WILEY | Hindawi

Research Article

Chaotic Honeybees Optimization Algorithms Approach for Traveling Salesperson Problem

Pedro Palominos [iD],[1] **Carla Ortega,**[1] **Miguel Alfaro** [iD],[1] **Guillermo Fuertes** [iD],[1,2] **Manuel Vargas** [iD],[1] **Mauricio Camargo** [iD],[3] **Victor Parada** [iD],[4] **and Gustavo Gatica** [iD][5]

[1]*Industrial Engineering Department, University of Santiago de Chile, Avenida Victor Jara 3769, Santiago, Chile*
[2]*Facultad de Ingeniería, Ciencia y Tecnología, Universidad Bernardo O'Higgins, Avenida Viel 1497, Ruta 5 Sur, Santiago, Chile*
[3]*Université de Lorraine, ERPI, Nancy F54000, France*
[4]*Department of Informatics Engineering, University of Santiago of Chile, Avenida Victor Jara 3659, Santiago, Chile*
[5]*Facultad de Ingeniería, Universidad Andres Bello, Antonio Varas 880, Santiago, Chile*

Correspondence should be addressed to Guillermo Fuertes; guillermo.fuertes@usach.cl

Due to the difficulty in solving combinatorial optimization problems, it is necessary to improve the performance of the algorithms by improving techniques to deal with complex optimizations. This research addresses the metaheuristics of marriage in honeybees optimization (MBO) based on the behavior of bees. The current study proposes a technique for solving combinatorial optimization problems within proper computation times. The purpose of this study focuses on the travelling salesperson problem and the application of chaotic methods in important sections of the MBO metaheuristic. Three experiments were conducted to measure the efficiency and quality of the solutions: (1) MBO with chaos to generate initial solutions (MBO2); (2) MBO with chaos in the workers (MBO3); and (3) MBO with chaos to generate initial solutions and the workers (MBO4). The application of chaotic functions in MBO was significantly better at solving the travelling salesperson problem.

1. Introduction

The term metaheuristics was first introduced by Glover [1] in 1986. Establishing a mathematical model for complex systems is often a difficult task [2]. Classical methods depend on the type of objective function and constraint and on the type of variable used in modeling the problem. In addition, the effectiveness of classical algorithms is highly dependent on the solution section (convex or nonconvex), the number of decision variables, and the number of constraints in the modeling of the problem [3]. However, classical optimization algorithms are insufficient for large-scale combinational and nonlinear problems [4]. According to Bingol and Alatas [5], artificial intelligence-based metaheuristic methods are generally categorized as physics-based, biologically-based, social-based, musical-based, sport-based, swarm-based, plant-based, chemistry-based, light-based, mathematics-based, and water-based. Efficient hybrid

methods have also been proposed by combining these algorithms. Based on a single solution (direct search algorithms), the following algorithms can be found: simulate annealing (SA) [6], taboo search (TS) [7, 8], random walk (RW) [9], and hill climbing (HC) [10], among others, and population-based algorithms such as spider monkey optimization (SMO) [11]; particle swarm optimization (PSO) [12–14]; ant colony optimization (ACO) [15]; artificial immune system (AIS) [16]; whale optimization [17]; genetic algorithm (GA) [18, 19]; firefly algorithm [20]; grey wolf optimizer (GWO) [21]; bee algorithm (BA) [22]; artificial bee colony (ABC) [23]; queen bee evolution (QBE) [24]; bee system (BS) [25, 26]; bee colonies optimization (BCO) [27]; BeeAdHoc [28, 29]; and marriage in honey bees optimization (MBO) [30, 31] and its different versions such as honey bees mating optimization (HBMO) [32, 33] fast marriage in honey bees optimization (FMHBO) [34], and honey bees optimization (HBO) [35].

MBO is a metaheuristic proposed by Abbass [30] in 2001, where queens, drones, larvae, and workers interact to solve different problems. The genetic material of queens and drones gives rise to larvae cared for by the workers, and the latter heuristics improve the solution [36]. According to Shamsaldin et al. [37], MBO has advantages over GA when performing a local search by iteration.

Alfaro et al. [38, 39] researched dynamic systems to find new heuristic techniques that generated better solutions or improved the convergence speed. They proposed using chaos theory as an optimization tool. Chaos is a nonlinear deterministic phenomenon, sensitive to initial conditions and capable of determining all states within a range without repetition [40, 41]. It is used in optimization because it is easy to traverse the search space and avoids local minimums.

Other studies [42–46] related to evolutionary algorithms enhanced by chaotic maps have been described. In those algorithms, the generation of random values for the different parameters in the model is replaced by the application of chaotic maps for generating those parameters. The performance of these metaheuristics with evolutionary operators generated with chaotic numbers is better than that with evolutionary operators with random numbers. Evolutionary algorithms reduce premature convergence and increase performance optimization. Numerous chaotic systems, such as the Henon map, Logistic map, Rossler map, Tent map, and Zaslavskii map, can be used. The Lorenz system results in the most efficient one [47].

Fuertes et al. [47] have studied the performance of chaotic numbers used with a raw genetic algorithm. Their work proposes a new edge for the chaotic genetic algorithm (CGA) and the importance of entropy in the initial population. CGA uses chaotic maps to modify the stochastic parameters of the genetic algorithm. The algorithm modifies the parameters of the initial population using chaotic series and then analyzes the entropy of such a population. The numerical experiment demonstrates a correlation between entropy and the performance of the algorithm. The study concludes that the chaotic series with larger entropy used in CGA has better performance optimization than random series.

Vargas et al. [48] studied the effect of entropy on the performance of the genetic algorithm modified using a time series of earthquakes and winds. In this study, the genetic algorithm modifies the stochastic parameters of the genetic algorithm with the chaotic sequences of numbers from the earthquake and wind time series. The experiment demonstrates better optimization performance compared to the stochastic genetic algorithm.

The purpose of this study is to analyze and evaluate whether chaos theory improves solution quality in MBO for the travelling salesperson problem (TSP) using chaotic methods in essential MBO sections. This research measured this technique's efficiency and solution quality.

To measure the obtained result, all experiments will be carried out using TSP. TSP is difficult to solve and requires a lot of computational time to find optimal solutions, so most prefer to use heuristic techniques [49–51]. Therefore, TSP was chosen to measure the efficiency of chaos in MBO.

1.1. Related Literature. Pan et al. [52] proposed a new neuronal circuit with a synapse module, a neural module, and chaotic characteristics. The study designed and studied a chaotic neural network (CNN), applied associative memory to bipolar images, and verified that the circuit was capable of chaotic associative memory.

Lin [53] proposed a predefined time stability theorem based on the Lyapunov function to facilitate the anti-synchronization of the two CNNs. The engineered driver causes antisync errors to converge within a predefined period.

Long et al. [54] studied finite-time control theory for a class of state-based switched inertial chaotic neural networks (SBSCINN) with distributed delays in practical applications. These require fast or limited convergence time to ensure CNN stability. The study also generated new criteria for stabilizing SBSCINNs.

Chen et al. [55] studied a fractional order discrete Hopfield neural network (FODHNN) in an image encryption system, showed that it had complex nonlinear dynamic behavior, and designed a timing controller.

Srivastava et al. [56] proposed an algorithm to improve data security using a hybrid model that uses an adaptive coding technique with a CNN hopfield. Their results indicated that the speed and security of the transmitted data are better than those of traditional algorithms. Additionally, the calculation time for the proposed algorithm was shorter than that of traditional algorithms.

Jiang and Chen [57] proposed a blind restoration model for super-resolution images based on CNN. The average values of blind restore time, image sharpness, and power consumption were 9.273 ms, 99.045%, and 118.524 J, respectively. The model restoration performance was good, and the super-resolution blind image was the most like the original image.

Lian et al. [58] proposed a new discontinuous Lyapunov function to solve local synchronization control for CNN with sampled data and saturation actuators. Additionally, the authors provided a method for optimizing the design of sampled-data controllers. Their proposed method could be extended to discrete-time CNN by using finite-sum inequalities.

Lee [59] proposed the use of CNN in quantum finance. They studied quantum finance oscillators (QFOs) using Lee oscillators and different QFO applications, including quantum financial prediction using CNN, deep learning CNN, and chaotic multi-agent based smart trading systems. They reported that chaotic neural oscillators could be adopted in almost any artificial intelligence tool or technology to improve efficiency and performance.

Lahmiri and Bekiros [60] implemented deep learning CNN to forecast the price of the three most traded digital currencies, Bitcoin, Digital Cash, and Ripple. The time series of all digital currencies exhibits fractal dynamics, long memory, and self-similarity. They found that deep learning forecasted the inherent chaotic dynamics of cryptocurrency markets very efficiently.

Wang and Zhang [61] designed two fractional observers for CNN with and without parametric uncertainties. Their

observation errors eventually converged in a small region using fractional stability criteria, and their proposed observers were robust.

Chen et al. [62] proposed a new multiheating dynamic PSO framework that combines the chaotic grouping mechanism (CGM) and the dynamic regrouping strategy (DRS), which they named CGPSO-DRS. Their method improved the convergence speed and time to find the global optimum compared to similar population-based approaches and next-generation PSO variants.

Rosalie et al. [63] proposed a Bayesian optimization model to select Rössler system parameters used in chaotic ant colony optimization for coverage (CACOC). They developed this model to manage the movements of an unmanned aerial vehicle swarm (UAVs), and their approach improved the UAV swarm coverage speed.

