JOURNAL OF HARBIN ENGINEERING UNIVERSITY
ISSN: 1006-7043

Vol 44 No. 7
July 2023

Exploring the Potential of Invasive Weed Optimization: A Population-Based Metaheuristic for Optimization Problems

[1]Sudha K, [2]M Suresh, [3]R.M. Mallika, [4]Aravabhumi Divya, [5]Kuruma Purnima, [6]Ethireddy Sasikala

[1]Associate Professor, Dept of Computer Science and Business Systems, R.M.D. Engineering College, Kavaraipettai-601206
[2]Professor & Principal, Department of ME, Gokula Krishna College of Engineering, Sullurpet
[3]Associate Professor, Department of CSE, Siddharth Institute of Engineering & Technology, Puttur, Tirupati (Dt)
[4]Assistant Professor, Department of CSE, N B K R Institute of Science and Technology, Vidyanagar
[5]Associate Professor, Department of CSE, Gokula Krishna College of Engineering, Sullurpet
[6]Assistant Professor, Department of ECE, Gokula Krishna College of Engineering, Sullurpet

Abstract: Invasive Weed Optimization (IWO) is a population-based metaheuristic algorithm inspired by the invasive behaviour of weeds. It aims to solve optimization problems through a process of competition, reproduction, mutation, and selection. Introduced by Mehrabian and Lucas in 2006, IWO has since undergone several iterations and improvements. The algorithm begins by initializing a population of candidate solutions, represented as "weeds." Each weed is evaluated based on a fitness function that captures the optimization criteria of the problem. Through reproduction, fitter individuals are selected as parents, and new offspring are generated through recombination and crossover operations. Mutation introduces random changes to the offspring, allowing for exploration of the search space. Competition plays a crucial role in IWO. Offspring compete with existing weeds, and if an offspring exhibits superior fitness, it replaces a less fit weed in the population. This competitive mechanism ensures that stronger solutions survive and propagate, gradually improving the quality of solutions over iterations. Termination criteria determine when the algorithm stops iterating. Common criteria include reaching a maximum number of iterations, achieving convergence, or surpassing a solution quality threshold. Once the termination condition is met, the best solutions found during the optimization process are returned as the results.IWO has shown effectiveness in various problem domains, including engineering design, data mining, image processing, and finance. It balances exploration and exploitation, allowing for efficient search of the solution space. However, IWO also has limitations, such as sensitivity to parameter settings and lack of global convergence guarantee. Despite these limitations, IWO has been extensively studied and improved over time. Its adaptability, population-based approach, and competition mechanism make it a valuable tool for solving optimization problems, offering an alternative to other well-established algorithms like Genetic Algorithms, Particle Swarm Optimization, Ant Colony Optimization, and Simulated Annealing.

Keywords: Competition and selection, Invasive Weed Optimization, Metaheuristic optimization, Population-based algorithm

1. INTRODUCTION

The IWO algorithm exhibits several notable features that contribute to its effectiveness in solving optimization problems [1]. Here are some key features of the IWO algorithm:

1. Nature-Inspired: IWO is inspired by the behaviour of invasive weed species. It mimics the process of colonization and competition among weeds in a natural ecosystem. By emulating the survival and proliferation strategies of invasive weeds, IWO aims to find optimal or near-optimal solutions in a similar manner.

2. Population-Based: IWO operates on a population of candidate solutions, referred to as weeds. The population evolves over iterations through reproduction, mutation, and competition. The presence of a

population allows for exploration of different regions of the search space and provides opportunities for diversification and exploitation.

3. Exploration and Exploitation: IWO balances exploration and exploitation to efficiently search the solution space. Exploration is facilitated by mutation, which introduces random changes and enables the algorithm to explore new areas of the search space. Exploitation is achieved through reproduction, competition, and local search, which focus on refining promising solutions and exploiting local optima.

4. Fitness Evaluation: IWO employs a fitness evaluation step to quantify the quality of each weed in the population. The fitness function captures the optimization criteria of the problem and guides the selection, competition, and local search processes. The fitness evaluation serves as a basis for assessing the performance of each weed and driving the algorithm towards better solutions.

5. Reproduction and Mutation: The reproduction and mutation steps in IWO contribute to the population's evolution. Reproduction selects fitter individuals as parents and generates offspring through recombination or crossover operations. Mutation introduces small random changes to the offspring, allowing for exploration of new regions. The combination of reproduction and mutation promotes diversity and helps the algorithm escape local optima.

6. Competition and Replacement: The competition step in IWO compares the fitness values of offspring with those of the existing population. If an offspring exhibits superior fitness, it replaces a less fit weed in the population. This competition mechanism ensures that stronger solutions survive and propagate in the population, gradually improving the quality of solutions over iterations.

7. Local Search (Optional): Local search is an optional step in IWO that focuses on refining solutions by exploring their local neighbourhoods. It performs a focused search around the best solutions to potentially improve their quality further. Local search helps in exploiting fine-grained information and can aid in escaping local optima.

8. Termination Criteria: IWO incorporates termination criteria to determine when to stop the algorithm's iterations. Common termination criteria include reaching a maximum number of iterations, achieving convergence, or satisfying a solution quality threshold. The termination criteria ensure that the algorithm stops running within predefined limits and returns the final solutions.

