Journal of Computational and Applied Mathematics 477 (2026) 117145

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

# A high-performance computing-based multi-island distributed migrating birds optimization algorithm implemented using MPI

Abdullah Tülek [a], Gültekin Kuvat [b,*]

[a] 100. Yil Vocational and Technical Anatolian High School, Türkiye
[b] Department of Computer Engineering, Faculty of Engineering, Balıkesir University, Türkiye

ARTICLE INFO

*Keywords:*
Multi-island distributed migrating birds optimization algorithm
HPC
Migration parameters
Regression analysis
ANN

ABSTRACT

Metaheuristic algorithms are commonly used optimization techniques for solving high-dimensional complex problems. To achieve more successful results, improvements are made on these metaheuristic algorithms. This study aims to enhance the performance of the Migrating Birds Optimization (MBO) Algorithm, a neighborhood-based metaheuristic algorithm. To this end, MBO is implemented on a high-performance computing (HPC) architecture using Open MPI, and the Multi-Island Distributed Migrating Birds Optimization (MIDMBO) Algorithm is developed. MIDMBO is a distributed metaheuristic optimization algorithm that utilizes multiple islands, where successful solutions migrate between islands. The migration process between islands, facilitated by MPI, enables MIDMBO to generate more successful outcomes. Therefore, studies focused on the migration process are of great importance. In this study, the relationship between algorithm performance and variations in fundamental migration parameters such as migration rate, migration interval, and the number of islands is examined. Regression analyses are performed on the results obtained from MIDMBO, and the model that best explains performance variation is identified. Additionally, artificial neural network (ANN) models are constructed, and the degree to which changes in parameter values explain the results is evaluated.

## 1. Introduction

Metaheuristic algorithms are methods used for solving complex problems that involve a large number of variables. Gravitational Search [1], Big Bang–Big Crunch [2], Differential Evolution [3], Particle Swarm Optimization [4,5], Whale Optimization [6,7], Grey Wolf [8], Artificial Bee Colony [9,10], Charged System Search [11], Migrating Birds [12], and Genetic Algorithms [13] are among the commonly used metaheuristic optimization algorithms. However, the problems being addressed are becoming more difficult, and the problem sizes are increasing. Therefore, there is ongoing research on island-based distributed metaheuristic algorithms, which can perform a more detailed search in the solution space to obtain better results. The island model is a method in which the selected metaheuristic algorithm is applied concurrently on subpopulations that are created independently of each other [14]. In island-based distributed metaheuristic algorithms, multiple subpopulations are used to simultaneously explore different regions of the solution space. Good solutions are exchanged between subpopulations at certain intervals. This process reduces the risk of getting stuck in local optima for solutions newly introduced to a subpopulation. Additionally, it facilitates better searching and leads to successful outcomes by increasing diversity [15]. The process of exchanging good solutions between subpopulations is called migration. Migration is a

* Corresponding author.
*E-mail addresses:* abdullahtulek@balikesir.edu.tr (A. Tülek), gkuvat@balikesir.edu.tr (G. Kuvat).

https://doi.org/10.1016/j.cam.2025.117145
Received 29 April 2025; Received in revised form 30 August 2025;
Available online 10 October 2025

process determined by migration rate (MR), migration interval (MI), number of islands (NIs), migration policy, communication model, and topology [16].

The migration process and the correct selection of migration parameters are the key reasons for the success of island-based distributed metaheuristic algorithms. Therefore, studies on the migration process are important. In one study [17], two different algorithms were proposed in which migration parameters for Ant Colony Optimization are dynamically determined. In the first algorithm, a fixed MI value and tolerance value were set. If the best solution is improved, the tolerance value is increased. The algorithm is run with a reduced MI value when the condition (tolerance value $\geq$ total iterations / MI value) is met. In the second algorithm, both MI and MR are dynamically determined based on the quality of the migrating solutions. According to experimental studies, the MI values of 25 and 50 with a decrease rate of 1 and 5 were used to solve the Traveling Salesman Problem with a system consisting of 4 subpopulations. The results showed that better outcomes were obtained with MI 50 and a decrease rate of 5. For the second proposed algorithm, results were obtained with MI values of 25 and 50, and MR values of 15, 30, and 50. When comparing the two algorithms, it was observed that the model where both MI and MR values are determined produced slightly better results. In this study, the relationship between time and results for different numbers of ants was demonstrated. Additionally, speedup was examined according to the number of nodes. The results obtained throughout the iterations for the proposed method and the serial model are provided.

In another study [18], an island-based Harmony Search algorithm was proposed. In this study, nine different scenarios were created with island numbers of 2, 5, and 10, MI of 50, 100, and 500, and MR of 10, 20, and 30. The results obtained for these scenarios were presented. A random ring topology was used in this study. The experiments were not conducted in a real distributed system. They were modeled and carried out on a single computer. In [19], the MR value in island-based genetic algorithms is calculated using fuzzy logic. The MR value was computed because of fuzzy rules formed using fitness values. Experiments were conducted for different subpopulation sizes and numbers, and it was shown that the proposed method for determining the MR produced more successful results. In another study, both the master-slave model and the island model were applied in parallel genetic algorithms. Synchronous and asynchronous communication were used in the island model. The results obtained for different numbers of cores were presented. It was shown that as the number of cores increased, the time spent decreased. The speedup results were analyzed based on the obtained times, and in some cases, super linear speedup was achieved [20].

In a study where parallel genetic algorithms (GA) were applied [21], a dynamic topology was proposed for island-based genetic algorithms, and the effects of different MR, MI, and subpopulation sizes on the search speed were demonstrated. In another study [22], two different methods, namely the Hybrid Heuristic and Genetic-based Task Scheduling Algorithm for Heterogeneous Computing (HHG) and the Hybrid Task Duplication and Genetic-based Task Scheduling Algorithm for Heterogeneous Computing (HTDG), were developed. These new methods were compared with NGA, EGA-TS, HEFT, and PEFT. According to the experimental results, HTDG produced 89 % better outcomes, while HHG achieved 56 % improvement. In the HHG and HTDG methods, most chromosomes in the initial population were generated randomly. In addition, guided chromosomes were incorporated into the population to enhance the algorithms. These guided chromosomes accelerated the convergence rate of the GA. In HHG and HTDG, the fitness value is computed in different ways. In these methods, the best 20 % of solutions in the population are directly transferred to the new population, while the remaining 80 % are generated through crossover and mutation. During the crossover phase, tournament selection was employed, with a crossover probability of 0.6 and a mutation probability of 0.2. In the Node-Based Performance comparison, HTDG demonstrated superior performance compared to other algorithms. HHG and NGA produced similar results, whereas EGA-TS performed less successfully. In comparisons based on the Communication to Computation Ratio (CCR), HTDG again outperformed the other algorithms. HHG was also more effective than NGA across all comparison types. In the Processors-Based Performance comparison, results were reported for 4, 8, and 16 processors, showing that HTDG achieved better performance than the other algorithms. In the Running Time-Based Performance Analysis, EGA-TS exhibited shorter execution time compared to the other algorithms. HHG's runtime was shorter than HTDG's, while HTDG, thanks to elitism, consumed less execution time than NGA. The results of these comparisons indicate that HTDG and HHG can be regarded as promising and effective approaches.

In another study [23], a parallel Artificial Bee Colony (ABC) algorithm with a dynamic migration strategy was proposed for solving the workshop scheduling problem. According to this strategy, if no improvement is made within the colony for a given MI value, a migration request is sent to neighboring colonies, and migration occurs. In a study using ABC algorithm, results for different island numbers, population sizes, problem sizes, and MI were presented. The Wilcoxon test was applied to the results, showing that the changes observed were statistically significant. Additionally, speedup values for different problem sizes were presented graphically [24]. In a different study [25], the sensor deployment problem was solved using the serial ABC algorithm (sABC), the parallel ABC algorithm with the conventional emigrant creation strategy (pABC), and the parallel ABC algorithm with the cooperative emigrant creation strategy (coop-pABC). The neighborhood topology of the pABC and coop-pABC algorithms is structured as a ring. For each sub-colony, only one emigrant was sent to a neighboring sub-colony and replaced with the worst solution in that colony. In the pABC algorithm, the best food source is migrated. In the coop-pABC algorithm, however, the best food source is reinforced with the parameters of a randomly selected food source. If the $i$-th parameter of the randomly selected food source improves the fitness value of the best food source, then the $i$-th parameter of the best food source is replaced with the corresponding parameter of the randomly selected food source. The coop-pABC algorithm produced better results than the pABC algorithm in all three experimental cases. When compared with the sABC algorithm, the coop-pABC algorithm achieved better performance in two out of three experimental cases. On the other hand, it was shown that the pABC algorithm yielded higher speedup and efficiency values compared to the coop-pABC algorithm. In a study using an island-based parallel memetic algorithm, a two-tier heterogeneous vehicle routing problem was solved [26].

In the study conducted in [27], a multi-objective problem, the flight assignment problem, was solved using a parallel evolutionary algorithm with the island model. In this study, the MI value was determined in three different ways, and the results were compared.

The first method used to determine the MI value was called dynamic MI. According to this method, after each iteration, it was decided whether migration would occur in the next step based on a specified difference value. In the second method, a fixed MI value of 5 was used. In the third method, a randomly generated value in the range of 0-1 was used, and migration occurred if the value was greater than 0.2; otherwise, migration did not occur. According to the results, it is stated that dynamically determining the MI produces better convergence and diversity values.

In [28], the workflow scheduling problem in cloud environments was solved using a distributed Grey Wolf algorithm. In this study, a random ring topology was used. In [29], four different migration topologies were proposed for the island-based Crow Search Algorithm. The study was conducted using 16 islands, and the effect of the proposed topologies on the problems was demonstrated. In a study using the island-based Differential Evolution (DE) algorithm, results from running the differential evolution algorithm with a single subpopulation were compared with those from synchronous and asynchronous parallel DE algorithms [30]. In another study [31], recent works on parallel GA were reviewed and presented. The journals, conferences, and countries where the most articles were published, along with the number of authors per country, were provided. Additionally, the parallel island, cellular, and hybrid models were described. The architecture and software used were outlined, and the studies conducted were indicated.

## 2. Multi-island distributed migrating birds optimization (MIDMBO)

Migrating Birds Optimization (MBO) algorithm is one of the metaheuristic algorithms that use a neighborhood search method. The algorithm models the Λ-shaped flight formation used by migrating birds to gain energy, which allows them to fly long distances [12]. Each bird benefits from the air current created by the bird before it, reducing its energy consumption and enabling it to fly longer distances. In a study conducted on a flock of 25 birds, it was observed that birds flying in a Λ-shape increased their range by up to 70 % [32]. Using this information, a random initial solution is generated for each bird in the algorithm.

One of the birds in the flock is considered the leader bird. It is assumed that half of the other birds are on the right side, and the other half are on the left side, forming a virtual Λ-shape. The leader bird generates a predetermined number of neighboring solutions. The best solution between the initial solution and the newly generated solutions is kept by the leader bird. From the remaining solutions, half of the best solutions (x solutions) are sent to the bird at the right rear, and the other half (x solutions) are sent to the bird at the left rear. The bird receiving new solutions from the leader compares them with the neighborhood solutions it has generated and keeps the best solution. The other x best solutions are then sent to the bird at the rear. This process continues along the right and left wings until the last bird. These operations constitute an iteration step in the algorithm. After a predetermined number of iterations, it is assumed that the leader bird is exhausted, and a leader bird change is performed. The leader bird is sent to the rightmost rear of the flock, which is assumed to be positioned in a Λ-shape. The birds on the right wing move forward one by one, and a new leader bird is chosen. When the next bird change interval arrives, the same process is performed on the left wing of the flock, which is assumed to be positioned in a Λ-shape. These steps are repeated until a predefined stopping criterion is met [12,33,34].