Tummala et al. [64] proposed a new algorithm called "war strategy optimization" (WSO) where the war strategy is modeled as an optimization process where each soldier moves dynamically toward the optimal value. The performance of the algorithm is compared with the other ten popular metaheuristic algorithms. Experimental results for several optimization problems demonstrate the superiority of WSO.

Alatas [65] developed a computational method inspired by the types and occurrence of chemical reactions called the artificial chemical reaction optimization algorithm (ACROA). The method is based on the principles of chemical reactions. These chemical reactions transform reactants through a sequence of reactions into products. The initial reactants are distributed in the feasible search region, obtaining high-quality partial solutions and achieving optimal solutions in a few cycles. This method includes global and local search capabilities and does not require a local search method to refine the search.

The proposed Chaotic MBO technique is based on a well-known metaheuristic-type method with more than a decade of research. The calibration of the MBO metaheuristic has been extensively studied, and it allows us to start the research from a very robust state-of-the-art. The research took advantage of this knowledge to implement chaotic determinism in the metaheuristic and to search for improvements in the optimization technique.

Compared to other techniques presented by many authors, the chaotic deterministic characteristics of the phenomena used for this study, such as chemical processes, human behavior, chaotic maps, or quantum phenomena, establish a nonlinear search for viable solutions. The results of these techniques have presented sustainably robust and high-quality solutions. Therefore, the selection of the MBO metaheuristic and its modification with chaotic determinism allows us to quickly obtain results of high accuracy.

1.2. Limitations and Contributions. The development of this new computational method was inspired by the behavior of honeybees through the interaction of the different members of the hive. This method establishes the global search capability through crossover cycles of the best partial solutions.

The proposed research improves the performance of the metaheuristic methodology by applying chaotic determinism for creating drones and the process of worker bees. The workers improve the solution quality with chaotic modifications like chaos simulated annealing (CSA), chaotic tabu search neural network based on a path (CTSNN), or chaotic swap local search (CSLS). Previous research has demonstrated the robustness of the MBO method for discarding partial optimal solutions; the proposed modification improves the ability to search for optimal solutions.

The metaheuristic method was used to solve the TSP problem; the results show an improvement in the performance of the chaotic MBO metaheuristics for this problem. For other engineering problems, it is necessary to replicate the experiment with new parameters.

Finally, this metaheuristic method does not require many parameters for its implementation; the search time does not increase quite significantly; and it can be adapted for multiobjective optimization models or parallel distributions.

2. Research Methodology

2.1. The Travelling Salesman Problem. TSP is a well-studied problem in combinatorial optimization because it is difficult to solve. A vendor wants to visit n cities only once and finally return to the city where they started their journey. The idea is to minimize the journey cost, where the cost of a city i to a city j corresponds to the distance travelled between said cities (C_{ij}). Computer scientists call it an NP-complex problem [66], because it cannot be solved by an algorithm in polynomial time.

Mathematically, TSP can be expressed as follows:
Objective function:

$$\min \sum_{i,j} c_{ij} \cdot x_{ij}. \tag{1}$$

Subject to:

$$\sum_i x_{ij} = 1 \quad \forall j \neq i,$$
$$\sum_j x_{ij} = 1 \quad \forall i \neq j, \tag{2}$$

$$u_1 = 1,$$
$$2 \leq u_i \leq n \quad \forall i \neq 1, \tag{3}$$

$$u_i - u_j + 1 \leq (n-1)\left(1 - x_{ij}\right) \quad \forall i \neq 1, \forall j \neq 1, \tag{4}$$

$$u_i \geq 0 \quad \forall i,$$
$$x_{ij} \in \{0, 1\} \quad \forall i, j. \tag{5}$$

Constraints (2)–(5) reach together to eliminate the subpaths in the solution.

TSP can be used to solve different problems. Robotics can minimize the number of machine movements, where each movement has an associated cost [67]. Distribution logistics can find the minimum path from the point of origin,

passing through known places and finally returning to the point of origin [68]. This happens in a dispatch system that minimizes fuel usage and ensures that the products are delivered as quickly as possible.

2.2. Honeybee Optimization Metaheuristics.

Among the methods inspired by nature is the artificial bee colony (ABC) algorithm, which belongs to the swarm intelligence family. Within the bee colony methods, there are two distinct types of metaheuristics. The first approach is the bee collecting pollen algorithm (BCPA), where bees seek to obtain the maximum amount of nectar [69].

MBO is based on the social behavior of bees in which queens, drones, larvae, and workers interact to solve overly complex combinatorial problems.

2.2.1. Description of the Metaheuristics.

A bee colony can be formed in two different ways. The first method is called independent foundation, in which the colony is created by one or more queens that build a nest, produce eggs, and feed the larvae. Later, the queens only take charge of laying eggs, and the workers conduct the work in the hive. The second method is called swarming, in which one or more queens and workers create the colony from the original colony, and the division of labor begins immediately; therefore, the queens are only responsible for laying eggs. Colony formation by a single queen is called haplometrosis; otherwise, it is called pleometrosis. Additionally, if the hive has only one queen during its life cycle, it is monogynous, and if it has many queens, it is polygynous [70].

Colonies are made up of queens, drones, larvae, and workers. Queens lay eggs, and male drones are haploid and can only affect the queen's genotype via mutations. Fertilization occurs in the air, and the queen interbreeds 7 to 20 drones in one session. The drone's genetic material is stored in the queen's spermatheca, and a random mixture of the stored sperm fertilizes the queen's eggs. The workers who care for the larvae also occasionally lay eggs.

Our model randomly creates an initial population of queens, and its solutions are improved using workers. Each queen initiates a mating flight with initial energy and speed.

The mating flight is a set of transitions where the queen moves in the air and mates (or does not mate) with a drone according to a probabilistic function. If mating occurs, the drone's sperm is stored within the queen's spermatheca. The flight ends when the queen's energy has been exhausted or the spermatheca is at maximum capacity. After that, her energy and speed decreased.

When the queen is in the nest, the drone's genetic material is selected to create larvae cared for by the workers. The workers are represented by heuristics that seek to improve the larva (solution).

Eventually, the best larvae become queens and replace the worst queens. The remaining queens and larvae are eliminated to begin a new flight cycle.

The probability that queens and drones cross is represented by the annealing function, which depends on the

| 1 | 3 | 6 | 2 | 5 | 4 |

FIGURE 1: Genotype example.

queen's fitness, the drone's fitness, and the queen's speed which is given as follows:

$$p(q, d) = \min\left\{1, e^{(-l(q,d)/s(q))}\right\}, \tag{6}$$

where $l(q, d) = \text{dist}(f(q), f(d))$ corresponds to the absolute difference between the fitness of the queen and the drone, $s(q)$ is the speed of the queen at that instant.

The probability of mating is greater at the beginning of the flight since the queen's energy is greater, and it is more likely the drone's fitness is equivalent to the queen's.

As mentioned above, after each transition, speed and energy must be reduced. The speed must be adjusted according to the equation:

$$s(t + 1) = \alpha \cdot s(t), \tag{7}$$

where α is a factor in the interval $[0, 1]$.

When energy is reduced, the following equation is used:

$$e(t + 1) = e(t) - \text{step}, \tag{8}$$

where step corresponds to the energy reduction, whose value is calculated according to the following equation:

$$\text{step} = \frac{0, 5 \cdot e(0)}{M}. \tag{9}$$

M is the capacity of the spermatheca.

Figure 1 shows the five main stages of MBO: (1) mating flight in which the queen probabilistically selects the drones she will mate with and adds their sperm to her spermatheca; (2) creation of larvae by crossing between the genes of the queen and the drones; (3) using workers (heuristics) to improve the solution represented by the larvae; (4) update the worker's fitness according to how well they improve the larvae; and (5) replace the worst queens with the best larvae.

Abbass [31, 71] only considered a single queen and a single worker to solve a particular class of the satisfaction problem (3-SAT) and compared the results obtained with the WalkSAT and GSAT heuristics. MBO with the worker WalkSAT was more satisfying than WalkSAT alone, and so was MBO with the GSAT worker. The whole experiment was developed with parameters taken from biology.

Teo and Abbass [72] changed the accepted trajectory between the drone and the queen. This time, the drone was accepted, and its sperm was added to the queen's spermatheca if it was fitter than the queen or fulfilled the new annealing function which is given as follows:

$$p(q, d) = e^{(\Delta(f)/s(q))}, \tag{10}$$

where $\Delta(f)$ corresponds to the difference between the fitness of the new trajectory and the queen and $s(q)$ is the speed of the queen at that instant.

Later, Abbass and Teo [36] changed the annealing function. Instead of using the difference between the queen's

fitness and the drone, they used the difference between the drone trajectories' fitness to ensure that a trajectory was only accepted if it was better than the previous ones used. The new annealing function was determined by the following equation:

$$p(q, d) = e^{(-(f(t) - f(t-1))/s(q))},\qquad(11)$$

where $(f(t) - f(t-1))$ corresponds to the difference in the fitness of the current trajectory and the previous trajectory and $s(q)$ is the speed of the queen at that instant.

The study addressed the 3-SAT problem using five workers: GSAT, random walk, probabilistic greedy, one-point crossover, and WalkSAT. The tests were conducted by taking each heuristic separately, using MBO with each worker, and using the workers together in MBO. The results showed that MBO behaves better when all the workers are used together and that the new modification [36] provides better results than the original version [31, 71] and the one proposed by [72].

2.2.2. State of the Art MBOs.

Phu-Ang and Thammano [73] proposed a new MBO-based memetic algorithm to solve the flexible workshop scheduling problem. They introduced four new features to the standard MBO algorithm to move the search away from the local optimum. Their proposed algorithm maintained two populations: the female and male, and their performance was compared with the dispersion search algorithms using path linkage, hybrid differential evolution, hybrid harmony search algorithms, and hybrid GA. MBO outperformed others on large and complex problem cases.