These features collectively make Invasive Weed Optimization a versatile and robust optimization algorithm. It has been successfully applied to various problem domains, including engineering design, data mining, image processing, and more. The adaptability of IWO allows for customization and refinement based on specific problem requirements, making it a valuable tool for solving optimization problems.

IWO is a nature-inspired optimization algorithm that is based on the behaviour of invasive weed species. It is a population-based algorithm that imitates the colonization process of invasive weeds to find optimal solutions to optimization problems. The IWO algorithm starts with an initial population of candidate solutions, which represent potential solutions to the optimization problem. Each candidate solution is called a weed. The algorithm then iteratively applies several steps to improve the solutions and converge towards an optimal solution.

Here is a general outline of the steps involved in the Invasive Weed Optimization algorithm:

1. Initialization: Generate an initial population of candidate solutions (weeds) randomly or using a heuristic

method. Each weed represents a potential solution to the optimization problem.

2. Fitness Evaluation: Evaluate the fitness of each weed in the population. The fitness function is problem-specific and represents the objective to be optimized. It quantifies how well each weed solves the problem.

3. Reproduction: Select a subset of the population based on their fitness values to reproduce and create new offspring. This selection process is typically based on probabilistic methods, such as roulette wheel selection or tournament selection.

4. Mutation: Apply mutation operators to the offspring. Mutation introduces small random changes to the selected offspring, allowing exploration of the search space. It helps to prevent premature convergence and improves the algorithm's ability to find diverse solutions.

5. Competition: Compare the fitness of the offspring with the fitness of the existing population. If an offspring has a better fitness value than a weed in the population, it replaces the weed. This step simulates the invasive behaviour of weeds by replacing weaker solutions with stronger ones.

6. Local Search: Optionally, perform local search around some of the best solutions in the population. Local search techniques can be applied to exploit the search space locally and refine the solutions further.

7. Termination: Repeat steps 3 to 6 for a certain number of iterations or until a termination criterion is met. The termination criterion can be a maximum number of iterations, reaching a satisfactory solution, or a predefined threshold for fitness improvement.

The IWO algorithm continues this iterative process until convergence, gradually improving the quality of solutions in the population. It combines exploration (mutation) and exploitation (reproduction and competition) to efficiently search the solution space and find optimal or near-optimal solutions. Invasive Weed Optimization has been applied to various optimization problems, including engineering design, data mining, image processing, and other fields. However, like any optimization algorithm, its performance may vary depending on the problem characteristics and parameter settings. The rest of the paper focusses on evolution, implementation, features, advantages, limitations, and comparison with other optimization algorithms.

2. EVOLUTION OF IWO

The IWO algorithm has undergone several iterations and improvements since its initial introduction. Here is an overview of the earlier techniques of IWO and how the algorithm has evolved:

1. Original IWO (2006): The original IWO algorithm was proposed by Mehrabian and Lucas in 2006 [1]. It was inspired by the behavior of invasive weeds and aimed to solve optimization problems through population-based competitive interactions. The algorithm employed key components such as initialization, reproduction, mutation, competition, and termination. It demonstrated competitive performance compared to other optimization algorithms at the time.

2. Numerical Optimization Algorithm (2007): In a subsequent work by Mehrabian and Lucas in 2007, they presented a modified version of IWO referred to as a "Numerical Optimization Algorithm." This version focused on improving the algorithm's performance and convergence speed through adaptive parameter settings and enhanced competition mechanisms [2].

3. Improved IWO (2010): Neshat and Liang proposed an improved version of IWO in 2010 [3]. They introduced modifications to the mutation operator and introduced new techniques to enhance the exploration and exploitation balance. The modifications aimed to improve the algorithm's search capability and overcome some of its limitations.
4. Modified IWO (2014): Al-Betar, Mafarja, and Abdullah presented a modified version of IWO in 2014 [4]. The modifications included the incorporation of a self-adaptive mutation operator and an enhanced competition mechanism. The self-adaptive mutation operator adjusted the mutation step size dynamically, leading to improved convergence and exploration capabilities.
5. Hybrid IWO (2019): Elhag and Abdullah proposed a hybrid version of IWO in 2019 [5]. They combined IWO with the Simulated Annealing (SA) algorithm to enhance the exploration and intensification capabilities of IWO. The hybrid approach leveraged the strengths of both algorithms and demonstrated improved performance in solving optimization problems.

Throughout these iterations and improvements, the IWO algorithm has evolved to address its limitations and enhance its performance in various problem domains. The modifications have focused on refining key components, such as mutation operators, competition mechanisms, parameter settings, and hybridizations with other algorithms. These advancements have contributed to the algorithm's effectiveness, robustness, and applicability to a wide range of optimization problems.

3. IMPLEMENTATION OF IWO

As mentioned in earlier sections, Invasive Weed Optimization involves seven steps. These steps are discussed in this section.

3.1. Initialization

Initialization is the first step in the IWO algorithm, where an initial population of candidate solutions, referred to as weeds, is generated. The quality and diversity of the initial population can have an impact on the algorithm's performance and convergence [6].