The MBO algorithm is used by many researchers in various studies. In [35], it was shown that MBO produces successful results for quadratic assignment problems but does not perform as well in the Traveling Salesman Problem (TSP). To obtain better results, seven different neighborhood methods were applied to TSP, and it was demonstrated that the algorithm's performance is affected by the chosen neighborhood method. In another study [36], two different crossover methods were used in the MBO algorithm to generate neighboring solutions. The results were analyzed using the Taguchi experimental design method, and an attempt was made to determine the ideal parameters. In [37], the MBO algorithm was run with a special neighborhood generation method to solve a multi-objective task assignment problem. According to the results, it was stated that the performance of the MBO algorithm improved. The MBO, when run with the special neighborhood function, was compared with the GA, Honey Bee Mating Optimization, and the known MBO algorithm, and it was shown that the developed method produced successful results. In [38], the MBO algorithm was compared with ABC, PSO, DE, and GA using 10 different test functions and 5 different dimensions, and the results were analyzed. In the study, the parameter values for five different systems were determined using the algorithms mentioned above, and the produced Mean Squared Error (MSE) values were compared. Based on the analysis, it was shown that the MBO algorithm produced successful results. In [39], the ABC algorithm was run together with the MBO algorithm. A new method was proposed by utilizing the success of the ABC in the search phase and the success of the MBO algorithm in the improvement phase. According to the proposed method, the ABC is first run to avoid local optima. Then, the MBO algorithm is executed to generate better solutions using the existing solutions.

There exist numerous studies in which the MBO algorithm has been utilized and further developed. However, the desire to achieve better results has led to the idea of applying the MBO algorithm within distributed systems using multiple subpopulations (islands). In this study, the Multi-Island Distributed Migrating Birds Optimization (MIDMBO) algorithm was implemented using a distributed island model and a high performance computing (HPC) architecture. The MBO algorithm was executed within each subpopulation for a predefined number of iterations (MI), after which selected solutions (in the amount of MR) were migrated from one subpopulation to another based on a ring topology. This process was carried out using the Message Passing Interface (MPI), a widely adopted parallel programming model for data communication in high performance computing systems [40–42]. MPI functions were employed to enable the transfer of high-quality solutions to neighboring islands according to the predefined ring topology. After executing MI iterations, each island identified its best solutions according to the MR and sent them to its neighboring island using the "MPI_Send" function. The neighboring island received these solutions through the "MPI_Recv" function and incorporated them into its own population. During this incorporation phase, in order to keep the population size constant and eliminate the negative impact of poor solutions on the search behavior, the incoming solutions replaced the least fit individuals in the population. Additionally, in each iteration step, the "MPI_Reduce" function was used to determine the best global solution across all islands. This function gathered the locally best solutions from each island to the master node, where the best among them was selected as the global best solution.

**Algorithm 1**
MPI Code Segments.

```
int main(int argc, char *argv[])
{
    /* Start up MPI */
    MPI_Init(&argc, &argv);
    /* Find out number of processes */
    MPI_Comm_size(MPI_COMM_WORLD, &pSize);
    /* Find out processes rank */
    MPI_Comm_rank(MPI_COMM_WORLD, &myRank);
.....

/* During the optimization process, the best MR solutions are migrated to the neighboring island at predefined MI iteration intervals, following the ring topology
    structure */
if (myRank == i)
{
    dest = (myRank + 1) % pSize;
    MPI_Send(&value[0], COL*MIG_RATE, MPI_DOUBLE, dest, iter, MPI_COMM_WORLD);
}
if (myRank == (i + 1) % pSize)
{
    source = (myRank - 1) % pSize;
    if (source<0) source = source + pSize;
      MPI_Recv(&incoming[0], COL*MIG_RATE, MPI_DOUBLE, source, iter, MPI_COMM_WORLD, &status);
}
.....

/* At each iteration, the most successful solution among those on all islands is selected and transferred to the master node */
MPI_Reduce(&MyMinFitness, &GlobalMinFitness, 1, MPI_DOUBLE, MPI_MIN, 0, MPI_COMM_WORLD);
.....
```

Algorithm 1 below presents example code [43] segments demonstrating the stages in which the MPI function is utilized in this study.

The ring topology used in this study is one of the most commonly adopted migration topologies in the literature. Its widespread use can be attributed to several factors: ease of implementation through coding, suitability for cluster computing environments, low communication overhead, and compatibility with various hardware architectures. In this topology, high-quality solutions within an island population are migrated to neighboring subpopulations in accordance with the ring structure. This enables the effective sharing of good solutions and contributes to an overall improvement in performance. Moreover, it helps reduce the risk of premature convergence to local optima, which is one of the major challenges in metaheuristic algorithms [15].

In multi-island metaheuristic optimization algorithms, the structure of the migration process is determined according to the factors provided below [16].

- Topology: Determines the subpopulation to which the solutions will migrate.
- MR: Determines the number of solutions to be migrated. It is expressed as a percentage based on the subpopulation size.
- MI: The iteration step or the predefined criterion between two migration operations.
- Migration Policy:
  - The selection method for the migrating solutions:
    - Select from the best solutions.
    - Select randomly.
  - The method of replacement after migration:
    - Replace with the worst solutions.
    - Replace randomly.
- Communication Model: Synchronous / Asynchronous.

In this study:

- Topology: A ring topology is used.
- MR: Set as 2 %, 6 %, 10 %, 20 %, and 40 %.
- MI: Set as 5, 10, 25, 50, and 100 iterations.
- Migration Policy:
  - Selection method for the migrating solutions:
    - Select from the best solutions.
  - Replacement method after migration:
    - Replace with the worst solutions.
- Communication Model: A synchronous model is used.

**Algorithm 2**
MIDMBO Pseudocode.

```
Generate n random initial solutions (birds) and place the solutions in the virtual V formation.
Define the parameters:
  M = leader bird change interval
  K = maximum number of iterations
  MI = migration interval
  MR = migration rate
for (i = 0; i < K; i++) do
  for (s = 0; s < MI; s++) do
    for (j = 0; j < M; j++) do
      for each bird in the swarm do
        if bird is leader then
          Try to improve the solution by generating k neighboring solutions,
          Send the best x solutions to the bird on the left back and right back.
        Else
          Use (k - x) generated neighboring solutions and x solutions come from front to improve
          the bird's position, then send x best unused solutions to the back bird.
        end if
      end for
    end for
    Change the leader solution (bird).
  end for
  Perform migration for MR percentage of birds.
end for
Return the best solution.
```

The MIDMBO Algorithm Pseudocode is provided in Algorithm 2 below [34].

There are various studies in literature where the island-based MBO algorithm has been applied. In one study [44], a PSO-based Improved Multi-Flocks Migrating Birds Optimization (IMFMBO) algorithm was developed, where multiple flocks were used instead of a single flock. In this study, the interaction among flocks was facilitated through the leader bird in each flock. The interaction model proposed in the method resembles the philosophy of the PSO algorithm. The new position of the leader bird in each flock is determined based on the previous leader's position and velocity, just like in PSO. Since all birds in a flock are potential leader candidates, their velocity and position information are maintained. When determining the leader bird's position, the best of the local bests among the flocks is considered the global best. This ensures that all flocks are guided toward more promising regions of the search space through inter-flock interaction. The interaction among flock leaders takes place during the leader change step. This way, the basic structure of the MBO algorithm is preserved while allowing each flock to continue searching independently until its leader is changed. To demonstrate the effectiveness of the proposed method, IMFMBO with 15 flocks and the standard MBO algorithm were applied to solve Traveling Salesman Problems (TSPs), and the results were compared. According to the results, IMFMBO produced better results across all problems. Furthermore, the IMFMBO results were compared with the results of various algorithms from different studies in literature, and it was shown to produce favorable outcomes in many cases.

In another comparative study [45], Simulated Annealing (SA), GA, MBO, and Clonal Selection Algorithm (CSA) were used to solve the Multidimensional Two-Way Number Partitioning Problem (MDTWNPP), and the results were presented comparatively. When applying the MBO algorithm, multiple flocks were used, and this approach was termed Multiple Flock Migrating Birds Optimization (MFMBO). This allowed a more effective exploration of the search space. In the MFMBO implementation, the number of flocks was set to 500. When analyzing results across different dataset sizes, it was noted that GA and SA outperformed both MFMBO and CSA. Comparing GA and SA revealed that they were consistently competing head-to-head across all datasets. Based on this, GA and SA were identified as the most powerful and successful metaheuristic methods for solving MDTWNPP. MFMBO was ranked as the third most successful method.

Many of the island-based MBO studies in the literature are implemented by creating multiple island structures on a single machine. In contrast to these studies, this work employs a real HPC architecture to establish the islands, enabling the migration of successful solutions between islands using MPI, and thus performs the development on a distributed system. Furthermore, the effects of migration parameters on the performance of the MIDMBO algorithm are analyzed using quadratic regression models. It is demonstrated that variations in the values of MR, MI and NIs have a high explanatory power on algorithm performance. Additionally, artificial neural network (ANN) model was constructed to model the relationships, where MR, MI, and NIs values were used as inputs and the fitness result as the output. The ANN results revealed a strong correlation between the output and the target values. The coefficients of determination ($R^2$) obtained from the ANN analysis were found to be higher than those obtained from the regression model, and this

**Table 1**
Benchmark functions.

| Name | Equation | Feature |
|---|---|---|
| Rosenbrock | $f_{Rosenbrock}(x) = \sum_{i=1}^{d-1}\left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right]$<br>$x_i \in [-2.048,\ 2.048]$<br>Global minimum: $x^* = (1,1...,1), f(x^*) = 0$ | Unimodal |
| Rotated Hyper-Ellipsoid | $f_{Rotated}(x) = \sum_{i=1}^{d}\sum_{j=1}^{i} x_j^2$<br>$x_i \in [-65.536,\ 65.536]$<br>Global minimum: $x^* = (0,0...,0), f(x^*) = 0$ | Unimodal |
| Zakharov | $f_{Zakharov}(x) = \sum_{i=1}^{d} x_i^2 + \left(\sum_{i=1}^{d} 0.5ix_i\right)^2 + \left(\sum_{i=1}^{d} 0.5ix_i\right)^4$<br>$x_i \in [-5,\ 10]$<br>Global minimum: $x^* = (0,0...,0), f(x^*) = 0$ | Unimodal |
| Rastrigin | $f_{Rastrigin}(x) = 10d + \sum_{i=1}^{d}[x_i^2 - 10\cos(2\pi x_i)]$<br>$x_i \in [-5.12,\ 5.12]$<br>Global minimum: $x^* = (0,0...,0), f(x^*) = 0$ | Multimodal |
| Schwefel | $f_{Schwefel}(x) = 418.9829d - \sum_{i=1}^{d} x_i \sin\left(\sqrt{\lvert x_i\rvert}\right)$<br>$x_i \in [-500,\ 500]$<br>Global minimum: $x^* = (420.9687,...,420.9687), f(x^*) = 0$ | Multimodal |

