.61E+03	1.61E+03	1.61E+03	1.61E+03
17	4.72E+05	8.54E+05	1.96E+05	4.92E+05	1.15E+06	7.99E+04	3.27E+05	9.69E+05	7.29E+04	**1.69E+05**	**5.31E+05**	**6.44E+04**
18	6.43E+05	3.94E+06	6.88E+04	9.58E+04	1.35E+05	6.86E+04	1.05E+05	1.64E+05	7.65E+04	1.02E+05	1.52E+05	**5.54E+04**
19	2.74E+03	5.36E+03	1.93E+03	4.32E+03	1.46E+04	1.92E+03	2.25E+03	3.55E+03	1.92E+03	2.42E+03	6.52E+03	**1.92E+03**
20	5.77E+04	9.05E+04	2.37E+04	1.47E+05	2.34E+05	9.04E+04	1.54E+05	2.62E+05	6.90E+04	6.49E+04	1.23E+05	2.85E+04
21	2.76E+04	5.34E+04	1.03E+04	6.39E+05	2.53E+06	4.38E+04	2.57E+05	9.16E+05	3.88E+04	2.87E+04	**4.96E+04**	1.29E+04
22	3.26E+03	4.60E+03	2.75E+03	4.41E+03	6.78E+03	3.29E+03	3.31E+03	4.12E+03	2.97E+03	**2.99E+03**	**3.54E+03**	**2.67E+03**
23	2.53E+03	2.54E+03	2.52E+03	2.53E+03	2.53E+03	2.52E+03	2.52E+03	2.52E+03	2.52E+03	**2.52E+03**	**2.52E+03**	**2.52E+03**
24	2.64E+03	2.64E+03	2.62E+03	2.60E+03	2.60E+03	2.60E+03	2.60E+03	2.60E+03	2.60E+03	**2.60E+03**	**2.60E+03**	**2.60E+03**
25	2.71E+03	2.72E+03	2.71E+03	2.71E+03	2.72E+03	2.71E+03	2.71E+03	2.71E+03	2.70E+03	2.71E+03	2.72E+03	2.71E+03
26	2.73E+03	2.90E+03	2.70E+03	2.77E+03	2.94E+03	2.70E+03	2.78E+03	2.92E+03	2.70E+03	**2.73E+03**	**2.80E+03**	**2.70E+03**
27	3.11E+03	3.15E+03	3.04E+03	3.30E+03	3.45E+03	3.20E+03	3.34E+03	3.46E+03	3.11E+03	3.14E+03	3.23E+03	3.08E+03
28	3.68E+03	3.76E+03	3.61E+03	3.96E+03	4.66E+03	3.68E+03	3.93E+03	4.28E+03	3.71E+03	3.73E+03	3.89E+03	**3.53E+03**
29	1.44E+04	2.84E+04	7.42E+03	1.67E+06	6.45E+06	8.59E+03	1.82E+04	4.65E+04	7.00E+03	2.51E+04	1.66E+05	**5.30E+03**
30	4.77E+05	1.63E+06	1.65E+04	1.30E+07	2.94E+07	1.71E+06	4.06E+05	2.55E+06	3.54E+04	1.15E+06	7.99E+06	6.16E+04

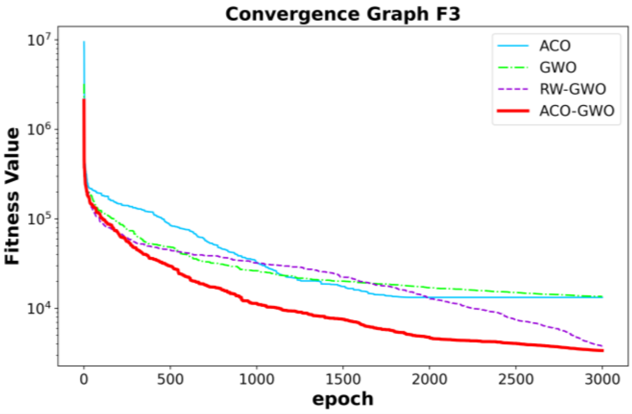
Furthermore, a detailed evaluation of the convergence curve is carried out. First, the convergence curves for unimodal functions are examined. Figure 4 illustrates the convergence behavior of four algorithms: ACO, GWO, RW-GWO, and ACO-GWO. The vertical axis represents the fitness values, while the horizontal axis represents the generation (or epoch) values. Specifically, Figure 4-b highlights that the ACO-GWO algorithm demonstrates a faster convergence towards the optimum value for function F2 within the first 500 epochs. Additionally, as depicted in Figure 4-c, the ACO-GWO algorithm achieves the lowest score compared to the other algorithms on unimodal problems, indicating its superior performance in quickly finding the optimal solution in simpler, unimodal landscapes. Next, the convergence curves for multimodal functions are analyzed. Figures 4-d to 4-f present the convergence curves of all the algorithms for functions F4, F9, and F10, respectively. The results show that the ACO-GWO algorithm consistently achieves the best fitness values compared to the other algorithms in these multimodal problems. This demonstrates the algorithm's effectiveness in navigating complex landscapes with multiple local optima. The ACO-GWO hybrid not only finds the best solutions but also does so more rapidly, demonstrating a faster approach towards the global optimum. Finally, the convergence analysis is conducted for hybrid and composite multimodal functions, which are more challenging due to their combination of different characteristics. Figures 4-g and 4-h plot the convergence curves for functions F17 and F26, respectively. The ACO-GWO algorithm achieves the lowest scores in these cases as well, outperforming the other algorithms. This indicates its robustness and versatility in handling highly complex optimization problems, where it successfully combines the exploratory power of ACO with the exploitative strength of GWO to consistently reach the best possible solutions.