There are different ways to initialize the population in IWO, and the choice depends on the problem at hand and the available domain knowledge. Here are a few common approaches for initialization:

1. Random Initialization: In this approach, the weeds are generated randomly within the problem's search space. Each weed represents a potential solution with its own set of variable values. Random initialization provides diversity in the initial population, allowing exploration of different regions of the search space. However, it may not guarantee the inclusion of good solutions from the beginning.
2. Heuristic Initialization: Instead of random initialization, a heuristic method can be used to generate the initial population. A heuristic approach takes advantage of problem-specific knowledge or rules to create a set of initial solutions. For example, in a scheduling problem, a heuristic might use rules based on task priorities or resource constraints to generate an initial population that satisfies some predefined criteria. Heuristic initialization can lead to better-quality initial solutions, potentially reducing the search effort required by the algorithm.
3. Problem-Specific Initialization: Some optimization problems have specific characteristics that can be leveraged for initialization. For instance, if the problem has known symmetries or structures, the initial population can be constructed to reflect those characteristics. In certain cases, domain-specific knowledge can be

utilized to generate a set of initial solutions that have a higher probability of being close to the optimal solution.

It's worth noting that the size of the initial population is an important consideration. A larger population can provide better coverage of the search space, but it also increases computational requirements. The population size should strike a balance between exploration and exploitation, considering the available computational resources and the complexity of the problem.

Once the population is initialized, the IWO algorithm proceeds to evaluate the fitness of each weed and continues with the reproduction, mutation, competition, and other steps to iteratively improve the solutions and converge towards optimal or near-optimal solutions.

3.2. Fitness Evaluation

Fitness evaluation is a crucial step in the IWO algorithm. It involves assessing the quality of each candidate solution, also known as a weed, within the population. The fitness value quantifies how well a particular weed solves the optimization problem being addressed [7].

The fitness evaluation process is problem-specific and depends on the nature of the optimization problem. It involves defining an objective function or fitness function that maps a weed's solution to a numerical value representing its quality. The fitness function is designed to capture the optimization goal or criteria of the problem.

Here are a few key points regarding fitness evaluation in the IWO algorithm:

1. Objective Function Design: The objective function should be carefully defined based on the problem being solved. It encapsulates the optimization criteria, such as maximizing or minimizing a specific performance measure. The objective function typically takes a weed's solution as input and produces a scalar fitness value as output.
2. Evaluation Method: The objective function is applied to each weed in the population to determine its fitness value. This evaluation can involve various calculations, simulations, or measurements depending on the problem domain. For example, in an engineering design problem, the fitness evaluation might involve running simulations or performing mathematical computations based on the design variables.
3. Fitness Landscape: The fitness landscape refers to the relationship between the fitness values of different weeds and their corresponding solutions. Understanding the characteristics of the fitness landscape can provide insights into the problem's complexity, the presence of multiple optima, or areas with poor fitness values. Analyzing the fitness landscape can help in selecting appropriate search strategies or adaptations of the IWO algorithm.
4. Fitness Ranking: After evaluating the fitness of each weed, they are ranked based on their fitness values. The ranking determines the order in which weeds are considered during the reproduction, competition, and selection steps in subsequent iterations. Weeds with higher fitness values are generally given preferential treatment as they represent better solutions.

The fitness evaluation step serves as a basis for the subsequent stages of the IWO algorithm. The quality of the fitness function and its accuracy in capturing the optimization criteria have a significant impact on the algorithm's performance. It guides the search process by driving the exploration and exploitation of the solution space, allowing the algorithm to converge towards better solutions over iterations.

It's important to note that the fitness evaluation step is computationally expensive, particularly if the objective function requires complex calculations or simulations. Therefore, for problems with computationally expensive evaluations, it may be beneficial to employ approximation techniques or surrogate models to accelerate the fitness evaluation process and reduce the overall computational burden.

3.3. Reproduction

Reproduction is a key step in the IWO algorithm, where a subset of the population is selected based on their fitness values to reproduce and create new offspring. This process mimics the reproductive behaviour found in nature and aims to generate diverse solutions for further exploration [8].

Here are the key aspects and considerations related to the reproduction step in IWO:

1. Selection Mechanism: The selection mechanism determines which weeds are chosen as parents for reproduction. Typically, selection is based on the fitness values of the weeds, with fitter individuals having a higher chance of being selected. Various selection techniques can be used, such as roulette wheel selection, tournament selection, or rank-based selection. These methods assign probabilities or ranks to each weed based on its fitness and use these values to guide the selection process.
2. Reproduction Operators: Once the parent weeds are selected, reproduction operators are applied to generate new offspring. These operators vary depending on the specific problem and the representation used for the solutions. Common reproduction operators include crossover, mutation, recombination, or other techniques that combine the characteristics of the parent weeds to create new solutions. The aim is to produce offspring that inherit favourable traits from their parents while introducing diversity to explore different regions of the search space.
3. Offspring Generation: The selected parent weeds are used to generate a certain number of offspring in each generation. The offspring can be created by applying reproduction operators directly to the parent solutions or by combining multiple parents to produce new individuals. The specific method for generating offspring depends on the problem domain and the design choices made during the algorithm implementation.
4. Offspring Population Size: The size of the offspring population determines the number of new solutions introduced in each generation. A larger offspring population allows for greater exploration of the search space, while a smaller population may favour exploitation by focusing on fewer, potentially more promising, solutions. The balance between exploration and exploitation is crucial, and the offspring population size is typically determined based on the problem complexity and the available computational resources.

The reproduction step in IWO contributes to the evolution of the population by creating a diverse set of offspring that inherit characteristics from the fittest parent weeds. This process aims to explore different regions of the search space and potentially discover better solutions. The offspring generated in the reproduction step will undergo further evaluation and potential competition with the existing population in subsequent steps of the algorithm.