**Table 2**
Optimization results for the Rosenbrock benchmark function.

| Rosenbrock | | Number of Island (NIs) | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 8 | | | 16 | | | 24 | | | 32 | | |
| Migration Rate (MR) | Migration Interval (MI) | Average | Best | Worst | Average | Best | Worst | Average | Best | Worst | Average | Best | Worst |
| No Migration (MBO) | | 42.8003 | 39.9944 | 43.6839 | 41.7795 | 37.1348 | 43.5224 | 41.2192 | 36.1843 | 43.5378 | 42.2141 | 40.0936 | 43.4766 |
| 2 | 5 | 42.2588 | 39.5672 | 94.6499 | 39.6733 | 38.7665 | 40.2659 | 39.3234 | 36.3136 | 40.0512 | 39.0897 | 36.7913 | 40.0107 |
| | 10 | 41.1747 | 40.4636 | 41.7137 | 40.6881 | 37.2553 | 41.453 | 40.8655 | 40.0973 | 41.2797 | 40.7338 | 40.1424 | 41.3918 |
| | 25 | 41.9561 | 41.3854 | 42.4258 | 41.8149 | 39.8422 | 42.3731 | 41.7857 | 37.8961 | 42.1396 | 41.6045 | 36.9264 | 42.0827 |
| | 50 | 42.4508 | 41.9047 | 42.7406 | 42.2943 | 40.3981 | 42.6241 | 42.101 | 38.4793 | 42.5271 | 42.0254 | 41.1277 | 42.5006 |
| | 100 | 42.3187 | 34.9392 | 42.97 | 42.4133 | 40.8004 | 42.867 | 42.0634 | 39.1925 | 42.7427 | 41.8991 | 38.2051 | 42.8111 |
| 6 | 5 | 39.6524 | 32.484 | 44.5874 | 38.9701 | 38.3971 | 39.8961 | 38.4481 | 34.8082 | 39.3717 | 38.2041 | 35.1375 | 39.1488 |
| | 10 | 40.8588 | 38.622 | 41.4322 | 40.4712 | 39.8509 | 40.929 | 40.3118 | 38.0623 | 40.9157 | 40.3477 | 39.9989 | 40.5997 |
| | 25 | 41.7559 | 40.9672 | 42.2681 | 41.6735 | 41.2049 | 42.0452 | 41.601 | 41.2939 | 42.0461 | 41.4932 | 40.5403 | 41.7935 |
| | 50 | 42.2055 | 41.8493 | 42.7463 | 42.0662 | 40.8021 | 42.4324 | 41.7661 | 37.8805 | 42.3481 | 42.0251 | 41.4201 | 42.231 |
| | 100 | 42.5158 | 42.0546 | 42.8275 | 42.1977 | 40.7196 | 42.6972 | 41.8093 | 39.7661 | 42.6779 | 42.1724 | 40.8098 | 42.508 |
| 10 | 5 | 39.5098 | 37.8695 | 41.4098 | 38.9564 | 37.6967 | 43.3842 | 38.4342 | 34.315 | 39.4961 | 38.1868 | 35.3181 | 39.0951 |
| | 10 | 40.6535 | 39.5142 | 41.3234 | 40.3027 | 39.3861 | 40.9126 | 40.1821 | 39.1598 | 40.6441 | 40.0364 | 38.2201 | 40.4822 |
| | 25 | 41.791 | 41.2859 | 42.2856 | 41.5546 | 40.7439 | 41.892 | 41.4273 | 40.9397 | 41.7723 | 41.4837 | 41.1623 | 41.8778 |
| | 50 | 42.1601 | 41.755 | 42.5234 | 42.0475 | 41.2442 | 42.4513 | 40.8995 | 39.62 | 42.4009 | 41.6011 | 38.6662 | 42.2299 |
| | 100 | 42.4176 | 41.1767 | 42.7913 | 42.1719 | 40.4356 | 42.6852 | 42.1031 | 34.8877 | 42.5702 | 41.8039 | 39.6212 | 42.5255 |
| 20 | 5 | 40.032 | 38.0905 | 45.6403 | 38.7953 | 36.0749 | 40.4274 | 38.4238 | 36.7053 | 39.2318 | 38.2634 | 36.7329 | 39.0775 |
| | 10 | 40.9245 | 38.9683 | 45.1013 | 40.2843 | 39.0987 | 40.9706 | 40.1138 | 38.8377 | 40.6016 | 39.899 | 38.7701 | 40.5663 |
| | 25 | 41.7408 | 41.2815 | 42.1898 | 41.4237 | 40.7194 | 41.9414 | 41.3863 | 40.751 | 41.6927 | 41.3686 | 40.7957 | 41.6842 |
| | 50 | 42.1333 | 41.5526 | 42.5914 | 41.9256 | 39.6513 | 42.4306 | 41.9511 | 41.5343 | 42.2339 | 41.8375 | 40.2857 | 42.2636 |
| | 100 | 42.587 | 42.2115 | 43.1323 | 42.4074 | 41.1029 | 42.9327 | 42.2162 | 39.2286 | 42.6996 | 42.1519 | 39.4544 | 42.6474 |
| 40 | 5 | 39.5278 | 31.9134 | 43.5778 | 39.1595 | 37.6661 | 43.9231 | 38.6324 | 35.5834 | 39.4096 | 38.4346 | 36.9022 | 39.2844 |
| | 10 | 40.8208 | 40.0613 | 42.9327 | 40.2178 | 36.0049 | 40.7672 | 40.2754 | 39.7024 | 40.6555 | 39.9858 | 36.8547 | 40.6033 |
| | 25 | 41.8291 | 41.6092 | 42.2596 | 41.4839 | 40.1517 | 42.0642 | 41.5473 | 41.1723 | 41.9406 | 41.4066 | 40.6107 | 41.8272 |
| | 50 | 42.3579 | 42.061 | 42.6125 | 42.1552 | 41.8694 | 42.6154 | 41.8912 | 40.73 | 42.4705 | 41.9804 | 39.5831 | 42.5461 |
| | 100 | 42.7639 | 42.2437 | 43.1522 | 42.4796 | 41.0327 | 43.0053 | 42.5049 | 41.4306 | 42.8819 | 42.2825 | 40.7511 | 42.7835 |
| MIDMBO Success Rate (%) | | 100 | 36 | 84 | 56 | 8 | 96 | 44 | 16 | 100 | 96 | 60 | 100 |

**Table 3**
Optimization results for the Rotated Hyper-Ellipsoid benchmark function.

| Rotated Hyper-Ellipsoid | | Number of Island (NIs) | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 8 | | | 16 | | | 24 | | | 32 | | |
| Migration Rate (MR) | Migration Interval (MI) | Average | Best | Worst | Average | Best | Worst | Average | Best | Worst | Average | Best | Worst |
| No Migration (MBO) | | 1.61E-16 | 3.16E-17 | 5.42E-16 | 1.01E-16 | 2.05E-17 | 2.51E-16 | 9.53E-17 | 3.71E-17 | 2.4E-16 | 7.30E-17 | 2.09E-17 | 1.37E-16 |
| 2 | 5 | 2.47E-22 | 6.81E-23 | 6.39E-22 | 1.61E-22 | 5.31E-23 | 3.55E-22 | 1.57E-22 | 5.03E-23 | 4.05E-22 | 1.55E-22 | 6.25E-23 | 2.81E-22 |
| | 10 | 2.31E-19 | 7.28E-20 | 5.02E-19 | 2.14E-19 | 1.04E-19 | 4.99E-19 | 1.78E-19 | 6.04E-20 | 3.79E-19 | 1.62E-19 | 9.34E-20 | 2.46E-19 |
| | 25 | 1.31E-17 | 4.91E-18 | 2.42E-17 | 1.03E-17 | 3.82E-18 | 2.16E-17 | 8.20E-18 | 3.63E-18 | 1.28E-17 | 9.13E-18 | 5.25E-18 | 1.79E-17 |
| | 50 | 4.75E-17 | 1.50E-17 | 8.78E-17 | 3.50E-17 | 1.28E-17 | 5.35E-17 | 2.83E-17 | 9.88E-18 | 5.26E-17 | 2.48E-17 | 9.53E-18 | 4.76E-17 |
| | 100 | 8.66E-17 | 1.41E-17 | 1.72E-16 | 7.08E-17 | 2.17E-17 | 1.39E-16 | 4.84E-17 | 1.24E-17 | 9.43E-17 | 5.04E-17 | 2.81E-17 | 7.94E-17 |
| 6 | 5 | 2.33E-26 | 3.77E-27 | 5.66E-26 | 1.55E-26 | 2.93E-27 | 3.48E-26 | 1.25E-26 | 4.76E-27 | 3.36E-26 | 9.43E-27 | 2.80E-27 | 1.67E-26 |
| | 10 | 2.33E-21 | 8.55E-22 | 5.15E-21 | 1.84E-21 | 9.48E-22 | 3.79E-21 | 1.72E-21 | 5.76E-22 | 4.53E-21 | 1.41E-21 | 6.64E-22 | 2.98E-21 |
| | 25 | 2.44E-18 | 7.78E-19 | 4.71E-18 | 1.66E-18 | 5.77E-19 | 3.35E-18 | 1.50E-18 | 8.23E-19 | 2.60E-18 | 1.37E-18 | 5.26E-19 | 2.27E-18 |
| | 50 | 2.24E-17 | 9.03E-18 | 3.95E-17 | 1.49E-17 | 6.81E-18 | 2.66E-17 | 1.55E-17 | 7.15E-18 | 3.20E-17 | 1.22E-17 | 4.86E-18 | 2.26E-17 |
| | 100 | 5.96E-17 | 1.47E-17 | 1.54E-16 | 4.53E-17 | 1.64E-17 | 9.45E-17 | 3.75E-17 | 1.50E-17 | 6.57E-17 | 3.38E-17 | 1.05E-17 | 5.07E-17 |
| 10 | 5 | 1.67E-28 | 3.13E-29 | 4.20E-28 | 8.38E-29 | 2.20E-29 | 2.64E-28 | 6.72E-29 | 2.03E-29 | 1.66E-28 | 6.08E-29 | 1.81E-29 | 1.70E-28 |
| | 10 | 1.78E-22 | 5.71E-23 | 3.65E-22 | 1.09E-22 | 4.74E-23 | 1.88E-22 | 1.00E-22 | 4.52E-23 | 2.29E-22 | 9.44E-23 | 4.89E-23 | 1.56E-22 |
| | 25 | 8.67E-19 | 3.45E-19 | 2.52E-18 | 8.11E-19 | 2.45E-19 | 1.90E-18 | 6.12E-19 | 2.82E-19 | 1.10E-18 | 5.50E-19 | 1.95E-19 | 8.13E-19 |
| | 50 | 1.25E-17 | 4.57E-18 | 2.32E-17 | 1.03E-17 | 2.79E-18 | 1.85E-17 | 9.61E-18 | 3.83E-18 | 1.70E-17 | 8.98E-18 | 3.55E-18 | 1.74E-17 |
| | 100 | 4.98E-17 | 1.26E-17 | 1.04E-16 | 3.76E-17 | 1.45E-17 | 7.89E-17 | 2.39E-17 | 1.18E-17 | 4.57E-17 | 2.66E-17 | 1.11E-17 | 5.07E-17 |
| 20 | 5 | 2.30E-31 | 4.37E-32 | 6.97E-31 | 8.96E-32 | 3.50E-32 | 2.00E-31 | 7.46E-32 | 2.14E-32 | 1.50E-31 | 6.74E-32 | 2.33E-32 | 1.56E-31 |
| | 10 | 4.62E-24 | 1.52E-24 | 1.08E-23 | 2.95E-24 | 6.22E-25 | 5.44E-24 | 2.62E-24 | 5.33E-25 | 5.44E-24 | 2.34E-24 | 1.14E-24 | 4.41E-24 |
| | 25 | 2.01E-19 | 7.32E-20 | 4.23E-19 | 1.35E-19 | 5.01E-20 | 2.78E-19 | 1.48E-19 | 1.04E-19 | 2.16E-19 | 1.16E-19 | 6.07E-20 | 2.14E-19 |
| | 50 | 6.29E-18 | 2.49E-18 | 1.37E-17 | 5.19E-18 | 2.26E-18 | 1.15E-17 | 4.63E-18 | 2.22E-18 | 1.04E-17 | 4.10E-18 | 2.48E-18 | 7.88E-18 |
| | 100 | 3.20E-17 | 6.88E-18 | 6.82E-17 | 2.48E-17 | 1.29E-17 | 4.75E-17 | 1.85E-17 | 5.24E-18 | 3.41E-17 | 1.89E-17 | 4.94E-18 | 3.68E-17 |
| 40 | 5 | 3.58E-33 | 4.67E-34 | 9.36E-33 | 1.35E-33 | 1.51E-34 | 3.46E-33 | 1.05E-33 | 1.86E-34 | 1.88E-33 | 9.41E-34 | 4.04E-34 | 2.23E-33 |
| | 10 | 4.27E-25 | 7.26E-26 | 1.57E-24 | 2.47E-25 | 1.24E-25 | 4.98E-25 | 1.66E-25 | 8.77E-26 | 3.38E-25 | 1.65E-25 | 6.83E-26 | 2.98E-25 |
| | 25 | 5.49E-20 | 1.78E-20 | 1.04E-19 | 3.71E-20 | 1.59E-20 | 6.48E-20 | 3.33E-20 | 1.36E-20 | 6.38E-20 | 3.32E-20 | 1.36E-20 | 6.23E-20 |
| | 50 | 3.11E-18 | 1.12E-18 | 6.34E-18 | 2.34E-18 | 6.32E-19 | 4.95E-18 | 2.16E-18 | 9.16E-19 | 3.56E-18 | 2.06E-18 | 1.16E-18 | 3.34E-18 |
| | 100 | 2.10E-17 | 5.90E-18 | 4.90E-17 | 1.75E-17 | 8.80E-18 | 3.35E-17 | 1.55E-17 | 6.59E-18 | 2.93E-17 | 1.22E-17 | 4.25E-18 | 2.37E-17 |
| MIDMBO Success Rate (%) | | 100 | 100 | 100 | 100 | 96 | 100 | 100 | 100 | 100 | 100 | 96 | 100 |