Wen et al. [74] proposed a modified HBMO (MHBMO) to integrate process planning and scheduling (IPPS) with uncertain processing time and due date based on the fuzzy set by designing an uncertainty measurement objective calculation method. Their method solves the IPPS of multiple objectives effectively.

Palominos et al. [75] proposed a new MBO metaheuristic to deal with shop flow scheduling problems by designing different mating flight spaces and achieved excellent results over the 120 instances they tested.

Solgi and Loáiciga [76] identified seven algorithms for solving continuous optimization problems and showed that ABC, bee evolution for genetic algorithms (BEGA), and MBO were the most efficient.

Vakil-Baghmisheh and Salim [77] proposed a modified version of MFMHBO and compared it with ABC, QBE, and FMHBO in four reference functions for various variables up to 100. MFMHBO was faster than the others in most cases, especially for the Griewank and Schwefel functions, and increased precision and the number of variables. In general, their modified algorithm is comparable to the other algorithms.

Prabhakar and Lee [78] studied different MBO techniques for transformation-based three-level character selection using wavelets for prostate cancer classification.

Çelik and Ülker [79] tried to solve TSP using MBO. TSP had previously been solved using heuristic methods (GA and SA), but their MBO algorithm converged toward the optimal solution in fewer iterations.

Nayak and Naik [80] presented a hybrid metaheuristic pi-sigma neural network (PSNN) based on honey bee mating that successfully solved the data mining classification problem. Their approach combined HBMO with PSNN and compared it with other techniques such as GA, differential evolution (DE), and PSO. Their approach was stable and reliable and provided better classification precision than others.

Yin et al. [81] designed an accurate and stable MBO algorithm for programming a scorching machine to minimize the weighted sum of work completion times.

Zanbouri and Jafari Navimipour [82] proposed creating cloud computing services where services were selected based on trust and quality (QoS). They used the media grouping algorithm k to reduce the search space and HBMO to achieve global optimization. Their method worked efficiently in terms of computational time, producing more reliable services, although it was not efficient for large-scale data sets.

Arun and Vijay Kumar [83] proposed a modified MBO for a view selection algorithm (MBOVSA) to select views with the lowest total cost and information processing time in a large data warehouse. MBOVSA selected better quality views than HRUA, one of the important greed-based view selection algorithms.

2.3. Experimental Settings

2.3.1. TSP Modelling using MBO.
Our technique is based on [30]. We implemented Abbass' proposal with some modifications in TSP. The MBO metaheuristic is described in Algorithm 1.

2.3.2. Defining Parameters.
The proposed algorithm requires a series of parameters: (1) number of queen bees; (2) number of drones; (3) capacity of the queen's spermatheca; (4) number of flights; (5) number of larvae; (6) initial energy; and (7) initial speed of the queen. Particular care was taken when choosing (7) (defined experimentally) because the speed influences the annealing function, which selects the genetic material of the drone. When (7) is low, it is more likely that a drone will not be chosen for a flight, and if it is high, it is more likely that all drones will be accepted. Additionally, since the fitness range can be broad in TSP, one should not select a fixed speed for all instances to avoid the above-mentioned problems. Therefore, (7) must be calculated considering the difference between the best and the worst queen when starting the algorithm. For parameter (6), values from previous studies published in [30] were used for the energy reduction and speed factors. Table 1 shows the parameter values used for the proposed MBO algorithm.

2.3.3. Representing the Solution.
The genotype stereotype will be the same for queens, drones, and larvae and corresponds to a sequence of cities to be visited by the traveler. The genotype length is the number of cities that must be

Define the number of queens Q, Workers W, larvae B, mating flights G, and spermatheca size
Define the energy and speed of the queens
Assign each worker with a different heuristic
Assign the genotype of the queens randomly
Choose a worker randomly and apply its genotype to the queen's genotype
While the number of mating flights $<= G$
For queen = 1 through Q
Assign queen power and speed
Assign step = $0.5 \times$ energy$/M$
Assign path randomly
As long as energy> 0 and the spermatheca is not full
The queen moves between different states and probabilistically chooses drones
If a drone is selected, then
Add your sperm to the queen's spermatheca
End
Upgrade energy and speed of the queen
While
End For
Generate larva by crossing and mutation
Use workers to improve larva
Update the fitness of the workers
While the best of larvae is better than the worst of queens
Replace the worst queen with the best larva
Remove the best larva from the larvae list
End while
Remove all larva
End While

ALGORITHM 1: MBO Metaheuristics.

TABLE 1: Algorithm parameters.

Parameters	Source of obtaining	Value
Number of queen bees	Experimental	3
Drone quantity	Experimental	200
Spermatheque capacity	Experimental	50
Number of flights	Experimental	100
Number of larvae	Experimental	30
Initial energy	[84]	0.9
Initial velocity	Experimental	Difference between the best and the worst queen at the start of the algorithm.
Energy reduction factor	[30]	$step = (0.5 \cdot e_0)/M$
Speed reduction factor	[30]	$\alpha = 0.9$

stopped by and varies according to the problem. Each gene contains the cities and the corresponding order. For example, if there is a three in position four, the city three will be after the fourth.

Figure 2 shows an example solution for a problem with six cities where the optimal order must be 1, 3, 6, 2, 5, and 4.

Each genotype has an associated fitness that corresponds to the tour distance.

2.3.4. Generating an Initial Population of Bees, Queens, and Drones. The initial generation of queens can be selected in three diverse ways. The first two queens are generated on the basis of two tour construction heuristics for TSP: the nearest neighbor [85] and the closest insertion [86]. The remaining queens and drones are randomly generated to introduce diversity [36].

2.3.5. Selection of Drone Genetic Material. Drone genetic material is selected by the MBO annealing function [30]. Drone (d) selection is more likely when its fitness is like the queen (q) and when the queen begins her flight.

A random number between 0 and 1 is generated to select the drone. When this number is less than or equal to the one provided by the annealing function, the drone's genetic material is added to the queen's spermatheca (if the random number $\leq p(q, d)$, then add genetic material from the drone to the queen's spermatheca).

2.3.6. Mating. After completing a flight, each queen returns to the nest to lay eggs. When all the queens have returned, the genetic materials used to generate larvae are selected. The number of larvae to be generated is an algorithm parameter,
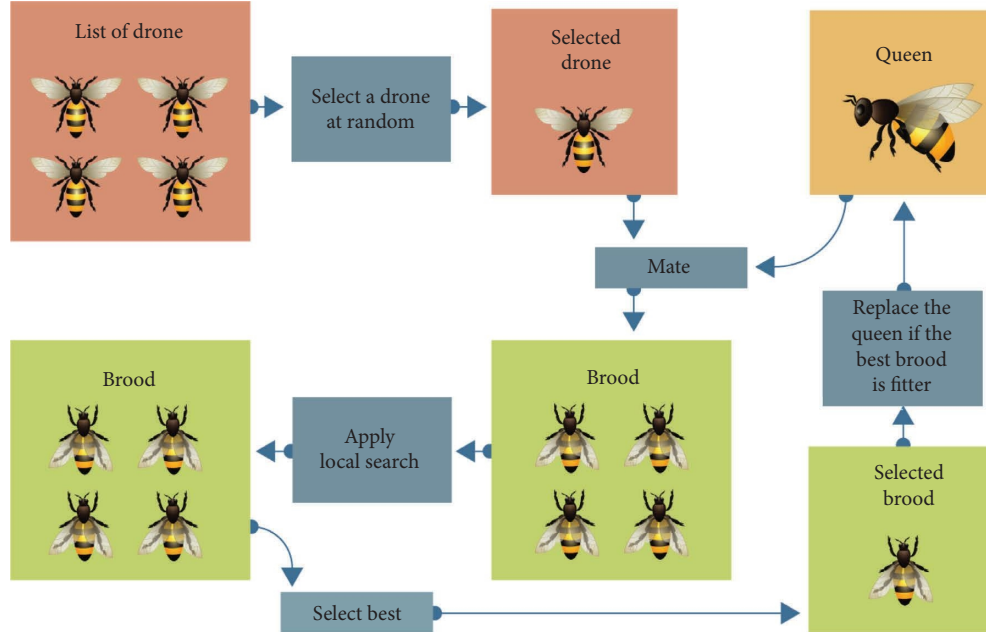
FIGURE 2: HBMO algorithm [71].

and each larva can come from any queen and any drone possessed by the chosen queen.

Queens are chosen by fitness, i.e., fitter queens are more likely to be selected. Drone genetic material is picked up randomly, and larvae are generated by crossing.

The crossing steps are as outlined below:

(1) Drones can only contribute half their genetic material because they are haploid [30, 87, 88]. Half of the drone's genes tag like this to ensure.

(2) A random interval and segment (i, j), where $i < j$. i represents the beginning of the segment, and j represents the end.

(3) Each queen's genes are checked, and they will be selected if they were not considered in the marked drone's interval. The set of genes chosen by the queen generates a list.

(4) The first selected $i - 1$ queen genes become part of the first $i - 1$ genes of the larva.

(5) The unlabeled genes in the marked drone segment (i, j) become part of the larva at the same position as the original gene.

(6) The queen's gene that has not been considered goes back in the same position as the larva.

(7) From position $j + 1$ the larva contains all other non-considered queen genes in the same order.

Figure 3 shows an example of the crossing process between a drone and a queen.