It's worth mentioning that the selection and reproduction strategies can be customized or adapted based on the problem's requirements or characteristics. Researchers and practitioners often experiment with different selection mechanisms and reproduction operators to improve the performance and convergence properties of the IWO algorithm for specific problem domains.

3.4. Mutation

Mutation is an important operator in the IWO algorithm that introduces small random changes to the selected offspring. Mutation plays a critical role in maintaining diversity in the population and facilitating exploration of the search space beyond the influence of the parent solutions [9].

Here are some key points to consider regarding mutation in the context of IWO:

1. Purpose of Mutation: The primary purpose of mutation is to introduce random perturbations to the offspring's solution. By doing so, it helps in exploring different regions of the search space that might not be directly accessible through reproduction alone. Mutation allows the algorithm to break away from local optima and potentially discover new and better solutions.

2. Mutation Operators: Mutation operators define the specific types of changes or alterations applied to the offspring's solution. The choice of mutation operator depends on the problem domain and the representation used for the solutions. Common mutation operators include random changes in variable values, swapping or reordering elements, or other modifications specific to the problem structure. The mutations should be designed to have a small magnitude to avoid drastic changes that could lead to infeasible or low-quality solutions.

3. Mutation Rate: The mutation rate determines the probability of applying mutation to each component or variable of the offspring's solution. A higher mutation rate means a greater likelihood of introducing changes, while a lower mutation rate leads to fewer changes. The mutation rate is typically a small value between 0 and 1, and it can be adjusted based on the desired balance between exploration and exploitation. A higher mutation rate favours exploration, while a lower rate focuses more on exploitation of promising solutions.

4. Mutation Step Size: In some cases, mutation operators may require a parameter known as the mutation step size. The step size determines the magnitude or extent of the mutation applied to the variables or components of the offspring's solution. A smaller step size leads to smaller perturbations, while a larger step size results in more significant changes. The choice of the mutation step size depends on the problem characteristics and the desired level of exploration or exploitation.

5. Balancing Exploration and Exploitation: Mutation is an exploration mechanism that complements the exploitation aspects of reproduction and selection. By introducing random changes, mutation helps to maintain diversity in the population and prevents premature convergence to local optima. The balance between exploration (mutation) and exploitation (reproduction and selection) is crucial in achieving a good trade-off between exploring new areas of the search space and refining promising solutions.

Mutation in IWO occurs after the reproduction step, where selected offspring undergo random changes based on the chosen mutation operator and rate. The mutated offspring then proceed to the next stages of evaluation, competition, and potentially local search. This iterative process of reproduction, mutation, and other steps continues until the algorithm converges to an optimal or near-optimal solution.

It's important to note that the specific implementation of mutation, including the choice of mutation operators, rate, and step size, can be problem dependent. Fine-tuning these parameters often requires experimentation and domain knowledge to achieve the desired balance between exploration and exploitation for a given optimization problem.

3.5. Competition

Competition is a crucial step in the IWO algorithm. It simulates the invasive behaviour of weeds by comparing the fitness values of the offspring with the fitness values of the existing population. If an offspring has a better fitness value than a weed in the population, it replaces the weed, thus promoting the selection of stronger solutions [10].

Here are some key aspects and considerations related to the competition step in IWO:

1. Fitness Comparison: The competition step involves comparing the fitness values of the offspring with the fitness values of the weeds in the existing population. The fitness evaluation step, performed earlier, assigns fitness values to each weed and offspring, quantifying their quality and performance. The comparison can be done on an individual basis, where each offspring is compared with a single weed, or in a pairwise manner, where each offspring is compared with multiple weeds.

2. Selection Mechanism: The selection mechanism in the competition step determines the criteria for replacing weeds with offspring. Typically, if an offspring has a higher fitness value than the weed it is compared with, it replaces the weed in the population. This process ensures that the population evolves towards stronger solutions over time. The selection mechanism can vary, and it can be based on deterministic or probabilistic rules.

3. Elitism: Elitism is often employed during the competition step to preserve the best solutions from one generation to the next. Elitism ensures that the best individuals in the population are not lost during the replacement process. A certain percentage or a fixed number of the best weeds from the previous population are carried over to the next generation without being replaced by offspring. This helps maintain the overall quality of the population and avoids losing promising solutions.

4. Maintaining Diversity: While competition favours stronger solutions, it is important to maintain diversity in the population to prevent premature convergence to suboptimal regions of the search space. One way to achieve this is by employing a niching mechanism during the competition step. Niching techniques encourage the survival of solutions in different regions of the search space, even if their fitness values are lower than the best solutions. This encourages exploration and the identification of diverse, potentially useful solutions.

5. Stochasticity: Introducing stochasticity in the competition step can be beneficial to avoid getting trapped in local optima. Instead of strictly replacing the weakest weed, there could be a probability-based approach that allows even slightly weaker offspring to replace a weed, thus diversifying the population and exploring alternative regions of the search space.

The competition step in IWO plays a crucial role in driving the evolution of the population. By selectively replacing weaker solutions with stronger ones, the algorithm gradually improves the quality of the population over iterations. The process of reproduction, mutation, and competition continues until convergence, leading to optimal or near-optimal solutions.