difference was shown to be statistically significant.

## 3. Experimental studies

In this section, the MBO and MIDMBO algorithms were implemented using four different values of NIs (8, 16, 24, 32), five different MR values in percentages (2, 6, 10, 20, 40), and five different MI values (5, 10, 25, 50, 100). The average, best, and worst results obtained from 30 independent runs are presented in tables and figures. Additionally, to examine the behavior of migration parameters that produced the best and worst average results for each test function when MIDMBO was applied with 32 subpopulations, the mean result at each iteration step over 30 independent runs was plotted. The same results were also plotted for MBO on the same graph to allow a comparative analysis. Moreover, using the average results obtained for all migration parameters with 32 islands, the performance of the algorithm across different migration parameters was visualized graphically and analyzed. The benchmark functions [34] used to analyze the MIDMBO algorithm are provided in Table 1. For both algorithms, the parameter values suggested in [12] were used. The algorithm parameters used are as follows:

- Number of individuals (birds) in the population (n): 51
- Number of neighbor solutions to be generated for each individual (k): 3
- Number of solutions to be shared with the next individual (x): 1
- Leader bird change interval (m): 10
- Maximum number of iterations (K): 2500

### 3.1. Algorithm performance

The results obtained from the Rosenbrock function are presented in Table 2 below. The outcomes after 2500 iterations were analyzed for different numbers of sub-populations. While keeping the number of NIs constant, the average, best, and worst values

**Table 4**
Optimization results for the Zakharov benchmark function.

| Zakharov | | Number of Island (NIs) | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 8 | | | 16 | | | 24 | | | 32 | | |
| Migration Rate (MR) | Migration Interval (MI) | Average | Best | Worst | Average | Best | Worst | Average | Best | Worst | Average | Best | Worst |
| No Migration (MBO) | | 93.8771 | 70.0435 | 118.251 | 81.7517 | 67.2699 | 106.729 | 84.8536 | 72.2978 | 104.519 | 82.2964 | 65.7492 | 99.988 |
| 2 | 5 | 0.010191 | 0.001366 | 0.036012 | 0.006391 | 0.002582 | 0.017352 | 0.004888 | 0.002035 | 0.01033 | 0.005012 | 0.001421 | 0.010328 |
| | 10 | 0.450773 | 0.200806 | 1.02817 | 0.548355 | 0.226736 | 1.16203 | 0.363198 | 0.180111 | 0.798371 | 0.310389 | 0.138084 | 0.476949 |
| | 25 | 10.886 | 4.05724 | 20.476 | 8.25528 | 3.30277 | 18.8105 | 6.80191 | 3.16566 | 10.2186 | 7.03163 | 3.00492 | 11.8111 |
| | 50 | 35.2726 | 23.3807 | 49.5387 | 32.2779 | 18.4934 | 49.5676 | 31.5737 | 16.3226 | 44.4873 | 26.3844 | 16.7091 | 35.5674 |
| | 100 | 72.5908 | 42.4558 | 100.798 | 62.4603 | 47.0551 | 80.0816 | 60.7321 | 47.0531 | 74.564 | 58.0707 | 43.9338 | 68.447 |
| 6 | 5 | 2.03503 | 3.06E-05 | 2.62E+01 | 3.56E-05 | 1.35E-05 | 1.20E-04 | 2.55E-05 | 8.66E-06 | 5.20E-05 | 2.10E-05 | 9.68E-06 | 4.02E-05 |
| | 10 | 0.109779 | 0.005154 | 2.72366 | 0.013554 | 0.005918 | 0.021723 | 0.010391 | 0.004482 | 0.02152 | 0.010445 | 0.004868 | 0.017812 |
| | 25 | 2.24844 | 1.11446 | 3.56327 | 1.81175 | 0.636774 | 4.28654 | 1.67647 | 0.60925 | 3.32408 | 1.62604 | 0.951742 | 2.79852 |
| | 50 | 15.3115 | 5.83687 | 24.0823 | 12.9457 | 8.11872 | 22.0009 | 12.5136 | 7.01437 | 18.9888 | 11.298 | 6.686 | 14.7565 |
| | 100 | 48.5627 | 25.9552 | 63.0658 | 41.8873 | 29.6555 | 54.5256 | 37.5366 | 27.1351 | 49.7059 | 34.1294 | 24.412 | 43.4492 |
| 10 | 5 | 6.81495 | 3.61E-06 | 5.14E+01 | 3.51E-06 | 8.62E-07 | 6.12E-06 | 3.16E-06 | 1.24E-06 | 6.66E-06 | 2.35E-06 | 1.08E-06 | 4.00E-06 |
| | 10 | 0.005042 | 0.001186 | 0.010727 | 0.003338 | 0.000948 | 0.009402 | 0.002316 | 0.000685 | 0.004327 | 0.002488 | 0.001314 | 0.006534 |
| | 25 | 1.21488 | 0.470678 | 1.98985 | 0.778279 | 0.402676 | 1.90388 | 0.865573 | 0.465125 | 1.6013 | 0.691632 | 0.404817 | 1.30459 |
| | 50 | 12.2899 | 5.98938 | 20.2851 | 9.14843 | 4.07839 | 14.5341 | 9.7071 | 6.48204 | 15.8171 | 8.19076 | 5.67124 | 11.6591 |
| | 100 | 38.1434 | 23.2486 | 60.2824 | 33.7014 | 20.0848 | 42.9621 | 30.9719 | 18.7766 | 43.0648 | 27.8168 | 17.7004 | 38.6728 |
| 20 | 5 | 4.55203 | 4.65E-07 | 2.60E+01 | 5.13E-07 | 2.03E-07 | 1.16E-06 | 2.85E-07 | 1.19E-07 | 4.70E-07 | 2.66E-07 | 1.05E-07 | 5.65E-07 |
| | 10 | 0.874037 | 0.000554 | 26.1816 | 0.000744 | 0.000294 | 0.00131 | 0.000561 | 0.000172 | 0.001105 | 0.000457 | 0.000164 | 0.001474 |
| | 25 | 0.532848 | 0.194659 | 1.12118 | 0.385998 | 0.187969 | 0.670941 | 0.360321 | 0.171376 | 0.547603 | 0.272285 | 0.119121 | 0.446326 |
| | 50 | 7.84123 | 4.25199 | 13.2085 | 6.1407 | 3.22415 | 10.3858 | 5.21741 | 3.0881 | 8.77367 | 4.83538 | 2.81888 | 8.03985 |
| | 100 | 31.5177 | 23.1587 | 42.7883 | 25.5149 | 13.7807 | 35.8071 | 24.1574 | 16.0185 | 30.4268 | 22.1686 | 16.3611 | 28.7465 |
| 40 | 5 | 4.33176 | 1.73E-07 | 2.65E+01 | 1.72E-07 | 3.59E-08 | 4.55E-07 | 1.59E-07 | 6.96E-08 | 4.42E-07 | 9.44E-08 | 2.09E-08 | 2.63E-07 |
| | 10 | 0.869815 | 0.000238 | 26.0717 | 0.000347 | 0.000151 | 0.000678 | 0.000276 | 7.38E-05 | 5.27E-04 | 0.000262 | 0.00014 | 0.000451 |
| | 25 | 0.355839 | 0.156072 | 0.850028 | 0.241879 | 0.133309 | 0.497343 | 0.213502 | 0.091745 | 0.338282 | 0.194893 | 0.075605 | 0.389359 |
| | 50 | 5.19498 | 3.04103 | 8.98752 | 4.52212 | 2.20824 | 7.66356 | 4.00865 | 2.14857 | 6.89501 | 3.70168 | 1.38457 | 5.49313 |
| | 100 | 25.355 | 12.3991 | 35.3752 | 23.121 | 13.543 | 34.8333 | 20.8633 | 14.1536 | 28.3777 | 19.7934 | 10.9749 | 29.0433 |
| MIDMBO Success Rate (%) | | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |

obtained for different MR and MI values are provided. Among these values, the best results are highlighted in blue, and the worst results are highlighted in yellow. In addition, the results generated with different MR and MI values were compared with the MBO results, and the cases in which MIDMBO performed better were presented using the success rate. According to the average results, MIDMBO produced the most successful results for NIs values of 8 and 32 at the 10-5 level, and for 16 and 24 at the 20-5 level. When considering the best results, the most successful outcomes were obtained for NIs of 8 at 40-5, 16 at 40-10, 24 at 10-5, and 32 at 6-5. Upon examining the data in the table, it can be observed that for all NIs, the most successful results were achieved when the MI was low. As the MI increases, the success of the results produced by the algorithm decreases [34].