2.3.7. Larva Mutation.
The larva mutates after being formed by the following steps:

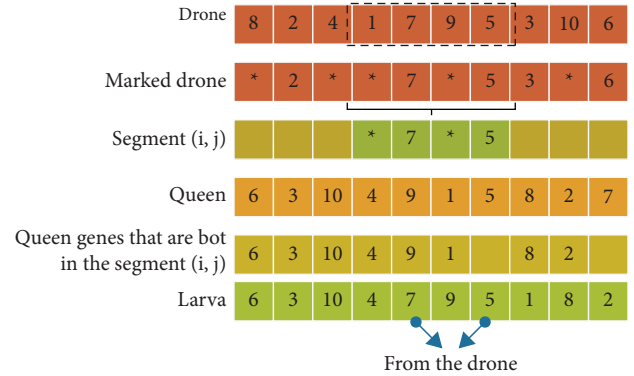(1) Each gene is visited, and a random number between 0 and 1 is generated.

FIGURE 3: Crossing.

(2) The gene is marked if the random number is less than or equal to a mutation probability.

(3) Marked genes are mutated.

We defined the mutation probability defined as 1% per [71].

2.3.8. Breeding Larvae Using Workers.
We considered three types of workers corresponding to SA [89], path-based tabu search neural network (TSNN) [90], and swap local search (SLS) [91]. TSP created crossovers in our solutions, so we included an additional worker called the 2-opt [92] heuristic to avoid them. The assignment of this worker offers the solutions provided by the other three workers who were not chosen.

2.3.9. Updating the Queen Bee Population.
The queen population is updated after a flight is finished. We use the same process as the original MBO algorithm (Algorithm 2).

> While the best of larvae is better than the worst of queens
>   Replace the worst queen with the best larva
>   Remove the best larva from the larvae list
> End while

ALGORITHM 2: Update the original queen population.

Chaotic initialization, giving an initial value to z_0
Generate the chaotic variables z_{ki}, $i = 1, 2, \ldots, n$, using the logistic function
The initial solution $x_0 = (x_{01}, x_{02}, \ldots, x_{0n})$ is produced by the formula $x_{0i} = a_i + (b_i - a_i) \times z_{ki}$, $i = 1, \ldots, n$.
Initialize maximum temperature T_{max}, minimum temperature T_{min} and the number of iterations L_{max}
$T = T_{max}$
$m = 0$
Set as the best solution $x^* = x_0$
Set as the best fitness $f^* = f(x^*)$
While $(T > T_{min})$
While $(m \leq L_{max})$
Generate a new solution $y_m = (x_{m1}, x_{m2}, \ldots, x_{mn})$ based on the formula $y_{mi} = x_{mi} + \alpha \times (b_i - a_i) \times z_{km}$, picking out i randomly and z_{km}
is delivered by the logistics map
$\alpha = \alpha \times e^{-\beta}$
   $\Delta E^* = f(y_m) - f^*$
   $\Delta E = f(y_m) - f(x_m)$
If $(\Delta E^* \leq 0)$
$x^* = y_m$
$f^* = f(x^*)$
If $(\Delta E \leq 0)$
$x_{m+1} = y_m$
$f^* = f(x^*)$
If $(\Delta E > 0)$
Accept status with probability $e^{(-\Delta E/T)}$
$m = m + 1$
End while
$L_{max} = L_{max} + d$, $m = 0$
$T = \delta \times T$
End while
Deliver the best solution found x^* and best fitness f^*

ALGORITHM 3: Pseudocode for CSA.

The solutions with the fittest queens and larvae generate new queens. The remaining solutions are discarded but can be considered in the next iteration. A new flight begins after the queen population has been updated, unless the stop criterion is met.

2.3.10. Chaotic MBO Modeling.

Chaos can be a useful tool for finding better solutions, but which MBO processes would benefit from chaos theory?

The parts of the MBO metaheuristic (Algorithm 1) must be considered, and a proposal to introduce chaos must be defined to answer this question. Chaos has successfully created initial solutions in other heuristics [93, 94].

Drones are generated like queens, for which chaos better explores the solution space.

In MBO, the workers improve the solutions. The effect of chaos cannot be evaluated if new chaotic heuristics replace MBO ones. Instead, chaos must be introduced into specific parts of the workers.

We carried out different types of tests to measure the performance of chaos in the different MBO stages: (1) tests with traditional MBO; (2) tests with the use of chaos only in the generation of initial solutions (queens and drones); (3) tests with the use of chaos only in the workers; and (4) tests with the use of chaos in the generation of initial solutions and the workers.

2.3.11. Chaotic Generation of Initial Queen and Drone Populations.

The initial populations of queen bees and the generation of the drone population are generated using chaos. One queen is born using the nearest neighbor heuristic, and another uses the closest insertion heuristic. There are two heuristics in both versions to ensure good solutions and improve the algorithm's performance.
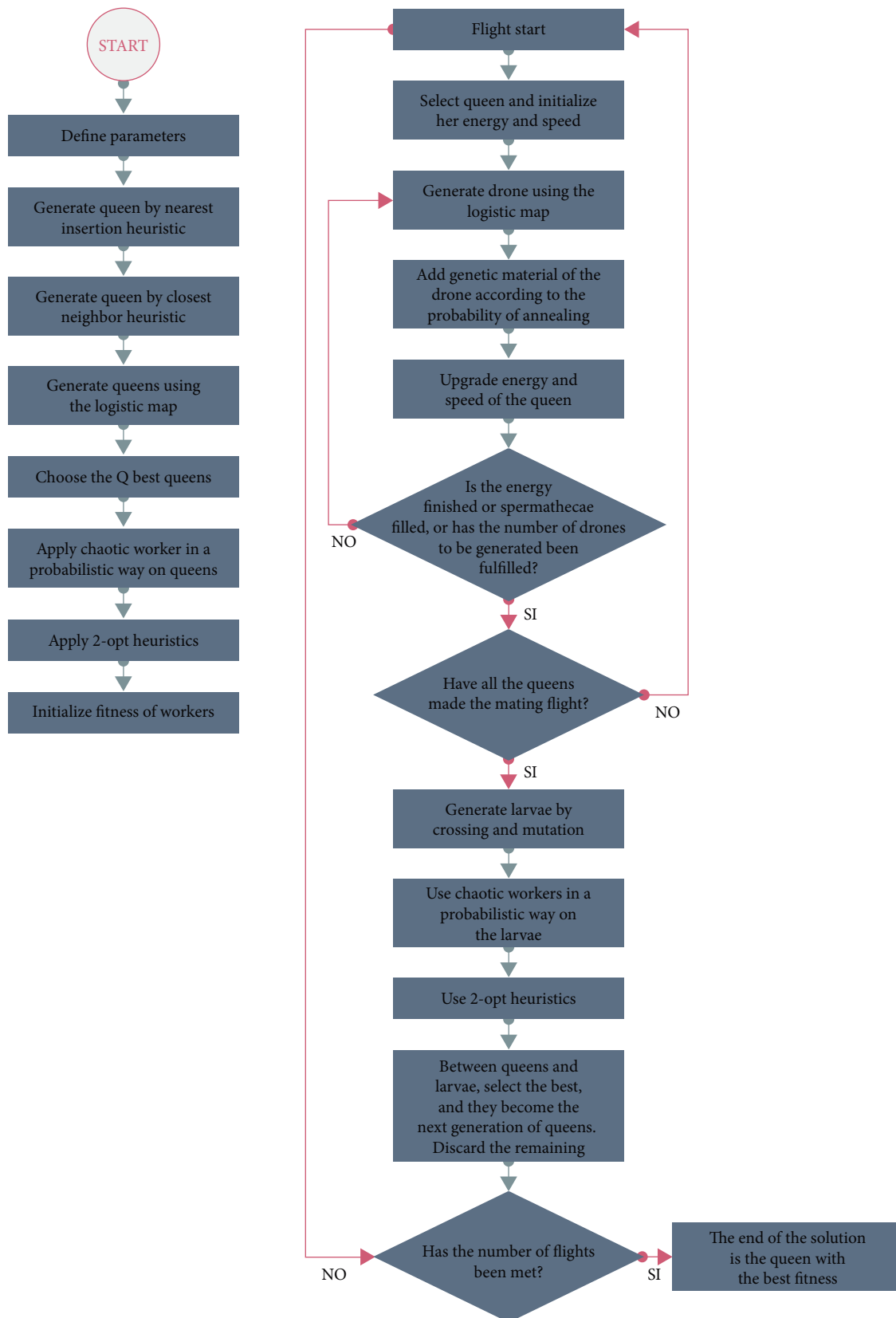
Figure 4: Chaotic MBO algorithm flow chart.

The remaining queens are born using chaotic maps from a randomly initialized solution vector where each gene takes a value between 0 and 1 for each value within the logistic map solution as shown in the following equation:

$$X_{n+1} = rX_n(1 - X_n), \tag{12}$$

where r is the control parameter (in this case, $r = 4$), whose value ensures that the system will enter a state of chaos. r has values between 0 and 1, except for the fixed points of 0.25, 0.5, and 0.75.

When a queen starts her flight, the first drone drops randomly and has a value between 0 and 1. The remaining drones are on the logistic map the same way as the queens.

2.3.12. Chaotic Breeding of Larvae using Workers.
After generating the queens and larvae by crossing and mutations, the workers improve the solution quality.

We use the same three workers with chaotic modifications (chaos simulated annealing (CSA), chaotic tabu search neural network based on a path (CTSNN), and chaotic swap local search (CSLS). Each worker performs a job that leads to better solutions.

We included a chaotic version of the traditional MBO to avoid crossing TSPs in the solutions. This 2-opt heuristic uses the output of any chaotic worker mentioned above. The chaotic workers use the same fitness value at the beginning of the algorithm.