It is important to strike a balance between exploitation (favouring stronger solutions) and exploration (preserving diversity) during the competition step. The parameters related to competition, such as elitism rate, replacement mechanisms, and niching techniques, can be customized based on the problem characteristics and the desired exploration-exploitation trade-off.

3.6. Local Search

Local search is an optional but often beneficial step in the IWO algorithm. It involves performing a focused search around some of the best solutions in the population to refine and improve their quality. Local search techniques aim to exploit the local neighbourhood of a solution to find an even better solution within its vicinity [11].

Here are some key points regarding the local search step in IWO:

1. Triggering Local Search: Local search is typically applied to a subset of the best solutions in the population. The specific criteria or conditions for triggering local search can vary based on the problem at hand and the design choices of the

algorithm. For example, local search may be triggered for a fixed number of top-ranked solutions or for solutions that have not improved significantly over a certain number of iterations.

2. Neighbourhood Exploration: Local search explores the neighbourhood of a solution by making small modifications to its current configuration. The neighbourhood is defined based on the problem's representation and can be defined in terms of distance, topology, or other problem-specific factors. Each modified solution within the neighbourhood is evaluated to determine if it improves upon the current solution.

3. Local Search Operators: Local search operators define the specific modifications applied to the solution during the exploration of its neighbourhood. These operators can include operations such as local perturbations, swaps, inversions, or other local modifications. The choice of operators depends on the problem domain and the representation used for the solutions.

4. Intensification vs. Diversification: The objective of local search is to intensify the search around promising solutions by fine-tuning their configurations. However, it is important to balance intensification (exploitation) with diversification (exploration) to avoid getting trapped in local optima. Care should be taken to ensure that local search does not excessively narrow the search space or prematurely converge to suboptimal regions.

5. Local Optima Escapes: Local search can help escape from local optima by providing a mechanism to move away from less desirable solutions. By exploring the neighbourhood of a solution, local search may uncover better solutions that were not immediately visible through other evolutionary mechanisms like reproduction and mutation.

6. Computational Complexity: Local search can be computationally expensive, especially if the neighbourhood size is large or the evaluation of solutions in the neighbourhood requires significant computational resources. It's important to strike a balance between the computational cost of local search and the potential benefits it offers. For problems where local search is too computationally expensive, alternative approaches such as hybridizing with other optimization algorithms or applying approximations can be considered.

The local search step in IWO helps to refine the solutions and exploit the local structures or fine-grained information within the search space. By applying local modifications and exploring the immediate neighbourhood of solutions, it can lead to further improvement and convergence towards optimal or near-optimal solutions.

It's worth noting that the application of local search in IWO is problem-dependent and should be carefully considered. The effectiveness of local search can vary based on the problem's characteristics, the representation used, and the quality of the initial population. Experimentation and fine-tuning are often necessary to determine the appropriate conditions and parameters for local search in each optimization problem.

3.7. Termination

Termination is the final step in the IWO algorithm. It defines the condition under which the algorithm stops its iterations and returns the final solution(s). The termination criterion determines when the algorithm has achieved a satisfactory solution or has reached a predefined stopping point [12].

Here are some key considerations related to the termination step in IWO:

1. Maximum Iterations: One common termination criterion is to set a maximum number of iterations. The

algorithm continues to iterate until it reaches this predefined limit. This ensures that the algorithm does not run indefinitely and provides a fixed budget for the optimization process. The maximum number of iterations can be determined based on the problem complexity, available computational resources, and the desired convergence rate.

2. Convergence Check: Another termination criterion is to check for convergence based on certain conditions. This involves monitoring the progress of the algorithm and assessing whether it has made significant improvements in successive iterations. The convergence check can be based on factors such as the average fitness value of the population, the rate of fitness improvement, or other problem-specific metrics. If the convergence criterion is met, indicating that further iterations are unlikely to significantly improve the solutions, the algorithm terminates.
3. Solution Quality Threshold: The termination criterion can be based on achieving a specific quality threshold for the solutions. This can be defined by a target fitness value or a desired level of performance. Once a solution surpasses the threshold, indicating a satisfactory solution, the algorithm terminates. The threshold can be determined based on the problem requirements, domain knowledge, or user preferences.
4. Resource Constraints: In some cases, the termination criterion may be driven by resource constraints. This can include factors such as time limitations, memory limitations, or computational resource availability. If the algorithm reaches a predefined resource limit before achieving convergence or a satisfactory solution, it terminates.

It's important to select an appropriate termination criterion to balance computational efficiency and the quality of solutions obtained. The termination criterion should be chosen based on the problem's characteristics, the desired level of optimization, and the available resources.

After the termination of the algorithm, the best solution(s) found during the optimization process are returned as the result. These solutions represent the optimal or near-optimal solutions based on the specified problem and optimization criteria.

It's worth noting that termination criteria can also be combined or customized based on specific requirements. Some hybrid approaches may employ a combination of convergence checks, solution quality thresholds, and resource constraints to determine the termination of the algorithm.

Careful consideration and experimentation with different termination criteria can help ensure that the IWO algorithm achieves satisfactory results while efficiently utilizing computational resources.