Table 3 below displays the results obtained from the Rotated Hyper-Ellipsoid function. The results were analyzed for different NIs. MIDMBO achieved the most successful average, best, and worst results across all sub-populations when the MR was 40 % and the MI was 5. An examination of the data in the table reveals that, for all NIs, the most successful results were obtained when migration was frequent, and the number of migrating individuals was high. As MI increases, the performance of the algorithm tends to decrease. When analyzing all average results of MIDMBO, it is observed that it consistently outperforms MBO. Looking at the best results, MBO was more successful in only 2 cases, while MIDMBO outperformed it in the remaining 98 cases. Similarly, when examining the worst results, MIDMBO produced better outcomes than MBO in all cases.

The outcomes from the Zakharov function are shown in Table 4. MIDMBO produced the best average result for NIs = 8 at 10 %-10, and for other NIs values, at 40 %-5. For NIs = 16, 24, and 32, the most successful results were achieved when migration was frequent, and the number of migrating individuals was high. As MI increases, similar to the Rosenbrock and Rotated Hyper-Ellipsoid functions, the performance of the algorithm decreases. When considering the best results, the most successful outcomes for all NIs were achieved when MR was 40 % and MI was 5. Examining the average, best, and worst results of MIDMBO, it is clear that it consistently outperformed MBO in all cases. This outcome clearly demonstrates the contribution of the migration process to the MIDMBO algorithm.

The results derived from the Rastrigin function are detailed in Table 5 below. MIDMBO produced the best average result for NIs = 8 at 40 %-100, for NIs = 16 and 24 at 10 %-50, and for NIs = 32 at 10 %-25. When NIs = 8, poor results were observed for low MI values. Looking at the best results, the most successful outcomes were obtained for NIs = 8 at 40 %-50, for NIs = 16 and 24 at 10 %-25, and for NIs = 32 at 20 %-5. Examining all the average results of MIDMBO, it is evident that it outperformed MBO in 99 out of 100 cases. For the best results, MIDMBO was more successful than MBO by 56 %, 84 %, 100 %, and 96 % for each respective sub-population. The results

**Table 5**
Optimization results for the Rastrigin benchmark function.

| Rastrigin | | Number of Island (NIs) | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 8 | | | 16 | | | 24 | | | 32 | | |
| Migration Rate (MR) | Migration Interval (MI) | Average | Best | Worst | Average | Best | Worst | Average | Best | Worst | Average | Best | Worst |
| No Migration (MBO) | | 80.8992 | 23.051 | 189.203 | 56.907 | 22.0072 | 190.465 | 46.2378 | 18.5705 | 122.959 | 39.7725 | 15.9728 | 127.09 |
| 2 | 5 | 59.001 | 41.7883 | 82.5815 | 54.3247 | 35.8185 | 91.5361 | 43.5128 | 17.9093 | 76.6117 | 32.5683 | 16.9143 | 61.6874 |
| | 10 | 50.5107 | 33.8286 | 88.5512 | 42.5511 | 26.8639 | 67.6571 | 26.5322 | 2.98488 | 46.763 | 14.5596 | 0.995088 | 45.7681 |
| | 25 | 34.6577 | 20.8941 | 55.7177 | 21.6238 | 3.98038 | 44.7731 | 7.99271 | 3.01948 | 30.8437 | 6.38475 | 1.00783 | 14.9259 |
| | 50 | 27.5603 | 14.926 | 41.7883 | 19.5828 | 7.00287 | 32.8336 | 15.8809 | 8.15446 | 23.9013 | 13.2975 | 8.77152 | 18.9287 |
| | 100 | 71.2448 | 21.1077 | 250.478 | 28.5329 | 16.3077 | 55.219 | 25.3193 | 13.0168 | 44.7353 | 25.5664 | 14.9451 | 68.447 |
| 6 | 5 | 64.4401 | 42.7832 | 97.5058 | 55.3197 | 19.8992 | 86.5613 | 35.7522 | 2.98488 | 67.6571 | 22.6187 | 4.9748 | 43.7782 |
| | 10 | 53.3629 | 34.8236 | 80.5916 | 43.8445 | 18.9042 | 77.6067 | 22.0218 | 6.96471 | 37.8084 | 10.3807 | 0.994959 | 36.8135 |
| | 25 | 42.8164 | 19.8992 | 69.647 | 22.3086 | 4.00197 | 38.8034 | 7.80231 | 0.996986 | 28.8538 | 2.94864 | 0.003928 | 5.97492 |
| | 50 | 29.8488 | 16.9143 | 43.7782 | 13.4374 | 6.96583 | 26.8639 | 7.24093 | 2.31418 | 12.9445 | 8.40255 | 4.00381 | 15.7013 |
| | 100 | 29.9385 | 13.9379 | 66.877 | 21.1806 | 15.9291 | 31.2627 | 18.4131 | 9.94979 | 27.8643 | 18.7882 | 13.149 | 27.9006 |
| 10 | 5 | 66.2311 | 50.7429 | 108.45 | 56.9116 | 25.8689 | 87.5563 | 40.8596 | 5.96975 | 65.6672 | 26.9302 | 2.98488 | 44.7731 |
| | 10 | 57.8071 | 35.8185 | 81.5865 | 39.0355 | 7.95967 | 69.647 | 22.4861 | 2.98488 | 41.7883 | 14.6922 | 7.8E-09 | 37.8084 |
| | 25 | 42.1199 | 24.874 | 52.7328 | 19.5679 | 1.98999 | 54.7227 | 6.67282 | 0.012466 | 21.8891 | 2.65718 | 0.000441 | 9.94959 |
| | 50 | 32.704 | 7.042 | 51.7378 | 10.9171 | 4.09152 | 25.8689 | 5.94003 | 2.09469 | 11.0112 | 6.16692 | 3.10815 | 11.5746 |
| | 100 | 27.146 | 13.934 | 101.42 | 19.1923 | 12.524 | 27.0215 | 16.9496 | 10.9451 | 23.0326 | 15.7165 | 10.3689 | 22.1609 |
| 20 | 5 | 67.8893 | 45.7681 | 97.5058 | 55.0544 | 20.8941 | 92.5311 | 42.3521 | 14.9244 | 78.6016 | 25.4709 | 0 | 53.7277 |
| | 10 | 57.0443 | 33.8286 | 96.5109 | 40.2958 | 13.9294 | 65.6672 | 23.9785 | 6.96471 | 50.7429 | 13.2661 | 1.65E-11 | 32.8336 |
| | 25 | 47.4263 | 21.8891 | 78.6016 | 25.6699 | 4.9748 | 47.758 | 9.64867 | 0.995332 | 24.874 | 3.52663 | 0.001627 | 19.8992 |
| | 50 | 34.0608 | 9.95223 | 55.7177 | 12.7026 | 3.62376 | 25.8689 | 7.89783 | 2.31294 | 13.9319 | 7.46382 | 1.01466 | 13.3077 |
| | 100 | 23.5982 | 12.9345 | 49.2153 | 15.351 | 7.08351 | 24.8765 | 16.8059 | 8.04243 | 23.2785 | 14.9346 | 6.20837 | 21.3037 |
| 40 | 5 | 73.3117 | 38.8034 | 95.5159 | 56.0825 | 32.8336 | 77.6066 | 41.7551 | 11.9395 | 79.5966 | 23.1494 | 3.97984 | 40.7933 |
| | 10 | 56.7548 | 34.8235 | 78.6017 | 37.9742 | 10.9445 | 65.6672 | 27.1624 | 2.98488 | 46.763 | 14.0621 | 3.18E-13 | 33.8286 |
| | 25 | 42.6174 | 20.8941 | 66.6622 | 24.874 | 4.97685 | 50.7429 | 14.5931 | 2.98494 | 32.8336 | 8.36664 | 0.000109 | 24.874 |
| | 50 | 29.9999 | 6.5234 | 44.7731 | 16.9628 | 6.85684 | 43.7782 | 11.2774 | 5.09783 | 20.9167 | 9.03287 | 3.06012 | 17.9111 |
| | 100 | 23.4617 | 13.9773 | 35.8187 | 18.8111 | 9.06622 | 27.2796 | 16.5466 | 9.94965 | 24.9143 | 16.0269 | 9.01948 | 20.9458 |
| MIDMBO Success Rate (%) | | 100 | 56 | 96 | 96 | 84 | 100 | 100 | 100 | 100 | 100 | 96 | 100 |

where MBO outperformed MIDMBO were mostly observed for NIs = 8. As for the worst results, MIDMBO was only less successful than MBO for one case.

Table 6 below presents the results obtained from solving the Schwefel function using the MBO and MIDMBO algorithms for various NIs, MR, and MI. MIDMBO produced the best average results for NIs = 8 at 10 %-100, for NIs = 16 at 10 %-50, for NIs = 24 at 20 %-50, and for NIs = 32 at 10 %-25. When NIs = 8 and MI was low, poor results were observed. This is due to frequent migration and the small NIs, causing the individuals within islands to quickly become like one another. As the individuals within an island become similar, different regions of the search space cannot be represented, and a high-quality search is not possible. Looking at the best results, the most successful outcomes were obtained for NIs = 8 at 6 %-50, for NIs = 16 and 32 at 20 %-25, and for NIs = 24 at 6 %-25. When considering all the average results, MIDMBO outperformed MBO in 89 out of 100 cases. Of these, 10 cases were for NIs = 8, and 1 case was for NIs = 16. Regarding the best results, MBO was more successful in 4 cases for NIs = 8, while MIDMBO was superior in all other cases. For the worst results, MIDMBO outperformed MBO for NIs = 16, 24, and 32 in a proportional manner. Based on all results from the Schwefel function, as NIs increases, the performance of MIDMBO improves [34].

According to the results for the Rosenbrock function presented in Table 2, the successful outcomes marked in blue are generally concentrated around the middle sections of the table. Conversely, the unsuccessful results, indicated in yellow, tend to shift toward the boundary values. Based on these data, it can be concluded that the effects of the MR and MI parameters and their interactions on performance have been successfully investigated through the conducted experiments. The results for the Rotated Hyper-Ellipsoid function shown in Table 3 demonstrate that a stable search structure was established using MIDMBO. All of the most successful results were obtained with the same migration parameter values. Similarly, nearly all of the least successful results correspond to the same parameter values. According to the results obtained from the Zakharov benchmark function in Table 4, a stable search pattern is also observed. While most of the best results were achieved with the same migration parameters, all of the unsuccessful results were generated using these parameters. Particularly, the results in Tables 3 and 4 clearly illustrate the significant impact of migration parameter selection on algorithm performance. For the Rastrigin benchmark function presented in Table 5, the most successful results are generally clustered in the central part of the table, whereas the unsuccessful results are more scattered. The scattered distribution of results, compared to other functions, is attributed to the multimodal nature of the Rastrigin function. Similarly, according to Table 6, which presents results for the Schwefel function, another multimodal function, the best results are found in the middle region of the