2.3.13. Chaos Simulated Annealing (CSA) Heuristic.
The first chaotic worker follows the same idea of simulated annealing (SA) in the traditional MBO modeling but with chaotic alterations (CSA).

CSA was created by Mingjun and Huan Wen [95] to take advantage of SA in combinatorial optimization problems and the improvements it brings to chaos theory.

The problem:

$$(P) \begin{cases} \min f(x), \\ x_i \in [a_i, b_i], \quad i = 1, 2, \ldots, n. \end{cases} \tag{13}$$

CSA and SA have the same structure but differ in two ways. The first difference is when creating an initial solution $x_0 = (x_{01}, x_{02}, \ldots, x_{0n})$ using the logistic map to obtain z_{ki}, according to following equation:

$$x_{0i} = a_i + (b_i - a_i) \cdot z_{ki}, i = 1, \ldots, n. \tag{14}$$

The second difference is when selecting a neighboring solution $y_m = (x_{m1}, x_{m2}, \ldots, x_{mn})$, where one of the current solution variables is random and it's modified using the following logistic map again:

$$y_{mi} = x_{mi} + \alpha \cdot (b_i - a_i) \cdot \leq z_{km}, \tag{15}$$

where α is a chaos weighting parameter.

Algorithm 3 shows the CSA heuristic proposed by [95].

We worked on chaotic genotypes where each gene has a chaotic variable between 0 and 1. However, not all solutions have a chaotic genotype, such as the larvae born after

TABLE 2: Description of the instances.

Instance	Number of cities
KroA100	100
KroB100	100
KroC100	100
KroD100	100
KroE100	100
KroA150	150
KroB150	150
d198	198
KroA200	200
KroB200	200
a280	280
pr299	299
lin318	318
rd400	400
fl417	417
pcb442	442
d493	492
d657	657
rat783	783
fl1577	1577

applying the crossover and mutation operators, which generate a solution based on the traditional MBO. The genotype is a chaotic solution.

Furthermore, chaotic MBO uses the same parameters as traditional MBO. Still, it is necessary to define the value of the chaos weighting (α), which we determined as one by experimentation.

It is also essential to define the chaotic function used, which is determined by the original CSA study and corresponds to the logistic map:

$$z_{k+1} = 4z_k(1 - z_k). \tag{16}$$

2.3.14. Path-Based Chaotic Tabu Search Neural Network Heuristic.
The second worker used for traditional MBO is path-based TSNN. Aono et al. [96] proposed a method based on the studies of [90], which incorporates chaotic noise into the equations.

The new algorithm also relies on 2-opt moves to generate its TSP solutions. The new chaotic noise procedure is based on equations (17)–(19).

$$\xi_{ij}(t + 1) = \beta\Delta_{ij}(t), \tag{17}$$

$$\zeta_{ij}(t + 1) = -\alpha \sum_{d=0}^{s-1} k_r^d x_{ij}(t - d), \tag{18}$$

$$\begin{aligned} x_{ij}(t + 1) = \xi_{ij}(t + 1) + \zeta_{a(i)a(j)}(t + 1) \\ + \zeta_{ij}(t + 1) + \gamma z_{ij}(t + 1), \end{aligned} \tag{19}$$

where Δ_{ij} is the difference between the fitness of the current solution (D_0) and the solution with a 2-opt movement between cities i and j (D_{ij}): $\Delta_{ij}(t) = D_0(t) - D_{ij}(t)$, β is the weight given to $\Delta_{ij}(t)$, α is the weighting parameter of the

TABLE 3: Metaheuristic results MBO_1, MBO_2, MBO_3, MBO_4.

Instance	Value optimum	MBO 1	Error %	MBO 2	Error %	MBO 3	Error %	MBO 4	Error %
KroA100	21282	21285.4	0.02	21285.4	0.02	21285.4	0.02	21285.4	0.02
KroB100	22141	22150.7	0.04	22139.1	−0.01	22139.1	−0.01	22139.1	−0.01
KroC100	20749	20750.8	0.01	20750.8	0.01	20750.8	0.01	20750.8	0.01
KroD100	21294	21297.3	0.02	21294.3	0.00	21294.3	0.00	21294.3	0.00
KroE100	22068	22113.7	0.21	22068.8	0.00	22068.8	0.00	22086.5	0.08
KroA150	26524	26606.9	0.31	26601.3	0.29	26535.1	0.04	26535.2	0.04
KroB150	26130	26155.8	0.10	26148.0	0.07	26137.8	0.03	26133.6	0.01
d198	15780	15848.8	0.44	15846.6	0.42	15823.7	0.28	15821.8	0.27
KroA200	29368	29452.2	0.29	29515.0	0.50	29433.9	0.22	29432.5	0.22
KroB200	29437	29583.3	0.50	29545.8	0.37	29539.6	0.35	29516.4	0.27
a280	2579	2603.0	0.93	2603.5	0.95	2588.5	0.37	2588.5	0.37
pr299	48191	48578.2	0.80	48467.8	0.57	48409.5	0.45	48389.6	0.41
lin318	42029	42649.6	1.48	42522.4	1.17	42513.1	1.15	42331.1	0.72
rd400	15281	15637.9	2.34	15651.0	2.42	15582.8	1.98	15590.5	2.03
fl417	11861	11961.7	0.85	11962.2	0.85	11931.1	0.59	11940.0	0.67
pcb442	50779	51748.4	1.91	51953.0	2.31	51586.9	1.59	51657.0	1.73
d493	35002	35689.1	1.96	35670.0	1.91	35478.2	1.36	35589.9	1.68
d657	48912	50430.2	3.10	50301.2	2.84	50298.8	2.84	50334.1	2.91
rat783	8806	9194.0	4.41	9186.8	4.32	9172.1	4.16	9236.5	4.89
fl1577	22249	22635.8	1.74	22696.0	2.01	22732.2	2.17	22653.4	1.82
Average error			1.07		1.05		0.88		0.91

taboo effect, k_r is the decay parameter of the taboo effect, $x_{ij}(t)$ is the output of neuron i, j in time t, and γ is the weight given to the chaotic noise z_{ij}.

According to Aono et al. [96], the logistic map should generate z_{ij} in a chaotic form, as shown in the following equation:

$$z_{ij}(t + 1) = 3.828 z_{ij}(t)\left(1 - z_{ij}(t)\right). \qquad (20)$$

The same parameters can be used for both the path-based TSNN and its chaotic version (CTSNN). The tests show different values of γ in the interval [20, 30000], generating the best results for $\gamma = 10000$.

2.3.15. Chaotic Swap Local Search Heuristic. Like the chaotic heuristic described above, chaotic noise gets the worker to swap local search and creates the chaotic swap local search.

To develop this new method, all swap movements (i, j), where $(i, j) = (j, i)$ e $i \neq j$, as in the following equation:

$$\xi_{ij}(t + 1) = \Delta_{ij}(t) + \gamma z_{ij}(t), \qquad (21)$$

where Δ_{ij} is the difference between the values of the current solution's objective function (D_0) and the solution with a swap movement of cities i and j (D_{ij}): $\Delta_{ij}(t) = D_0(t) - D_{ij}(t)$ and γ is the weight given to the chaotic noise z_{ij}.

The next movement selected corresponds to the one with the highest value of ξ_{ij}, which is not necessarily the one that delivers the best fitness because of the chaotic variable, but the one with the swap movement between cities i and j. Finally, the solution with the best fitness is generated when all iterations are carried out.

This worker uses the same parameters as the nonchaotic version, adding chaotic noise through the following equation:
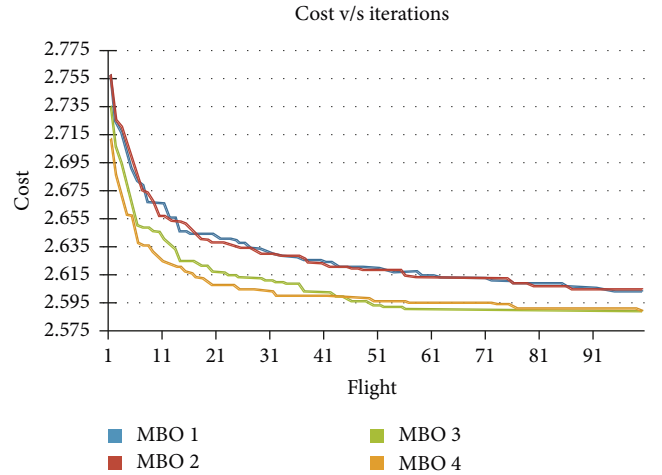

FIGURE 5: Convergence curve for instance d198.

$$z_{ij}(t + 1) = 3.828 z_{ij}(t)\left(1 - z_{ij}(t)\right). \qquad (22)$$

We selected the chaos weighting experimentally, taking values of γ in the interval [20, 100] and obtaining the best results for $\gamma = 40$.

2.3.16. Chaotic MBO Flow Chart. Figure 4 shows a flow chart describing the chaotic MBO metaheuristic procedure described above to solve TPS.

3. Results

We carried out the computational experiment in the programming language C and extracted the used instances from the TSPLIB database [97] (Table 2). We solved each instance using the traditional MBO (MBO_1) using a common

Table 4: ANOVA for the kroE100 instance.

Origin of variations	Sum of squares	Degrees of freedom	Average of the squares	F	P value	Critical value for F
Between groups	6772,7963	3	2257,59877	6,82512997	0,00357222	3,23887152
Within groups	5292,43845	16	330,777403	—	—	—
Total	12065,2347	19	—	—	—	—

Table 5: Tukey's test for the kroE100 instance.