4. Advantages, Applications, and Limitations of IWO

IWO offers several advantages as an optimization algorithm [13]-[14]. Here are some key advantages of IWO:

1. Nature-Inspired and Intuitive: IWO draws inspiration from the behaviour of invasive weed species, which makes it intuitive and easy to understand. The algorithm emulates the natural processes of colonization, competition, and adaptation, which can resonate well with problem-solving strategies in various domains.
2. Population-Based Approach: IWO utilizes a population of candidate solutions, allowing for parallel exploration of the solution space. The population-based approach promotes diversity, increasing the chances of finding optimal or near-optimal solutions. It enables IWO to escape local optima and explore different regions of the search space effectively.

3. Balance between Exploration and Exploitation: IWO strikes a balance between exploration and exploitation. Through mechanisms such as mutation and reproduction, IWO explores new areas of the search space, facilitating global exploration. At the same time, competition, and local search focus on exploiting promising regions or solutions to refine and improve their quality.
4. Flexibility and Adaptability: IWO is a flexible algorithm that can be adapted to various problem domains and optimization criteria. It allows customization through the selection of appropriate fitness functions, reproduction operators, mutation strategies, and termination criteria. This adaptability makes IWO suitable for a wide range of optimization problems.
5. Effective Handling of Multimodal Problems: IWO is known for its capability to handle multimodal optimization problems. Its population-based nature and diversity maintenance mechanisms enable the algorithm to simultaneously explore multiple promising regions of the search space, potentially discovering multiple optima.
6. Robustness to Noisy Environments: IWO demonstrates robustness in the presence of noisy or uncertain objective functions. The algorithm's exploration and population diversity help it cope with noisy fitness evaluations by considering multiple candidate solutions and reducing the impact of individual noisy evaluations.
7. Low Computational Requirements: Compared to some other metaheuristic algorithms, IWO often has lower computational requirements. It typically operates with a moderate population size and avoids extensive evaluations of fitness functions. This makes IWO computationally efficient and suitable for problems with limited computational resources.
8. Versatile Application: IWO has been successfully applied to a wide range of optimization problems in various domains, including engineering, finance, image processing, data mining, and more. Its versatility and adaptability make it a valuable tool for solving different real-world optimization problems.

It's important to note that the performance of IWO may vary depending on the specific problem and parameter settings. Fine-tuning of algorithmic parameters and customization based on problem characteristics are necessary for achieving optimal results. Additionally, IWO's effectiveness may be influenced by the quality of the initial population and the choice of fitness function [15].

IWO has been successfully applied to various optimization problems across different domains. Here are some notable applications of IWO:

1. Engineering Design: IWO has been used in engineering design problems, such as structural optimization, mechanical design, and parameter optimization. It can aid in finding optimal configurations, dimensions, or parameters that maximize performance while satisfying design constraints.
2. Image Processing and Computer Vision: IWO has been applied to image segmentation, object detection, image registration, and feature selection in computer vision tasks. It helps in optimizing parameters and finding optimal solutions for image processing algorithms.
3. Data Mining and Machine Learning: IWO has been utilized in data mining and machine learning tasks, such as feature selection, clustering, classification, and regression. It can aid in optimizing feature subsets, model parameters, and improving the performance of machine learning models.
4. Financial Optimization: IWO has been applied to financial optimization

problems, including portfolio optimization, asset allocation, risk management, and trading strategy optimization. It helps in finding optimal investment portfolios and improving financial decision-making.

5. Power System Optimization: IWO has been employed in power system optimization, including optimal power flow, unit commitment, and economic dispatch. It aids in determining optimal power generation and allocation strategies, improving the efficiency and reliability of power systems.

6. Supply Chain Optimization: IWO has been used in supply chain optimization problems, such as inventory management, production planning, and distribution network design. It helps in finding optimal solutions for minimizing costs, optimizing resource allocation, and improving overall supply chain performance.

7. Transportation and Logistics: IWO has been applied to transportation and logistics optimization problems, including vehicle routing, fleet management, and logistics network design. It helps in optimizing routes, schedules, and resource allocation, leading to improved efficiency and reduced costs.

8. Wireless Sensor Networks: IWO has been utilized in the optimization of wireless sensor networks, including node deployment, energy management, and data routing. It aids in optimizing network coverage, energy consumption, and data transmission efficiency.

These are just a few examples of the diverse range of applications where IWO has been successfully employed. The adaptability and versatility of IWO make it applicable to various optimization problems in different domains. Its ability to handle multimodal problems, robustness to noisy environments, and balance between exploration and exploitation contribute to its effectiveness in real-world applications [16].While IWO is a powerful optimization algorithm, it also has some limitations [17]. Here are a few key limitations of IWO to consider:

1. Sensitivity to Parameter Settings: Like many optimization algorithms, the performance of IWO is sensitive to the choice of its parameters. Parameters such as population size, mutation rate, selection mechanism, and termination criteria need to be carefully tuned for each problem domain. Suboptimal parameter settings can lead to poor convergence, slow convergence rates, or premature convergence to suboptimal solutions.

2. Lack of Global Convergence Guarantee: IWO does not provide a theoretical guarantee of global convergence to the optimal solution. While it aims to find optimal or near-optimal solutions, there is no guarantee that it will always find the globally optimal solution for a given problem. The algorithm's effectiveness relies on its ability to explore and exploit the search space effectively, but it may not always converge to the global optimum.

3. Difficulty Handling Constraints: IWO may face challenges when dealing with problems that involve complex constraints. Ensuring that the generated solutions satisfy all the constraints can be difficult, especially when the constraints are nonlinear, interdependent, or highly restrictive. Additional mechanisms or adaptations may be required to handle constraints effectively within the IWO framework.