**Table 6**
Optimization results for the Schwefel benchmark function.

| Schwefel | | Number of Island (NIs) | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 8 | | | 16 | | | 24 | | | 32 | | |
| Migration Rate (MR) | Migration Interval (MI) | Average | Best | Worst | Average | Best | Worst | Average | Best | Worst | Average | Best | Worst |
| No Migration (MBO) | | 4544.07 | 3565.29 | 5354 | 4217.86 | 3565.29 | 4761.81 | 4032.07 | 3329.93 | 4757.26 | 4021.45 | 3326.89 | 4521.9 |
| 2 | 5 | 4911.48 | 3923.64 | 5788.28 | 4787.52 | 3208.46 | 5935.56 | 3609.12 | 2494.79 | 5592.39 | 2824.34 | 830.586 | 4163.55 |
| | 10 | 4573.93 | 3448.37 | 5472.44 | 3916.65 | 1188.94 | 5232.53 | 2756.77 | 830.586 | 4640.33 | 1895.16 | 592.192 | 3917.57 |
| | 25 | 4338.73 | 2851.62 | 6061.59 | 2601.59 | 1304.34 | 4165.06 | 1740.98 | 834.623 | 4280.47 | 1120.29 | 481.136 | 1779.61 |
| | 50 | 4242.83 | 2853.15 | 5829.27 | 2597.95 | 1544.28 | 3802.16 | 2122.45 | 1575.5 | 3088.5 | 2085.17 | 1541.85 | 2848.59 |
| | 100 | 3435.27 | 2255.18 | 5190.03 | 3224.61 | 2613.64 | 4512.79 | 2939.22 | 2018.01 | 3445.33 | 2895.03 | 2134.93 | 3679.55 |
| 6 | 5 | 4814.91 | 1996.75 | 6072.21 | 4114.05 | 1898.05 | 5475.47 | 2815.13 | 1185.9 | 5306.94 | 2247.59 | 592.192 | 3445.33 |
| | 10 | 4532.98 | 2734.7 | 5686.54 | 3790.83 | 2254.88 | 5829.27 | 1986.37 | 1068.98 | 3679.17 | 1803.85 | 593.709 | 3088.5 |
| | 25 | 4477.81 | 2139.48 | 6426.01 | 2533.56 | 1185.9 | 4752.7 | 1221.41 | 236.878 | 2021.04 | 826.529 | 236.913 | 1661.17 |
| | 50 | 3827.47 | 1781.15 | 5707.8 | 1815.06 | 1184.49 | 2848.59 | 1185.84 | 596.781 | 1667.22 | 1318.22 | 592.735 | 2134.98 |
| | 100 | 3550.61 | 2268.84 | 4989.58 | 2524.61 | 1779.48 | 3370.48 | 2459.69 | 1542.74 | 3332.49 | 2306.75 | 1422.2 | 2966.17 |
| 10 | 5 | 4690.55 | 3569.84 | 6257.5 | 3929.66 | 2371.8 | 5429.93 | 2942.58 | 1187.42 | 4997.17 | 2017.75 | 710.631 | 3805.2 |
| | 10 | 4798.86 | 3683.73 | 6172.43 | 3642.98 | 2018 | 5329.71 | 2195.61 | 593.709 | 3568.32 | 1586.21 | 592.192 | 2608.68 |
| | 25 | 4490.42 | 2968.54 | 5473.95 | 2319.31 | 1185.9 | 3922.12 | 1359.87 | 593.845 | 2371.8 | 679.87 | 355.316 | 1782.64 |
| | 50 | 3768.6 | 2021.04 | 5354 | 1441.89 | 713.536 | 2490.24 | 1081.47 | 473.776 | 1659.83 | 849.619 | 241.91 | 1421.41 |
| | 100 | 3401.34 | 1898.74 | 4755.74 | 2332.58 | 1541.31 | 3198.58 | 2148.44 | 1424.41 | 2724.32 | 2083.94 | 1421.29 | 2729.03 |
| 20 | 5 | 4869.98 | 3644.25 | 5943.18 | 3958.77 | 2368.77 | 5473.95 | 3432.53 | 2014.97 | 4500.64 | 2312.73 | 710.631 | 4040.56 |
| | 10 | 4676.58 | 3408.89 | 6787.4 | 3449.84 | 1662.69 | 5476.99 | 2227.14 | 829.069 | 3805.2 | 1743.72 | 947.507 | 3088.5 |
| | 25 | 4448.26 | 3544.03 | 5592.39 | 2560.89 | 473.754 | 4515.83 | 1419.52 | 475.277 | 3682.21 | 916.437 | 118.578 | 2014.97 |
| | 50 | 3934.32 | 1784.16 | 4872.66 | 1799.43 | 599.548 | 3326.89 | 1059.79 | 594.887 | 1541.3 | 891.839 | 475.439 | 1787.09 |
| | 100 | 3477.22 | 1783.77 | 5114.09 | 2044.34 | 1421.31 | 2846.01 | 1941.82 | 1304.69 | 2610.24 | 1814.07 | 1066.19 | 2260.44 |
| 40 | 5 | 4986.55 | 3220.63 | 6380.47 | 3592.77 | 2353.58 | 5709.31 | 2861.7 | 1304.34 | 4620.6 | 2449.95 | 592.192 | 3662.47 |
| | 10 | 4559.95 | 2971.58 | 6189.14 | 3578.14 | 2136.44 | 5232.53 | 2737.39 | 1188.94 | 4159 | 1735.73 | 592.192 | 3308.67 |
| | 25 | 4802.05 | 3328.41 | 6307.57 | 2920.97 | 1302.82 | 4632.75 | 1905.94 | 830.586 | 2851.62 | 1070.61 | 236.884 | 2371.8 |
| | 50 | 4448.15 | 2971.58 | 6545.97 | 2278.58 | 1189.52 | 3442.3 | 1692.78 | 948.453 | 2727.12 | 1430.57 | 711.887 | 1909.86 |
| | 100 | 3519.45 | 2251.87 | 4994.13 | 2274.72 | 1185.97 | 3085.59 | 2336.1 | 1778.16 | 2965.92 | 2173.94 | 1304.4 | 2725.79 |
| MIDMBO Success Rate (%) | | 60 | 84 | 28 | 96 | 100 | 60 | 100 | 100 | 88 | 100 | 100 | 100 |

table. Examination of the unsuccessful results reveals that they generally occur in the same region. Experiments conducted on benchmark functions with diverse characteristics demonstrate that migration parameters have a direct effect on algorithm performance. It is evident that the selected migration parameter values were appropriate, resulting in a consistent search behavior.

The success rate is a proportional measure used to compare the results of the MBO algorithm and the MIDMBO algorithm on an equal number of islands, in scenarios where no migration occurs. For each island count, twenty-five different MIDMBO results generated with varying migration parameters were compared to the MBO results, and the percentage of cases in which MIDMBO outperformed MBO was reported as the success rate. This comparison was presented separately for average, best, and worst results. The purpose of this analysis is to demonstrate the contribution of the migration process to the MBO algorithm. When benchmark functions are examined individually, MIDMBO outperforms MBO across all results for the Rotated Hyper-Ellipsoid and Zakharov functions, with a success rate of 100 %. For the Rastrigin and Schwefel functions, MIDMBO shows superior performance for nearly all migration parameter settings. According to the Rosenbrock function results, in twelve comparison points, eight exhibit a success rate above 50 %, indicating that MIDMBO achieved a more effective search. Among these, three success rate values reach 100 %. Overall, when considering success rate values collectively, it is evident that the migration process significantly contributes to the performance of the MIDMBO algorithm, enabling it to achieve better results than MBO in many cases.

### 3.2. *Exploration and exploitation phases*

In population-based metaheuristic algorithms, optimization occurs in two phases: exploration and exploitation [46]. During the exploration phase, solutions explore the entire search space, reducing the risk of getting stuck in local optima. This helps to identify regions where good solutions are likely to be found. In the exploitation phase, a more detailed search is performed among the good solutions to discover even better results. Since migration parameters directly affect search behavior, their impact on both the exploration and exploitation phases should be examined. In this phase of the study, the most successful (MIDMBO-1) and least successful (MIDMBO-2) results of MIDMBO were compared with the MBO results throughout the iterations.

The convergence graphs showing the most successful and least successful MIDMBO results for each function are presented in Fig. 1. All results are presented for 32 islands. In Fig. 1 (a), the results obtained for the Rosenbrock function are shown graphically. Upon

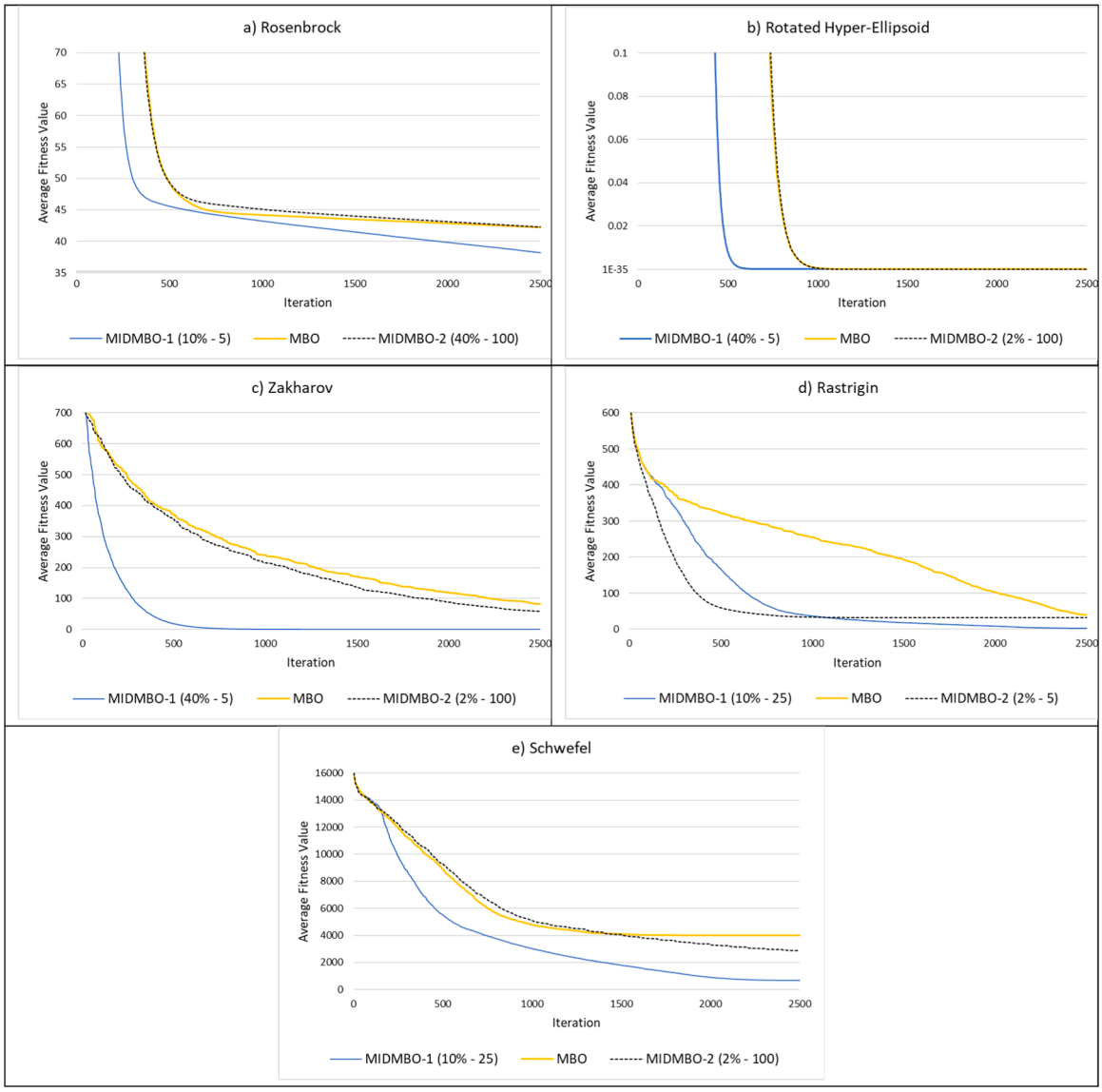


**Fig. 1.** Convergence graphs for the benchmark functions.

examining the behavior of MIDMBO-1 (10 %-5), it is observed that the algorithm performs a faster search than MBO during the exploration phase and continues to produce good results by performing a detailed search during the exploitation phase. On the other hand, MIDMBO-2 (40 %-100) shows behavior like MBO but continues to search during the exploitation phase, producing better results towards the end. Fig. 1 (b) presents the results obtained for the Rotated Hyper-Ellipsoid function. The convergence graphs for MIDMBO-1 (40 %-5) and MIDMBO-2 (2 %-100) are shown alongside MBO. In the MIDMBO-1 graph, it is evident that the algorithm conducts a faster search during the exploration phase, reaching good results in early steps. During the exploitation phase, although the results do not appear to be excellent due to reaching very small values, a detailed search continues to yield good results, as seen in the results in the tables. In the MIDMBO-2 graph, it shows behavior similar to MBO, but the final result produced is more successful.