Type	N	Subset for alpha = .05	
		1	2
MBO$_2$	5	22068,8	—
MBO$_3$	5	22068,8	—
MBO$_4$	5	22086,52	22086,52
MBO$_1$	5	—	22113,74
p-value		0,43708573	0,12380896

Table 6: ANOVA for the d198 instance.

Origin of variations	Sum of squares	Degrees of freedom	Average of the squares	F	P value	Critical value for F
Between groups	3136,37901	3	1045,45967	7,45520814	0,00242198	3,23887152
Within groups	2243,71398	16	140,232124	—	—	—
Total	5380,09299	19	—	—	—	—

Table 7: Tukey's test for the d198 instance.

Type	N	Subset for alpha = .05	
		1	2
MBO$_4$	5	15821,84	—
MBO$_3$	5	15823,74	—
MBO$_2$	5	—	15846,64
MBO$_1$	5	—	15848,82
P value		0,99403698	0,99107353

pseudorandom number generator (PRNG) in the evolutionary operators and three chaotic MBO variants: (1) MBO with chaos in the initial solution generation (MBO$_2$); (2) MBO with chaos in the workers (MBO$_3$); and (3) MBO with chaos in the initial solution generation and the workers (MBO$_4$). We executed each instance five times under the parameters already described in the previous points to find an average error for each one.

The solution quality is in equation (7), which is the percentage error between the average value of the solutions found (PSE) for the pure MBO and the three chaotic variants with the best-known solutions (BKS) of the TSPLIB as shown in the following equation:

$$\text{Percent Error} = \frac{(PSE - BKS)}{BKS} * 100. \quad (23)$$

The results for the four metaheuristics are shown in Table 3. MBO$_3$ and MBO$_4$ provided the best results. However, ANOVA and Tukey statistical analyzes (with a 95% confidence interval) revealed that 30% of the instances were significantly different across the four methods applied.

Additionally, we found the chaotic modifications improved convergence speed (Figure 5). MBO$_3$ and MBO$_4$ converged faster than MBO$_1$.

3.1. Discussion of the Results. The calculated percentage error of pure MBO was 1.07% when it was compared with the results of 20 instances with the best values found in the literature [98]. When generating solutions, MBO with chaos in the workers and MBO with chaos in both have percentage errors of 1.05%, 0.88%, and 0.91%, respectively.

In the pure MBO, the TSNN worker had the highest efficiency based on routes, followed by SA and later SLS. The CTSNN worker-generated the best results in the versions with chaos based on routes, then CSLS and CSA.

At first sight, chaos improves solution quality when analyzed. The ANOVA analysis revealed six instances of significant differences between the four methods. The six TSP instances compared with the TSPLIB library are kroE100, d198, a280, pr299, fl417, and pcb442. Twenty experiments per metaheuristic were performed for each one.

Following the results of the ANOVA test, the Tukey nonparametric evaluation was used to compare the two groups of metaheuristics. The results of the Tukey non-parametric test in five out of six instances show a statistical difference between metaheuristics with chaos and metaheuristics with MBO1. The results of the ANOVA test and the Tukey nonparametric test are shown in Tables 4–15.

TABLE 8: ANOVA for the a280 instance.

Origin of variations	Sum of squares	Degrees of freedom	Average of the squares	F	P value	Critical value for F
Between groups	1081,4057	3	360,468567	39,0980787	$1,3599E-07$	3,23887152
Within groups	147,513567	16	9,21959795	—	—	—
Total	1228,91927	19	—	—	—	—

TABLE 9: Tukey's test for the a280 instance.

Type	N	Subset for alpha = 0.05	
		1	2
MBO$_3$	5	2588,52	—
MBO$_4$	5	2588,54	—
MBO$_1$	5	—	2602,98
MBO$_2$	5	—	2603,46
$p$value		0,99999957	0,99426841

TABLE 10: ANOVA for the pr299 instance.

Origin of variations	Sum of squares	Degrees of freedom	Average of the squares	F	P value	Critical value for F
Between groups	107695,484	3	35898,4947	3,5938604	0,03703161	3,23887152
Within groups	159821,432	16	9988,83951	—	—	—
Total	267516,916	19	—	—	—	—

TABLE 11: Tukey's test for the pr299 instance.

Type	N	Subset for alpha = 0.05	
		1	2
MBO$_4$	5	48389,56	—
MBO$_3$	5	48409,5	48409,5
MBO$_2$	5	48467,78	48467,78
MBO$_1$	5	—	48578,2
p value		0,61333354	0,07194982

TABLE 12: ANOVA for the fl417 instance.

Origin of variations	Sum of squares	Degrees of freedom	Average of the squares	F	P value	Critical value for F
Between groups	3678,40891	3	1226,1363	11,7611591	0,00025472	3,23887152
Within groups	1668,04826	16	104,253016	—	—	—
Total	5346,45717	19	—	—	—	—

TABLE 13: Tukey's test for the fl417 instance.

Type	N	Subset for alpha = 0.05	
		1	2
MBO$_3$	5	11931,1	—
MBO$_4$	5	11940,04	—
MBO$_1$	5	—	11961,74
MBO$_2$	5	—	11962,22
p value		0,52713516	0,99984659

Table 14: ANOVA for the pcb442 instance.

Origin of variations	Sum of squares	Degrees of freedom	Average of the squares	F	P value	Critical value for F
Between groups	378562,522	3	126187,507	3,43694377	0,04223558	3,23887152
Within groups	587440,544	16	36715,034	—	—	—
Total	966003,066	19	—	—	—	—

Table 15: Tukey's test for the pcb442 instance.

Type	N	Subset for alpha = 0.05	
		1	2
MBO_3	5	51586,96	—
MBO_4	5	51657	51657
MBO_1	5	51748,44	51748,44
MBO_2	5	—	51953,04
p value		0,55674431	0,10864054

The results show that chaotic functions applied to the MBO metaheuristic significantly improve solutions regarding the TSP. All the solutions are equal to or greater than those obtained with pure MBO.

4. Conclusions

The research and analysis of new methods for solving difficult optimization problems are of great significance for this type of concern. When introduced, chaos theory in MBO metaheuristics, specifically when generating solutions and in the workers, created four algorithms. The first considers traditional MBO without chaos; the second only uses chaos when generating solutions; the third uses chaos in the workers; and the fourth uses chaos when generating solutions and in the workers. In this study, TSP was used to measure the performance of each method.

This research demonstrates the ability to improve performance using chaos theory. However, it is impossible to extrapolate the results obtained with this chaotic MBO heuristic to all engineering problems. The performance of each heuristic is closely related to the solution area of the problem; for the TSP problem, the results show better performance by introducing chaos to the general MBO model. The technique improved the global search capability of the MBO3 heuristic with a chaotic modification of the worker bees by concentrating on the highest number of high-performance solutions.

Future work should focus on connecting MBO with chaos theory and using it in other stages of MBO, such as the mutation that increases larvae, the drone quality of life before mating, and new chaotic heuristics in the workers. New versions can be used by the technique using parallel distributions or by hybridizing with different chaotic mappings to improve performance.

Data Availability

The original data used to support the findings of this study can be obtained from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

Acknowledgments

References

[1] F. Glover, "Future paths for integer programming and links to artificial intelligence," *Computers & Operations Research*, vol. 13, no. 5, pp. 533–549, 1986.

[2] B. Alatas and H. Bingol, "A physics based novel approach for travelling tournament problem: optics inspired optimization," *Information Technology and Control*, vol. 48, no. 3, pp. 373–388, 2019.

[3] S. Akyol and B. Alatas, "Plant intelligence based metaheuristic optimization algorithms," *Artificial Intelligence Review*, vol. 47, no. 4, pp. 417–462, 2017.

[4] B. Alatas and H. Bingol, "Comparative assessment of light-based intelligent search and optimization algorithms," *Light & Engineering*, vol. 28, pp. 51–59, 2020.

[5] H. Bingol and B. Alatas, "Chaos based optics inspired optimization algorithms as global solution search approach," *Chaos, Solitons & Fractals*, vol. 141, Article ID 110434, 2020.

[6] E. Bastianetto, S. Ceschia, and A. Schaerf, "Solving a home energy management problem by simulated annealing," *Optics Letters*, vol. 15, no. 5, pp. 1553–1564, 2020.

[7] C. Ghnatios, R. M. Hage, and I. Hage, "An efficient tabu-search optimized regression for data-driven modeling," *Comptes Rendus Mecanique*, vol. 347, no. 11, pp. 806–816, 2019.

[8] S. Dávila, M. Alfaro, G. Fuertes, M. Vargas, and M. Camargo, "Vehicle routing problem with deadline and stochastic service times: case of the ice cream industry in Santiago city of Chile," *Mathematics*, vol. 9, no. 21, p. 2750, 2021.

[9] S. Gupta and K. Deep, "A novel random walk grey wolf optimizer," *Swarm and Evolutionary Computation*, vol. 44, pp. 101–112, 2019.

[10] S. Tarwani and A. Chug, "Investigating optimum refactoring sequence using hill-climbing algorithm," *Journal of Information and Optimization Sciences*, vol. 41, no. 2, pp. 499–508, 2020.

[11] J. C. Bansal, H. Sharma, S. S. Jadon, and M. Clerc, "Spider monkey optimization algorithm for numerical optimization," *Memetic Computing*, vol. 6, no. 1, pp. 31–47, 2014.

[12] D. Dabhi and K. Pandya, "Enhanced velocity differential evolutionary particle swarm optimization for optimal scheduling of a distributed energy resources with uncertain scenarios," *IEEE Access*, vol. 8, Article ID 27001, 2020.