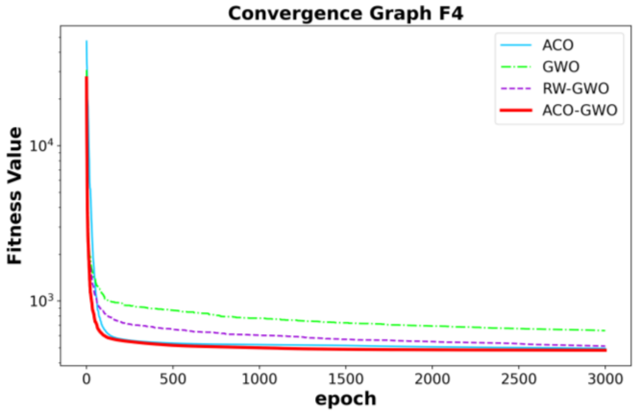
(a) Convergence Graph for F1



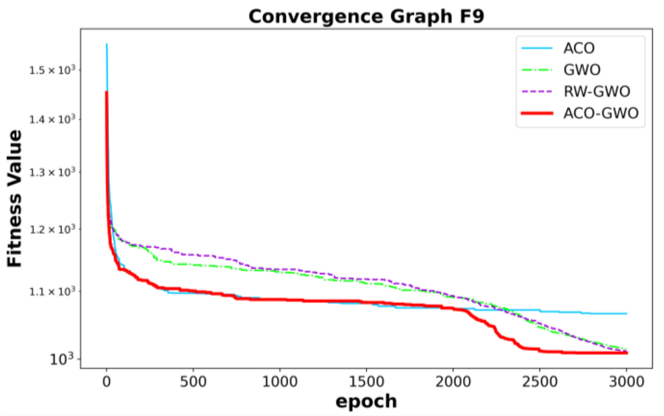
(b) Convergence Graph for F2



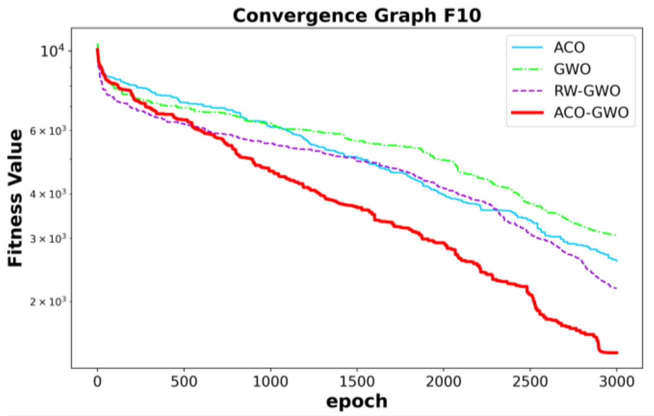
(c) Convergence Graph for F3



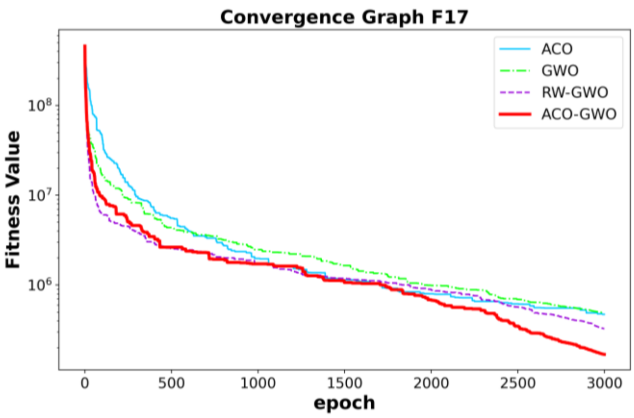
(d) Convergence Graph for F4



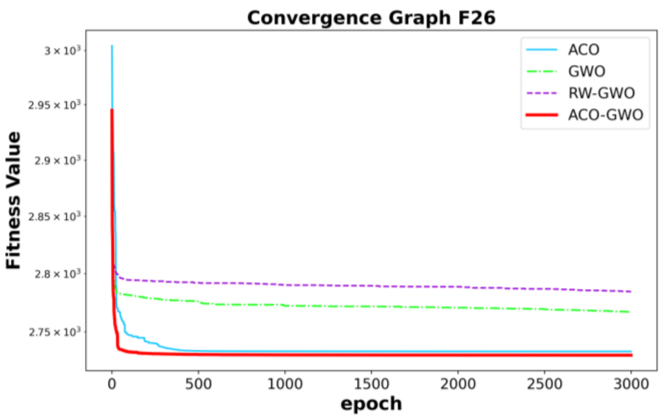
(e) Convergence Graph for F9



(f) Convergence Graph for F10



(g) Convergence Graph for F17



(h) Convergence Graph for F26

Figure 4. Convergence Graph of Certain Functions

Hence, all those convergence curves indicate that ACO-GWO is the most efficient metaheuristic algorithm among its predecessors. ACO-GWO saw in the graph a similarity to GWO and RW-GWO algorithms' convergence rate, but ACO-GWO could search better towards an optimum solution. The combination of ACO as the first initiation of search agent and GWO as the second agent for exploitation in ACOGWO could provide an exploitation–exploration balance.

5- Conclusion

This paper presents a solution to a significant issue of optimization. Hybrid algorithms have gained popularity for handling optimization issues. The proposed solution involves a hybrid algorithm, Ant Colony Optimization (ACO) and Grey Wolf Optimizer (GWO). The main idea is to integrate GWO's strengths in exploitation and ACO's capabilities in exploration so as not to get trapped in local optima. Maintaining this balance of exploitation and exploration makes the speed and accuracy of the convergence of the proposed algorithm. This research uses public benchmarks from CEC 2014. Thirty benchmark mathematical functions for unimodal, multimodal, and hybrid problems in evaluation functions are used to validate the proposed ACOGWO compared to the original GWO, original ACO, and RWGWO algorithm. The results show that the ACO-GWO algorithm outperforms its predecessors in several benchmark function cases. In addition, the proposed ACO-GWO algorithm could achieve an exploitation-exploration