Fig. 1 (c) displays the results obtained for the Zakharov function. According to the results for MBO, MIDMBO-1 (40 %-5), and MIDMBO-2 (2 %-100), it is observed that MBO performs a very slow search and, despite continuing to search in the later iterations, fails to produce sufficiently good results. In the MIDMBO-1 graph, the algorithm performs a very fast search during the exploration phase and continues to produce good results by conducting a detailed search during the exploitation phase. MIDMBO-2 shows similar behavior to MBO but produces better results in the exploitation phase. The results for the Rastrigin function are shown in Fig. 1 (d). In this case, MBO performs a very slow search, producing better results in the latter part of the iterations. In the MIDMBO-1 (10 %-25) graph, the algorithm performs a slower search during the exploration phase but continues to produce good results by conducting a detailed search during the exploitation phase. In the MIDMBO-2 (2 %-5) graph, it is observed that the algorithm performs a faster search due to frequent migration, but fails to improve its results during the exploitation phase. Fig. 1 (e) presents the results for the

**Table 7**
T-test results.

| T-Test | | NIs | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 8 | | 16 | | 24 | | 32 | |
| | | Average | Best | Average | Best | Average | Best | Average | Best |
| Rosenbrock | MR-MI | 10-5 | 40-5 | 20-5 | 40-10 | 20-5 | 10-5 | 10-5 | 6-5 |
| | p | 3.005E-22 | 2.122E-12 | 4.808E-10 | 8.663E-05 | 3.877E-07 | 5.519E-07 | 6.916E-07 | 2.762E-25 |
| Rotated Hyper-Ellipsoid | MR-MI | 40-5 | | 40-5 | | 40-5 | | 40-5 | |
| | p | 5.949E-09 | | 1.189E-11 | | 4.527E-12 | | 7.636E-11 | |
| Zakharov | MR-MI | 10-10 | 40-5 | 40-5 | | 40-5 | | 40-5 | |
| | p | 2.581E-11 | 1.204E-37 | 5.663E-27 | | 2.985E-30 | | 3.600087E-33 | |
| Rastrigin | MR-MI | 40-100 | 40-50 | 10-50 | 10-25 | 10-50 | 10-25 | 10-25 | 20-25 |
| | p | 4.858E-07 | 3.483E-06 | 1.989E-06 | 5.051E-05 | 1.212E-08 | 1.670E-08 | 2.197E-08 | 0.007707 |
| Schwefel | MR-MI | 10-100 | 6-50 | 10-50 | 20-25 | 20-50 | 6-25 | 20-25 | 20-25 |
| | p | 4.522E-11 | 2.537E-05 | 2.544E-33 | 1.368E-11 | 1.511E-36 | 3.228E-33 | 6.498E-43 | 1.577E-38 |

Schwefel function. MBO performs very slow search and is unable to continue the search in the exploitation phase. In the MIDMBO-1 (10 %-25) graph, the algorithm conducts a very fast search during the exploration phase and continues to produce good results by conducting a detailed search during the exploitation phase. MIDMBO-2 (2 %-100) shows behavior similar to MBO but produces better results in the exploitation phase due to the effect of migration. When all the results are evaluated together, it is evident that varying MR and MI values have different effects on the functions, influencing the results of the MIDMBO algorithm and altering its behavior in the exploration and exploitation phases [34].

The results obtained with MIDMBO show significant differences compared to the results obtained with MBO. In the study, whether these differences are statistically significant was investigated through hypothesis testing. A t-test was performed to determine if there is a significant difference between the MIDMBO and MBO results for the most successful average and best results for each number of islands. The two-sample t-test is a commonly used method to test the equality of two means. Before performing the t-test, it is necessary to determine whether the variances are homogeneous. As a preliminary test to assess the suitability of the two-sample t-test, the homogeneity of the variances is evaluated using tests such as the F-test. In this study, the F-test was first conducted, and the homogeneity of the variances was checked at the 5 % significance level. For groups with equal variances, a two-sample t-test assuming equal variances was used, while for groups with unequal variances, a two-sample t-test assuming unequal variances was performed [47,48].

When the p-value obtained from the t-test is less than 0.05, it indicates that the optimization performed using MIDMBO yields significantly better results than MBO at the 5 % significance level. As shown in Table 7 below, the t-test results between the MBO and MIDMBO groups yield values very close to zero. The results obtained from both the MBO and MIDMBO algorithms are statistically significant. The t-test results demonstrate that the migration process substantially improves the outcomes compared to MBO, highlighting the significant contribution of migration to the overall performance of the algorithm [34].

The contribution of the migration process to the algorithm's performance is clearly demonstrated by the obtained results. In addition, the t-test results confirm that this contribution is statistically significant. According to the findings, while the average and best results for the Rosenbrock, Rastrigin, and Schwefel functions were achieved at different parameter levels, for the Rotated Hyper-Ellipsoid function, they occurred at the same levels. For the Zakharov function, variation was observed only at the level where the number of islands was set to 8. In the subsequent sections of the study, the extent to which changes in migration parameters explain the variation in performance values obtained through the MIDMBO algorithm was investigated.

### *3.3. Model building using regression analysis and artificial neural networks*

In this stage, regression analysis and artificial neural network models were developed to examine the impact of the migration process and parameter variations on the algorithm's performance.

Regression analysis is used to estimate the value of a dependent variable ($y$) based on the range of values of one or more independent variables ($x$). Simple Linear Regression refers to a model with a single independent variable and describes the dependency of the dependent variable on it. It is expressed as shown in Eq. (1), where $\varepsilon$ denotes the random error term and $\beta_0$ is the intercept term:

$$Y = \beta_0 + \beta_1 x + \varepsilon \tag{1}$$

Multiple Linear Regression (MLR) is a statistical technique used to predict the outcome of a dependent variable using more than one independent variable. Its primary objective is to model the linear relationship between the dependent variable and the independent variables under analysis. In this study, a model with three independent variables (MR, MI, and NIs) was employed for the regression analysis. The model with three independent variables is given in Eq. (2):

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \varepsilon \tag{2}$$

Polynomial regression is a special case of regression analysis used when the data fits a non-linear relationship between the dependent and independent variables, represented by a polynomial equation [49]. In addition to MLR, this study also employed interaction, pure quadratic, and quadratic polynomial models involving the three independent variables. The interaction model takes into account both the main effects of the independent variables and their mutual interactions [50]. It is formulated as shown in Eq. (3):

**Table 8**
Coefficients of determination.

| Benchmark function | Model | Adjusted R-Squared | p-value |
|---|---|---|---|
| Rosenbrock | Multivariate linear | 0.569 | 3.87e-18 |
| | **Quadratic** | **0.824** | 5.96e-32 |
| | Pure quadratic | 0.814 | 4.27e-33 |
| | Interactions | 0.57 | 2.06e-16 |
| Rotated Hyper-Ellipsoid | Multivariate linear | 0.728 | 1.15e-27 |
| | **Quadratic** | **0.915** | 3.78e-46 |
| | Pure quadratic | 0.768 | 1.21e-28 |
| | Interactions | 0.869 | 4.27e-40 |
| Zakharov | Multivariate linear | 0.777 | 7.57e-32 |
| | **Quadratic** | **0.912** | 2.15e-45 |
| | Pure quadratic | 0.827 | 1.59e-34 |
| | Interactions | 0.858 | 1.88e-38 |
| Rastrigin | Multivariate linear | 0.556 | 1.65e-17 |
| | **Quadratic** | **0.857** | 4.76e-36 |
| | Pure quadratic | 0.8 | 1.32e-31 |
| | Interactions | 0.603 | 5.33e-18 |
| Schwefel | Multivariate linear | 0.678 | 3.84e-24 |
| | **Quadratic** | **0.89** | 4.76e-41 |
| | Pure quadratic | 0.85 | 2.64e-37 |
| | Interactions | 0.711 | 2.94e-24 |

**Table 9**
Quadratic models.

| | Quadratic Models |
|---|---|
| Rosenbrock | $y = 40.915 - 0.0573 * MR + 0.08808 * - 0.1005 * NIs + 0.00023991 * MR * MI + 0.00016886 * MR * NIs + 0.0003048 * MI * NIs + 0.00095447 * MR^2 - 0.00067389 * MI^2 + 0.0014097 * NIs^2$ |
| Rotated Hyper-Ellipsoid | $y = 6.8379e-18 - 1.1212e-18 * MR + 5.7587e-19 * MI - 4.7641e-19 * NIs - 1.1808e-20 * MR * MI + 9.9057e-21 * MR * NIs - 1.0126e-20 * MI * NIs + 2.3086e-20 * MR^2 + 1.9636e-21 * MI^2 + 1.1178e-20 * NIs^2$ |
| Zakharov | $y = 6.4843 - 0.90811 * MR + 0.35337 * MI - 0.3174 * NIs - 0.0094879 * MR * MI + 0.0026629 * MR * NIs - 0.0039336 * MI * NIs + 0.021329 * MR^2 + 0.0025347 * MI^2 + 0.0066498 * NIs^2$ |
| Rastrigin | $y = 94.15 - 0.21998 * MR - 1.3638 * MI - 3.1258 * NIs - 0.0040859 * MR * MI + 0.00022828 * MR * NIs + 0.012292 * MI * NIs + 0.0081023 * MR^2 + 0.0094102 * MI^2 + 0.033598 * NIs^2$ |
| Schwefel | $y = 7733.5 - 43.198 * MR - 70.665 * MI - 271.92 * NIs - 0.077643 * MR * MI - 0.26492 * MR * NIs + 0.76541 * MI * NIs + 1.1561 * MR^2 + 0.46092 * MI^2 + 3.5074 * NIs^2$ |

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_{12} X_1 X_2 + \beta_{13} X_1 X_3 + \beta_{23} X_2 X_3 + \varepsilon \tag{3}$$

The pure quadratic regression model consists of the intercept term, the linear terms of the independent variables, and the squared terms of each variable. This model is given in Eq. (4):

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_{11} X_1^2 + \beta_{22} X_2^2 + \beta_{33} X_3^2 + \varepsilon \tag{4}$$

The quadratic regression model includes both the squared terms and the interaction terms of the independent variables [51]. The complete model is expressed in Eq. (5):

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_{12} X_1 X_2 + \beta_{13} X_1 X_3 + \beta_{23} X_2 X_3 + \beta_{11} X_1^2 + \beta_{22} X_2^2 + \beta_{33} X_3^2 + \varepsilon \tag{5}$$

Using regression analysis, the coefficients of determination ($R^2$) were employed to investigate and compare the explanatory power of the changes in the MR, MI, and NIs parameters on the results obtained with MIDMBO. For the test functions, models such as Multivariate linear, quadratic, pure quadratic, and interactions were developed based on the average values obtained. The dependent variable in these models is the average fitness values of the test functions, while the independent variables are the MR, MI, and NIs parameters. All models demonstrated a high level of significance in terms of explanatory power. The adjusted squared value was used to determine the best explanatory model. The coefficients of determination for the models are presented in Table 8.