[13] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization: an overview," *Swarm Intell*, vol. 1, no. 1, pp. 33–57, 2007.

[14] B. Alatas, E. Akin, and A. B. Ozer, "Chaos embedded particle swarm optimization algorithms," *Chaos, Solitons & Fractals*, vol. 40, no. 4, pp. 1715–1734, 2009.

[15] M. Dorigo and T. Stützle, "Ant colony optimization: overview and recent advances," in *Handbook of Metaheuristics*, Springer, Canada, 2019.

[16] T.-P. Chung and F. Chen, "A complete immunoglobulin-based artificial immune system algorithm for two-stage assembly flowshop scheduling problem with part splitting and distinct due windows," *International Journal of Production Research*, vol. 57, no. 10, pp. 3219–3237, 2019.

[17] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.

[18] G. Fuertes, M. Alfaro, I. Soto, R. Carrasco, D. Iturralde, and C. Lagos, "Optimization model for location of RFID antennas in a supply chain," in *Proceedings of the International Conference on Computers Communications and Control*, pp. 203–209, Oradea, Romania, May 2018.

[19] J. Sabattin, G. Fuertes, M. Alfaro, L. Quezada, and M. Vargas, "Optimization of large electric power distribution using a parallel genetic algorithm with dandelion strategy," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 26, no. 5, pp. 2648–2660, 2018.

[20] X.-S. Yang, "Firefly algorithm, lévy flights and global optimization," in *Research and Development in Intelligent Systems XXVI* Springer, London, UK, 2010.

[21] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.

[22] D. Pham, S. Otri, E. Koc, A. Ghanbarzadeh, S. Rahim, and M. Zaidi, "Technical note: the bees algorithm," in *Manufacturing Engineering Centre*, Cardiff University, Cardiff, UK, 2005.

[23] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007.

[24] S. H. Jung, "Queen-bee evolution for genetic algorithms," *Electronics Letters*, vol. 39, no. 6, pp. 575-576, 2003.

[25] T. Sato and M. Hagiwara, "Bee system: finding solution by a concentrated search," in *Proceedings of the International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, pp. 3954–3959, Orlando, FL, USA, October 1997.

[26] P. Lucic and D. Teodorovic, "Bee system: modeling combinatorial optimization transportation engineering problems by swarm intelligence," in *Preprints of the TRISTAN IV Triennial Symposium on Transportation Analysis* Sao Miguel, Azores Islands, Portugal, 2001.

[27] D. Teodorović, "Bee colony optimization (BCO)," in *Studies in Computational Intelligence* Vol. 248, Springer, Berlin, Germany, 2009.

[28] H. F. Wedde, F. Muddassar, P. Thorsten et al., "BeeAdHoc: an energy efficient routing algorithm for mobile ad hoc networks inspired by bee behavior," in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 153–160, Washington DC, USA, June 2005.

[29] H. F. Wedde and M. Farooq, "BeeHive - new ideas for developing routing algorithms inspired by honey bee behavior," in *Handbook of Bioinspired Algorithms and Applications* Chapman & Hall/CRC, London, UK, 2005.

[30] H. A. Abbass, "MBO: marriage in honey bees optimization - a haplometrosis polygynous swarming approach," in *Proceedings of the International Congress on Evolutionary Computation*, pp. 207–214, Seoul, Korea, May 2001.

[31] H. A. Abbass, "A single queen single worker honey-bees approach to 3-SAT," in *The Genetic and Evolutionary Computation Conference*, Northcott Drive, Canberra ACT, 2600, Australia, 2001.

[32] O. Bozorg Haddad and A. Afshar, "MBO (marriage bees optimization), a new heuristic approach in hydrosystems design and operation," in *Proceedings of the 1st International Conference on Managing Rivers in the 21st Century: Issues and Challenges*, pp. 499–504, Penang, Malaysia, September 2004.

[33] O. B. Haddad, A. Afshar, and M. A. Mariño, "Honey-bees mating optimization (HBMO) algorithm: a new heuristic approach for water resources optimization," *Water Resources Management*, vol. 20, no. 5, pp. 661–680, 2006.

[34] C. Yang, J. Chen, and X. Tu, "Algorithm of fast marriage in honey bees optimization and convergence analysis," in *Proceedings of the International Conference on Automation and Logistics*, pp. 1794–1799, Jinan, China, August 2007.

[35] P. Curkovic and B. Jerbic, "Honey-bees optimization algorithm applied to path planning problem," *International Journal of Simulation Modelling*, vol. 6, no. 3, pp. 154–164, 2007.

[36] J. Teo and H. A. Abbass, "A true annealing approach to the marriage in honey-bees optimization algorithm," *International Journal of Computational Intelligence and Applications*, vol. 03, no. 02, pp. 199–211, 2003.

[37] A. S. Shamsaldin, T. A. Rashid, R. A. Al-Rashid Agha, N. K. Al-Salihi, and M. Mohammadi, "Donkey and smuggler optimization algorithm: a collaborative working approach to path finding," *Journal of Computational Design and Engineering*, vol. 6, no. 4, pp. 562–583, 2019.

[38] M. Alfaro, M. Vargas, G. Fuertes, and J. P. Sepúlveda-Rojas, "Proposal of Two Measures of Complexity Based on Lempel-Ziv for Dynamic Systems: An Application for Manufacturing Systems," *Mathematical Problems in Engineering*, vol. 2018, Article ID 8692146, 11 pages, 2018.

[39] M. Alfaro, G. Fuertes, M. Vargas, J. Sepúlveda, and M. Veloso-Poblete, "Forecast of Chaotic Series in a Horizon superior to the Inverse of the Maximum Lyapunov Exponent," *Complexity*, vol. 2018, Article ID 1452683, 9 pages, 2018.

[40] G. Fuertes, M. Vargas, I. Soto, K. Witker, M. Peralta, and J. Sabattin, "Project-based learning versus cooperative learning courses in engineering students," *IEEE Latin America Transactions*, vol. 13, no. 9, pp. 3113–3119, 2015.

[41] B. Alatas, "Chaotic harmony search algorithms," *Applied Mathematics and Computation*, vol. 216, no. 9, pp. 2687–2699, 2010.

[42] R. Ebrahimzadeh and M. Jampour, "Chaotic genetic algorithm based on lorenz chaotic system for optimization problems," *International Journal of Intelligent Systems and Applications*, vol. 5, no. 5, pp. 19–24, 2013.

[43] D. Ghosh and J. Singh, "Spectrum-based multi-fault localization using chaotic genetic algorithm," *Information and Software Technology*, vol. 133, Article ID 106512, 2021.

[44] M. Tahir, A. Tubaishat, F. Al-Obeidat, B. Shah, Z. Halim, and M. Waqas, "A novel binary chaotic genetic algorithm for feature selection and its utility in affective computing and healthcare," *Neural Computing & Applications*, vol. 34, no. 14, Article ID 11453, 2020.

[45] X. Wei, S. Yuan, and Y. Ye, "Optimizing facility layout planning for reconfigurable manufacturing system based on chaos genetic algorithm," *Production & Manufacturing Research*, vol. 7, no. 1, pp. 109–124, 2019.

[46] Y. Shen, "Improved chaos genetic algorithm based state of charge determination for lithium batteries in electric vehicles," *Energy*, vol. 152, pp. 576–585, 2018.

[47] G. Fuertes, M. Vargas, M. Alfaro, R. Soto-Garrido, J. Sabattin, and M. A. Peralta, "Chaotic genetic algorithm and the effects of entropy in performance optimization," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 29, no. 1, Article ID 013132, 2019.

[48] M. Vargas, G. Fuertes, M. Alfaro, G. Gatica, S. Gutierrez, and M. Peralta, "The Effect of Entropy on the Performance of Modified Genetic Algorithm Using Earthquake and Wind Time Series," *Complexity*, vol. 2018, Article ID 4392036, 13 pages, 2018.

[49] C. Archetti, D. Feillet, A. Mor, and M. G. Speranza, "Dynamic traveling salesman problem with stochastic release dates," *European Journal of Operational Research*, vol. 280, no. 3, pp. 832–844, 2020.

[50] L. Jiang, M. Dhiaf, J. Dong, C. Liang, and S. Zhao, "A traveling salesman problem with time windows for the last mile delivery in online shopping," *International Journal of Production Research*, vol. 58, no. 16, pp. 5077–5088, 2020.

[51] T. Derya, E. Dinler, and B. Keçeci, "Selective generalized travelling salesman problem," *Mathematical and Computer Modelling of Dynamical Systems*, vol. 26, no. 1, pp. 80–118, 2020.

[52] C. Pan, Q. Hong, and X. Wang, "A novel memristive chaotic neuron circuit and its application in chaotic neural networks for associative memory," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 40, no. 3, pp. 521–532, 2021.

[53] L. Lin, "Predefined-time Antisynchronization of Two Different Chaotic Neural Networks," *Complexity*, vol. 2020, Article ID 7476250, 11 pages, 2020.

[54] C. Long, G. Zhang, and Z. Zeng, "Novel results on finite-time stabilization of state-based switched chaotic inertial neural networks with distributed delays," *Neural Networks*, vol. 129, pp. 193–202, 2020.

[55] L. Chen, H. Yin, T. Huang, L. Yuan, S. Zheng, and L. Yin, "Chaos in fractional-order discrete neural networks with application to image encryption," *Neural Networks*, vol. 125, pp. 174–184, 2020.