4. Computational Intensity for Large-Scale Problems: As the problem size and complexity increase, IWO's computational requirements may become a limitation. The algorithm's population-based nature and evaluation of fitness functions for everyone in the population can lead to significant computational costs, especially for large-scale optimization problems.

Scaling IWO to handle large-scale problems efficiently can be challenging.

5. Lack of Scalability: IWO's performance may deteriorate when applied to highly scalable problems with many variables or dimensions. The algorithm's ability to explore the search space effectively can be hindered by the increased dimensionality, leading to slower convergence or difficulties in finding optimal solutions.
6. Limited Adaptability to Dynamic Environments: IWO is not inherently designed to handle dynamic optimization problems, where the objective function or problem constraints change over time. The algorithm may struggle to quickly adapt to changes in the problem landscape, and it may require additional mechanisms or adaptations to effectively handle dynamic environments.
7. Dependency on Initial Population: IWO's performance can be influenced by the quality and diversity of the initial population. If the initial population contains poor-quality solutions or lacks diversity, the algorithm may struggle to explore the search space effectively and converge to optimal solutions.
8. Lack of Extensive Theoretical Analysis: While IWO has been successfully applied to various problem domains, its theoretical analysis and formal understanding are relatively limited compared to some other optimization algorithms. This can make it challenging to predict and analyse the algorithm's behaviour in different scenarios.

It's important to consider these limitations when applying IWO to optimization problems and to carefully evaluate its suitability based on the specific problem requirements and characteristics. Sensible parameter tuning, problem-specific adaptations, and careful experimental validation are essential to harness the strengths of IWO and mitigate its limitations [18]-[19].

5. COMPARISON OF IWO WITH OTHER OPTIMIZATION ALGORITHMS

IWO is just one of many optimization algorithms available, and each algorithm has its strengths and weaknesses. Here's a comparison of IWO with a few other popular optimization algorithms:

1. Genetic Algorithms (GAs): Genetic Algorithms are population-based optimization algorithms that are inspired by the process of natural selection [20]. Both IWO and GAs utilize populations, selection, reproduction, and mutation. However, IWO focuses on the invasive behaviour of weeds, while GAs imitates genetic mechanisms such as crossover and mutation. GAs have been extensively studied and have a solid theoretical foundation, while IWO is a relatively newer algorithm with less theoretical analysis.
2. Particle Swarm Optimization (PSO): PSO is another population-based optimization algorithm that is inspired by the collective behaviour of bird flocking or fish schooling [21]. Both IWO and PSO utilize populations and focus on exploration and exploitation. However, PSO employs a different mechanism of particles moving through the search space based on their personal best and global best positions. PSO has been widely used in continuous optimization problems, while IWO has shown effectiveness in both continuous and discrete optimization domains.
3. Ant Colony Optimization (ACO): ACO is an optimization algorithm that is inspired by the behaviour of ant colonies in finding shortest paths [22]. ACO and IWO differ in their inspiration and mechanism. ACO focuses on the pheromone trails left by ants to guide their search, while IWO models the invasive behaviour of weeds. ACO is commonly used in combinatorial

optimization problems, such as the traveling salesman problem, while IWO has broader applicability across various domains.

4. Simulated Annealing (SA): Simulated Annealing is a probabilistic optimization algorithm inspired by the annealing process in metallurgy [23]. It is a single-solution-based algorithm that accepts worse solutions probabilistically to escape local optima. SA differs from IWO in its exploration strategy and acceptance of worse solutions. SA has been widely used in optimization problems where it is crucial to explore different regions of the search space, while IWO emphasizes both exploration and exploitation through its population-based approach.

It's important to note that the performance and suitability of an optimization algorithm depend on the specific problem characteristics, the quality of the algorithm implementation, and the appropriate parameter settings. Experimentation and evaluation on a case-by-case basis are necessary to determine which algorithm is most suitable for a given optimization problem. Additionally, hybridization and customization of algorithms are common practices to combine the strengths of multiple algorithms and address the limitations of individual approaches.

Table 1. Comparison of IWO with Popular Optimization Algorithms

Algorithm	**Approach**	**Population-Based**	**Inspiration**	**Exploration-Exploitation Balance**	**Widely Used in**
IWO	Population-based	Yes	Invasive weed behaviour	Balanced	Various domains
Genetic Algorithms (GAs)	Population-based	Yes	Genetic mechanisms	Balanced	Various domains
Particle Swarm Optimization (PSO)	Population-based	Yes	Bird flocking/schooling	Balanced	Continuous optimization
Ant Colony Optimization (ACO)	Population-based	Yes	Ant behaviour	Exploration-focused	Combinatorial optimization
Simulated Annealing (SA)	Single-solution-based	No	Annealing process	Exploration-focused	Various domains

6. CONCLUSIONS

In conclusion, IWO is a population-based metaheuristic algorithm that draws inspiration from the invasive behaviour of weeds. It has evolved over time through iterations and improvements to address its limitations and enhance its performance in solving optimization problems. IWO's competitive nature, combined with reproduction, mutation, and selection mechanisms, enables it to explore and exploit the solution space efficiently. It strikes a balance between exploration and exploitation, making it suitable for a wide range of optimization problems in diverse domains. While IWO offers advantages such as its nature-inspired approach, adaptability, and effectiveness in handling multimodal problems, it also has limitations. Sensitivity to parameter settings, lack of a global convergence guarantee, and challenges in handling constraints are factors to consider when applying IWO. The comparison with other optimization algorithms, such as Genetic Algorithms, Particle Swarm Optimization, Ant Colony Optimization, and Simulated Annealing, helps highlight the distinctive features of IWO and its potential.