balance. Even though ACO-GWO has one disadvantage: since ACO-GWO directly combines two algorithms (ACO and GWO) with two different agents, it has superior demands on computational complexity. Therefore, future research is needed to modify the ACO-GWO algorithm or employ other strategies to lower computational demands.

6- Declarations

6-1- Author Contributions

Conceptualization, J.A.W.; preparation of introduction, J.A.W.; preparation of related work, J.A.W. and S.H.; data collection and analysis, R.W. and J.A.W.; preparation of the pre-processing, R.W. and J.A.W.; compilation of theory and methodology, R.W. and J.A.W.; validation of theory and methodology, R.W. and S.H; compilation of theory and methods of accuracy measurement, R.W. and J.A.W.; preparation of results and discussion, J.A.W.; drawing conclusion, R.W. and J.A.W. All authors have read and agreed to the published version of the manuscript.

6-2- Data Availability Statement

Publicly available datasets were analyzed in this study. Public benchmark from CEC 2014. In addition, this benchmark set consists of 30 unconstrained optimization problems of unimodal (F_1-F_3), multimodal (F_4-F_{16}), hybrid (F_{17}-F_{22}) and composite (F_{23}-F_{30}) types.

6-3- Funding

Gadjah Mada University funded this research with the RTA Grant Number 5075/UN1.P.II/Dit-Lit/PT.01.01/2023.

6-4- Acknowledgements

This study would not have been possible without the support of The Gadjah Mada University with the Final Project Recognition Grant 5075/UN1.P.II/Dit-Lit/PT.01.01/2023.

6-5- Institutional Review Board Statement

Not applicable.

6-6- Informed Consent Statement

Not applicable.

6-7- Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this manuscript. In addition, the ethical issues, including plagiarism, informed consent, misconduct, data fabrication and/or falsification, double publication and/or submission, and redundancies have been completely observed by the authors.