[56] G. Srivastava, C. N. S. Vinoth Kumar, V. Kavitha, N. Parthiban, and R. Venkataraman, "Two-stage data encryption using chaotic neural networks," *Journal of Intelligent and Fuzzy Systems*, vol. 38, no. 3, pp. 2561–2568, 2020.

[57] C. Jiang and Q. Chen, "Construction of blind restoration model for super-resolution image based on chaotic neural network," *Chaos, Solitons & Fractals*, vol. 131, Article ID 109498, 2020.

[58] H. H. Lian, S. P. Xiao, Z. Wang, X. H. Zhang, and H. Q. Xiao, "Further results on sampled-data synchronization control for chaotic neural networks with actuator saturation," *Neurocomputing*, vol. 346, pp. 30–37, 2019.

[59] R. S. T. Lee, "Chaotic neural networks in quantum finance," in *Quantum Finance*Springer, Singapore, 2020.

[60] S. Lahmiri and S. Bekiros, "Cryptocurrency forecasting with deep learning chaotic neural networks," *Chaos, Solitons & Fractals*, vol. 118, pp. 35–40, 2019.

[61] S. Wang and X. Zhang, "Observer design for fractional-order chaotic neural networks with unknown parameters," *IEEE Access*, vol. 8, Article ID 120044, 2020.

[62] K. Chen, B. Xue, M. Zhang, and F. Zhou, "Novel chaotic grouping particle swarm optimization with a dynamic regrouping strategy for solving numerical optimization tasks," *Knowledge-Based Systems*, vol. 194, Article ID 105568, 2020.

[63] M. Rosalie, E. Kieffer, M. R. Brust, G. Danoy, and P. Bouvry, "Bayesian optimisation to select Rössler system parameters used in chaotic ant colony optimisation for coverage," *Journal of Computational Science*, vol. 41, Article ID 101047, 2020.

[64] T. S. L. V. Ayyarao, N. S. S. Ramakrishna, R. M. Elavarasan et al., "War strategy optimization algorithm: a new effective metaheuristic algorithm for global optimization," *IEEE Access*, vol. 10, Article ID 25073, 2022.

[65] B. Alatas, "ACROA: artificial chemical reaction optimization algorithm for global optimization," *Expert Systems with Applications*, vol. 38, no. 10, Article ID 13170, 2011.

[66] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, The MIT Press, London, U.K, 3rd ed edition, 2009.

[67] B. Yang, W. Li, J. Wang, J. Yang, T. Wang, and X. Liu, "A novel path planning algorithm for warehouse robots based on a two-dimensional grid model," *IEEE Access*, vol. 8, Article ID 80347, 2020.

[68] A. Bretin, G. Desaulniers, and L. M. Rousseau, "The traveling salesman problem with time windows in postal services," *Journal of the Operational Research Society*, vol. 72, no. 2, pp. 383–397, 2021.

[69] X. Lu and Y. Zhou, "A novel global convergence algorithm: bee collecting pollen algorithm," in *International Conference on Intelligent Computing*Berlin, Germany, 2008.

[70] M. Koudil, K. Benatchba, A. Tarabet, and E. B. Sahraoui, "Using artificial bees to solve partitioning and scheduling problems in codesign," *Applied Mathematics and Computation*, vol. 186, no. 2, pp. 1710–1722, 2007.

[71] H. A. Abbass, "An agent based approach to 3-SAT using marriage in honey-bees optimization," *Int. J. Knowledge-based Intell. Eng. Syst.*, vol. 6, no. 2, pp. 64–71, 2002.

[72] J. Teo and H. A. Abbass, "An Annealing Approach to the Mating-Flight Trajectories in the Marriage in Honey Bees Optimization Algorithm," Technical Report CA04/01, School of Computer Science, University of New South Wales at ADFA, Canberra, Australia, 2001.

[73] A. Phu-ang and A. Thammano, "Memetic algorithm based on marriage in honey bees optimization for flexible job shop scheduling problem," *Memetic Computing*, vol. 9, no. 4, pp. 295–309, 2017.

[74] X. Wen, X. Li, L. Gao, K. Wang, and H. Li, "Modified honey bees mating optimization algorithm for multi-objective uncertain integrated process planning and scheduling problem," *International Journal of Advanced Robotic Systems*, vol. 17, no. 3, Article ID 172988142092523, 2020.

[75] P. Palominos, F. Toledo, A. Véjar, and M. Alfaro, "Marriage in honey bees optimization algorithm for flow-shop problems," *Informatica Economica*, vol. 16, no. 2, pp. 27–34, 2012.

[76] R. Solgi and H. A. Loáiciga, "Bee-inspired metaheuristics for global optimization: a performance comparison," *Artificial Intelligence Review*, vol. 54, no. 7, pp. 4967–4996, 2021.

[77] M. T. Vakil-Baghmisheh and M. Salim, "A modified fast marriage in honey bee optimization algorithm," in *Proceedings of the International Symposium on Telecommunications*, pp. 950–955, Tehran, Iran, December 2010.

[78] S. K. Prabhakar and S. W. Lee, "Transformation based tri-level feature selection approach using wavelets and swarm computing for prostate cancer classification," *IEEE Access*, vol. 8, Article ID 127462, 2020.

[79] Y. Çelik and E. Ülker, "A marriage in honey bee optimisation approach to the asymmetric travelling salesman problem," *Int. J. Innov. Comput. Inf. Control*, vol. 8, no. 6, pp. 4123–4132, 2012.

[80] J. Nayak and B. Naik, "A novel honey-bees mating optimization approach with higher order neural network for classification," *Journal of Classification*, vol. 35, no. 3, pp. 511–548, 2018.

[81] Y. Yin, W. H. Wu, T. C. E. Cheng, C. C. Wu, and W. H. Wu, "A honey-bees optimization algorithm for a two-agent single-machine scheduling problem with ready times," *Applied Mathematical Modelling*, vol. 39, no. 9, pp. 2587–2601, 2015.

[82] K. Zanbouri and N. Jafari Navimipour, "A cloud service composition method using a trust-based clustering algorithm and honeybee mating optimization algorithm," *International Journal of Communication Systems*, vol. 33, no. 5, Article ID e4259, 2020.

[83] B. Arun and T. V. Kumar, "Materialized view selection using marriage in honey bees optimization," *International Journal of Natural Computing Research*, vol. 5, no. 3, pp. 1–25, 2015.

[84] M. Mazo, "Programación de tareas en una máquina: un enfoque mediante la metaheurística MBO," Master thesis, University of Santiago de Chile, Santiago, RM, Chile, 2009.

[85] G. Gutin, A. Yeo, and A. Zverovich, "Traveling salesman should not be greedy: domination analysis of greedy-type heuristics for the TSP," *Discrete Applied Mathematics*, vol. 117, no. 1–3, pp. 81–86, 2002.

[86] C. Nilsson, *Heuristics for the Traveling Salesman Problem*, Linköping, Sweden, 2003.

[87] Y. Li, D. Yao, W. Chen, J. Zheng, and J. Yao, "Adaptive particle swarm optimizer for beam angle selection in radiotherapy planning," in *Proceedings of the International Conference on Mechatronics and Automation*, pp. 421–425, Niagara Falls, ON, Canada, July 2005.

[88] H. A. Abbass, "A monogenous MBO approach to satisfiability," in *Proceedings of the International Conference on Computational Inteligence for Modelling Control and Automation*, Vienna, Austria, 2001.

[89] S. Kirkpatrick, "Optimization by simulated annealing: quantitative studies," *Journal of Statistical Physics*, vol. 34, no. 5–6, pp. 975–986, 1984.

[90] M. Hasegawa, T. Ikeguchi, and K. Aihara, "Solving large scale traveling salesman problems by chaotic neurodynamics," *Neural Networks*, vol. 15, no. 2, pp. 271–283, 2002.

[91] M. B. Teitz and P. Bart, "Heuristic methods for estimating the generalized vertex median of a weighted graph," *Operations Research*, vol. 16, no. 5, pp. 955–961, 1968.

[92] D. S. Johnson, "The traveling salesman problem: a case study in local optimization," in *Local Search in Combinatorial Optimization*, E. H. L. Aarts and J. K. Lenstra, Eds., Wiley, London, UK, 1997.

[93] R. Carrasco, M. Vargas, I. Soto, G. Fuertes, and M. Alfaro, "Copper metal price using chaotic time series forecasting," *IEEE Latin America Transactions*, vol. 13, no. 6, pp. 1961–1965, 2015.

[94] S. Das and P. Saha, "Performance of swarm intelligence based chaotic meta-heuristic algorithms in civil structural health monitoring," *Measurement*, vol. 169, Article ID 108533, 2021.

[95] J. Mingjun and T. Huanwen, "Application of chaos in simulated annealing," *Chaos, Solitons & Fractals*, vol. 21, no. 4, pp. 933–941, 2004.

[96] S. Aono, Y. Nishio, and M. Hasegawa, "Effect of chaos noise for tabu search neural network with 2-opt for TSPs," in *Proceedings of the International Workshop on Nonlinear Circuit and Signal Processing*, pp. 141–143, Rome, Italy, 2004.

[97] X. Q. Zuo and Y. S. Fan, "Solving the job shop scheduling problem by an immune algorithm," in *Proceedings of the International Conference on Machine Learning and Cybernetics*, pp. 3282–3287, Toyama, Japan, 2005.

[98] G. Reinelt, "TSPLIB," 2021, http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/.

[99] P. Palominos, V. Parada, M. Alfaro, C. Ortega, and G. Gatica, "Chaotic methods applied to the marriage in honey bees optimization metaheuristic," in *Proceedings of the International Conference on Production Research: Innovation in Product and Production*, Chicago, Illinois, 2011.