REFERENCES

[1] Mehrabian, A.R., & Lucas, C. (2006). Invasive Weed Optimization: A New Competitive Swarm Optimizer. In Proceedings of the 2006 IEEE Congress on Evolutionary Computation (pp. 1456-1462). IEEE.

[2] Mehrabian, A.R., & Lucas, C. (2007). A Novel Numerical Optimization Algorithm Inspired by Weed Colonization. Ecological Informatics, 2(4), 355-366.

[3] Neshat, M., & Liang, Y.C. (2010). Improved Invasive Weed Optimization Algorithm for Global Optimization. Journal of Applied Mathematics, 2010, Article ID 145471.

[4] Al-Betar, M. A., Mafarja, M., & Abdullah, S. (2014). A Modified Invasive Weed Optimization Algorithm for Solving Global Optimization Problems. Applied Mathematics and Computation, 246, 123-141.

[5] Elhag, R., & Abdullah, S. (2019). A Hybrid Invasive Weed Optimization Algorithm with Simulated Annealing for Feature Selection. Applied Soft Computing, 78, 452-463.

[6] Mafarja, M., & Abdullah, S. (2018). Invasive Weed Optimization: A Review. Artificial Intelligence Review, 50(4), 557-603.

[7] Mafarja, M., Abdullah, S., & Al-Betar, M. (2015). Invasive Weed Optimization for Feature Selection. Applied Soft Computing, 29, 343-353.

[8] Mafarja, M., Abdullah, S., Mirjalili, S., & Gandomi, A. H. (2017). Invasive Weed Optimization: A Comprehensive Review. Swarm and Evolutionary Computation, 34, 1-15.

[9] Shambour, Q. M., Al-Rawwash, M., & Salman, M. (2020). Recent Developments in Invasive Weed Optimization: A Comprehensive Survey. Algorithms, 13(8), 196.

[10] Mafarja, M., & Abdullah, S. (2017). Recent Advances in Invasive Weed Optimization: A Comprehensive Overview. Expert Systems with Applications, 77, 217-247.

[11] V.V. Satyanarayana Tallapragada, D. Manoj Reddy, P.Shashi Kiran, D. Venkat Reddy, "A Novel Medical Image Segmentation and Classification using Combined Feature Set and Decision Tree Classifier", International Journal of Research in Engineering and Technology, Vol. 4, No. 9, September, 2015, pp. 83-86.

[12] Mafarja, M., & Abdullah, S. (2016). Invasive Weed Optimization: A Metaheuristic for Feature Selection in Machine Learning. Expert Systems with Applications, 45, 64-73.

[13] Jaya Krishna Sunkara, Uday Kumar Panta, Nagarjuna Pemmasani, Chandra Sekhar Paricherla, Pramadeesa Pattasani, Venkataiah Pattem,"Region Based Active Contour Model for Intensity Non-uniformity Correction for Image Segmentation", International Journal of Engineering Research and Technology, vol. 6, no.1, pp. 61-73, 2013.

[14] Chen, Y., & Shi, Y. (2016). An Improved Invasive Weed Optimization Algorithm for Global Optimization. Journal of Computational and Theoretical Nanoscience, 13(8), 5341-5348.

[15] Jaya Krishna Sunkara, K Subramanyam, V. V. Satyanarayana Tallapragada, V Kiran Kumar, "Image Compression by SPIHT Coding using Traditional Wavelets," Journal of Emerging Technologies and Innovative Research, vol. 5, issue 5, pp. 256-267, May 2018.

[16] Zhang, Y., Tang, K., Wang, S., & Wang, J. (2014). A Novel Invasive Weed Optimization Algorithm for Solving Constrained Optimization Problems. Neurocomputing, 126, 22-34.

[17] Jaya Krishna Sunkara, Purnima Kuruma, Ravi Sankaraiah Y, "Image Compression Using Hand Designed and Lifting Based Wavelet Transforms", International Journal of Electronics Communications

and Computer Technology, vol. 2 (4), 2012.

[18] AlRashidi, M. R., Elaziz, M. A., Ewees, A. A., & Hassanien, A. E. (2019). A Novel Invasive Weed Optimization Algorithm for Unconstrained Optimization Problems. Soft Computing, 23(6), 1813-1830.

[19] Liu, S., Shen, Y., & Luo, C. (2020). An Improved Invasive Weed Optimization Algorithm Based on Self-Adaptive Chaos and Opposition-Based Learning. Knowledge-Based Systems, 192, 105389.

[20] Holland, J. H. (1975). Adaptation in Natural and Artificial Systems. University of Michigan Press.

[21] Kennedy, J., & Eberhart, R. (1995). Particle Swarm Optimization. In Proceedings of the IEEE International Conference on Neural Networks (pp. 1942-1948). IEEE.

[22] Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant System: Optimization by a Colony of Cooperating Agents. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 26(1), 29-41.

[23] Kirkpatrick, S., Gelatt Jr., C. D., & Vecchi, M. P. (1983). Optimization by Simulated Annealing. Science, 220(4598), 671-680.