7- References

[1] Brezočnik, L., Fister, I., & Podgorelec, V. (2018). Swarm intelligence algorithms for feature selection: A review. Applied Sciences (Switzerland), 8(9), 1521. doi:10.3390/app8091521.

[2] El-Kenawy, E. S., & Eid, M. (2020). Hybrid grey wolf and particle swarm optimization for feature selection. International Journal of Innovative Computing, Information and Control, 16(3), 831–844. doi:10.24507/ijicic.16.03.831.

[3] Mirjalili, S. (2019). Ant colony optimization. Studies in Computational Intelligence, 780(November), 33–42. doi:10.1007/978-3-319-93025-1_3.

[4] Faris, H., Aljarah, I., Al-Betar, M. A., & Mirjalili, S. (2018). Grey wolf optimizer: a review of recent variants and applications. Neural Computing and Applications, 30(2), 413–435. doi:10.1007/s00521-017-3272-5.

[5] Shen, C., & Zhang, K. (2022). Two-stage improved Grey Wolf optimization algorithm for feature selection on high-dimensional classification. Complex & Intelligent Systems, 8(4), 2769–2789. doi:10.1007/s40747-021-00452-4.

[6] Sharma, R., Marikkannu, P., & Sungheetha, A. (2019). Three-dimensional MRI brain tumour classification using hybrid ant colony optimization and Grey Wolf optimiser with proximal support vector machine. International Journal of Biomedical Engineering and Technology, 29(1), 34–45. doi:10.1504/IJBET.2019.096879.

[7] Kyaw, K. S., & LIMSIRORATANA, S. (2019). Traditional and Swarm Intelligent Based Text Feature Selection for Document Classification. 2019 19th International Symposium on Communications and Information Technologies (ISCIT), Ho Chi Minh City, Vietnam. doi:10.1109/iscit.2019.8905200.

[8] Menghour, K., & Souici-Meslati, L. (2016). Hybrid ACO-PSO based approaches for feature selection. International Journal of Intelligent Engineering and Systems, 9(3), 65–79. doi:10.22266/ijies2016.0930.07.

[9] Ghosh, M., Guha, R., Sarkar, R., & Abraham, A. (2020). A wrapper-filter feature selection technique based on ant colony optimization. Neural Computing and Applications, 32(12), 7839–7857. doi:10.1007/s00521-019-04171-3.

[10] Wang, Z., Gao, S., Zhou, M., Sato, S., Cheng, J., & Wang, J. (2023). Information-Theory-based Nondominated Sorting Ant Colony Optimization for Multiobjective Feature Selection in Classification. IEEE Transactions on Cybernetics, 53(8), 5276–5289. doi:10.1109/tcyb.2022.3185554.

[11] Jona, J. B., & Nagaveni, N. (2014). Ant-cuckoo colony optimization for feature selection in digital mammogram. Pakistan Journal of Biological Sciences, 17(2), 266–271. doi:10.3923/pjbs.2014.266.271.

[12] Wang, H., Hu, Z., Yang, Z., & Guo, Y. (2020). A Novel Grey Wolf Optimization Based Combined Feature Selection Method. Bio-inspired Computing: Theories and Applications, BIC-TA 2019, Communications in Computer and Information Science, vol 1159, Springer, Singapore. doi:10.1007/978-981-15-3425-6_45.

[13] Nadimi-Shahraki, M. H., Taghian, S., & Mirjalili, S. (2021). An improved grey wolf optimizer for solving engineering problems. Expert Systems with Applications, 166, 113917. doi:10.1016/j.eswa.2020.113917.

[14] Mohammed, H., & Rashid, T. (2020). A novel hybrid GWO with WOA for global numerical optimization and solving pressure vessel design. Neural Computing and Applications, 32(18), 14701–14718. doi:10.1007/s00521-020-04823-9.

[15] Purushothaman, R., Rajagopalan, S. P., & Dhandapani, G. (2020). Hybridizing Grey Wolf Optimization (GWO) with Grasshopper Optimization Algorithm (GOA) for text feature selection and clustering. Applied Soft Computing Journal, 96, 106651. doi:10.1016/j.asoc.2020.106651.

[16] Sharma, S., Kapoor, R., & Dhiman, S. (2021). A Novel Hybrid Metaheuristic Based on Augmented Grey Wolf Optimizer and Cuckoo Search for Global Optimization. 2021 2nd International Conference on Secure Cyber Computing and Communications (ICSCCC). doi:10.1109/icsccc51823.2021.9478142.

[17] Al-Wajih, R., Abdulkadir, S. J., Aziz, N., Al-Tashi, Q., & Talpur, N. (2021). Hybrid binary grey Wolf with Harris hawks optimizer for feature selection. IEEE Access, 9, 31662–31677. doi:10.1109/ACCESS.2021.3060096.

[18] Hoseini, Z., Varaee, H., Rafieizonooz, M., & Kim, J. H. J. (2022). A New Enhanced Hybrid Grey Wolf Optimizer (GWO) Combined with Elephant Herding Optimization (EHO) Algorithm for Engineering Optimization. Journal of Soft Computing in Civil Engineering, 6(4), 1–42. doi:10.22115/SCCE.2022.342360.1436.

[19] Bhandari, A. S., Kumar, A., & Ram, M. (2023). Reliability optimization and redundancy allocation for fire extinguisher drone using hybrid PSO–GWO. Soft Computing, 27(20), 14819–14833. doi:10.1007/s00500-023-08560-8.

[20] Ahmad, I., Qayum, F., Rahman, S. U., & Srivastava, G. (2024). Using Improved Hybrid Grey Wolf Algorithm Based on Artificial Bee Colony Algorithm Onlooker and Scout Bee Operators for Solving Optimization Problems. International Journal of Computational Intelligence Systems, 17(1), 111. doi:10.1007/s44196-024-00497-6.

[21] Dadaneh, B. Z., Markid, H. Y., & Zakerolhosseini, A. (2016). Unsupervised probabilistic feature selection using ant colony optimization. Expert Systems with Applications, 53, 27–42. doi:10.1016/j.eswa.2016.01.021.

[22] Janaki Meena, M., Chandran, K. R., Karthik, A., & Vijay Samuel, A. (2012). An enhanced ACO algorithm to select features for text categorization and its parallelization. Expert Systems with Applications, 39(5), 5861–5871. doi:10.1016/j.eswa.2011.11.081.

[23] Moradi, P., & Rostami, M. (2015). Integration of graph clustering with ant colony optimization for feature selection. Knowledge-Based Systems, 84, 144–161. doi:10.1016/j.knosys.2015.04.007.

[24] Widians, J. A., Wardoyo, R., & Hartati, S. (2022). A Study on Text Feature Selection Using Ant Colony and Grey Wolf Optimization. 2022 Seventh International Conference on Informatics and Computing (ICIC), Bali, Indonesia. doi:10.1109/icic56845.2022.10007019.

[25] Ma, W., Zhou, X., Zhu, H., Li, L., & Jiao, L. (2021). A two-stage hybrid ant colony optimization for high-dimensional feature selection. Pattern Recognition, 116, 107933. doi:10.1016/j.patcog.2021.107933.

[26] Paniri, M., Dowlatshahi, M. B., & Nezamabadi-pour, H. (2020). MLACO: A multi-label feature selection algorithm based on ant colony optimization. Knowledge-Based Systems, 192, 105285. doi:10.1016/j.knosys.2019.105285.

[27] Singh, N., & Singh, S. B. (2017). Hybrid Algorithm of Particle Swarm Optimization and Grey Wolf Optimizer for Improving Convergence Performance. Journal of Applied Mathematics, 2017, 1–15. doi:10.1155/2017/2030489.

[28] Chantar, H., Mafarja, M., Alsawalqah, H., Heidari, A. A., Aljarah, I., & Faris, H. (2020). Feature selection using binary grey wolf optimizer with elite-based crossover for Arabic text classification. Neural Computing and Applications, 32(16), 12201–12220. doi:10.1007/s00521-019-04368-6.

[29] Aktaş, M., & Kılıç, F. (2021). Binary grey wolf optimizer using archeology and astronomy news for text classification. International Conference on Innovative Engineering Applications (CIEA), 20-22 May, 2021, Muş, Turkey.

[30] Yousef, J., Youssef, A., & Keshk, A. (2021). A Hybrid Swarm Intelligence Based Feature Selection Algorithm for High Dimensional Datasets. IJCI. International Journal of Computers and Information, 8(1), 67–86. doi:10.21608/ijci.2021.62499.1040.

[31] Hu, P., Pan, J. S., & Chu, S. C. (2020). Improved Binary Grey Wolf Optimizer and Its application for feature selection. Knowledge-Based Systems, 195, 105746. doi:10.1016/j.knosys.2020.105746.

[32] Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey Wolf Optimizer. Advances in Engineering Software, 69, 46–61. doi:10.1016/j.advengsoft.2013.12.007.