For each benchmark function, the Quadratic model has the highest coefficient of determination. The Quadratic model explains the effect of changes in the parameter values on the fitness values most effectively. Including not only the linear effects but also the interaction and squared terms of the variables as separate predictors in the quadratic regression model enhances the model's capacity by accounting for nonlinear effects. The high values of the coefficient of determination ($R^2$) indicate that the linear, interaction, and quadratic effects of the MR, MI, and NIs parameters all contribute significantly to the prediction performance.

Table 9 below shows the regression coefficients for the Quadratic model. When the quadratic functions and regression coefficients are examined, it is observed that the coefficients of *MR* and *NIs* are consistently negative, while the coefficients of $MR^2$ and $NIs^2$ are consistently positive. However, it is noteworthy that the direction (sign) of the *MI* coefficient varies. Furthermore, for the $MR * NIs$ interaction term, only the coefficient corresponding to the Schwefel function takes a negative value, whereas for the quadratic term

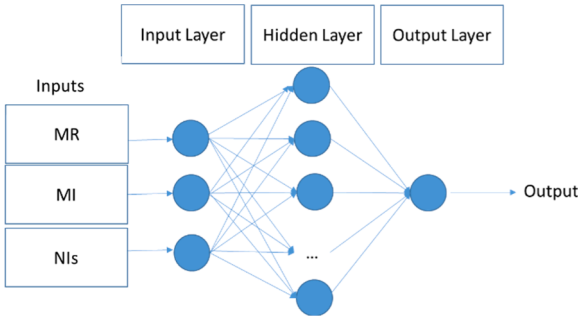


**Fig. 2.** ANN structure.

$MI^2$, only the Rosenbrock function exhibits a negative coefficient. This indicates that changes in the parameter values tend to have a similar directional impact on the fitness values across different benchmark functions.

An ANN, fundamentally designed to resemble the human central nervous system, is a distributed processor composed of basic computational units called neurons. Artificial neurons are capable of receiving a combination of signals from external sources or other neurons. In this study, a multilayer perceptron is employed. The multilayer perceptron has the ability to solve problems that are not linearly separable. This structure consists of at least three layers of nodes. In the intermediate or hidden layers, each node is a neuron that utilizes a nonlinear activation function. Neurons can transform signals through a specific function referred to as the activation function. The input value located at each node in the previous layer is multiplied by a varying connection weight. In this way, data is stored within the network parameters, particularly in the weights associated with each connection. The analysis of the system is left to the network itself, which independently establishes its own criteria to reproduce behavior and thus enables it to make predictions about the system. ANNs demonstrate strong performance when attempting to forecast the trend of a variable by observing the past behavior of various factors within an evolving system [52–54].

ANN architecture employed in this study consists of three layers: an input layer, a hidden layer, and an output layer. For each function, the ANN was constructed using three input variables (MR, MI, and NIs) corresponding to the independent variables, and one output representing the dependent variable. The structure of the developed ANN model is shown in Fig. 2.

In ANNs, all data are divided into three subsets: training, testing, and validation. The training dataset is used to identify patterns within the data and to train the model. The validation dataset is utilized during the training process to fine-tune the model's hyperparameters and prevent overfitting, ensuring that the model generalizes well to new data. Finally, the testing dataset is used to evaluate the model's performance after training is complete, providing an unbiased assessment of how well the model can predict outcomes on unseen data [54]. The training subset constitutes a significant portion of the dataset and is used to optimize the network's weights and bias values throughout the learning process. In this study, the Levenberg-Marquardt algorithm was selected as the training algorithm, which iteratively adjusts the weights and is commonly used to solve nonlinear least squares problems [55]. The validation subset is used to adjust the hyperparameters of the ANN model during the training process. It helps assess the model's performance on unseen data and aids in making decisions regarding necessary adjustments. The test subset is employed to evaluate the final performance of the trained ANN model. It provides an unbiased assessment of the model's generalization capabilities on unseen data [56]. For ANNs, validation is performed to assess the reliability and generalizability of the models developed based on the predicted dataset. Prediction models are typically trained on a specific dataset. Through validation, the model is tested on new and unseen data, allowing predictions to be generalized beyond the training set [57].

One of the most important results to be evaluated is the correlation between the predicted values generated by the ANN and the experimental values. Values close to 1 indicate that the ANN model has effectively captured the data patterns and variations. The alignment of ANN predictions with the experimental data suggests high accuracy. High correlation values demonstrate that the ANN model is reliable for predicting experimental outcomes [55].

In this study, the correlations between the outputs and targets are demonstrated using R values. The relationship between the output and the target for each function is presented in Fig. 3, along with the corresponding R values. An R value close to 1 indicates a strong relationship, while a value close to 0 suggests a random relationship. For models where R is greater than 0.8, it can be concluded that there is a strong correlation, reflecting a good fit to the observed data points [54]. Thus, R values close to 1 highlight the impact of the target values on the output. As shown below, the R values for the test data are 0.97988, 0.98888, 0.99343, 0.98030, and 0.97545, respectively. When these R values are examined together, it is evident that a successful model has been established, demonstrating a strong relationship between the target and the output. The ANN model, which uses the results of all functions together, was constructed with 4 inputs (function, MR, MI, NIs) and 1 output. The result of 0.99031 obtained from the test data indicates that the functions have a significant effect on the output.

When analyzing the accuracy of ANN models, the determination coefficients indicate that they closely align with models related to learning [58]. The determination coefficient, $R^2$, between the outputs and targets is a measure of how well the variation in the output is explained by the targets and outputs. High $R^2$ values indicate a good match between the observed and predicted data [59]. In this study, the performance level of the developed ANN models was as follows: for Rosenbrock, 96 % ($R^2 = 0.96016$), for Rotated Hyper-Ellipsoid, 97 % ($R^2 = 0.97788$), for Zakharov, 98 % ($R^2 = 0.98690$), for Rastrigin, 96 % ($R^2 = 0.96098809$), and for Schwefel, 95

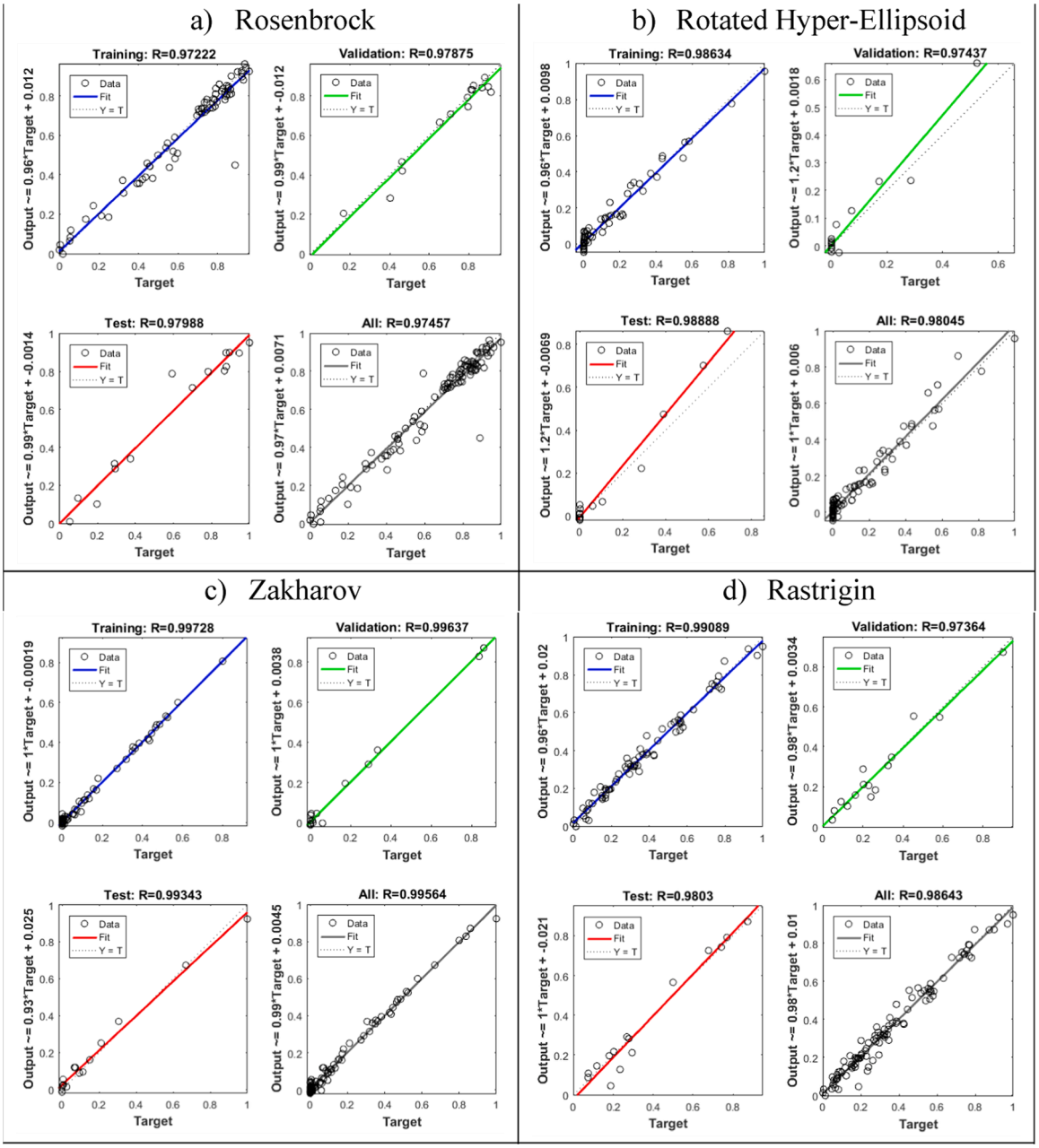


**Fig. 3.** ANN model performance analysis plot.

% ($R^2 = 0.951502703$). To compare the prediction success of the models in this study, the $R^2$ values were considered as the performance indicator. For Quadratic Regression, the Adjusted R-Squared values were taken into account, while for the ANN, the R-Squared value of the linear model between the output and target was considered. The results of the models obtained for the test functions are provided in Table 10 below.

According to the results obtained, the training, testing, and validation outcomes of the ANN model exhibit higher explanatory power compared to the results of the quadratic regression model. To statistically validate the explanatory power of the models, the non-parametric Wilcoxon Signed-Rank test was employed. Since the significance level obtained from the test was $0.043 < 0.05$, it indicates that there is a statistically significant difference between the two models in terms of coefficient of determination performance [25]. In the problem of investigating the effect of migration parameters on the fitness value, since the analytical form of the function is unknown, both regression analysis and the use of ANN, which involves fewer assumptions and explores relationships beyond the quadratic form, were utilized for prediction. After conducting research with the ANN, it can be stated that the results obtained were

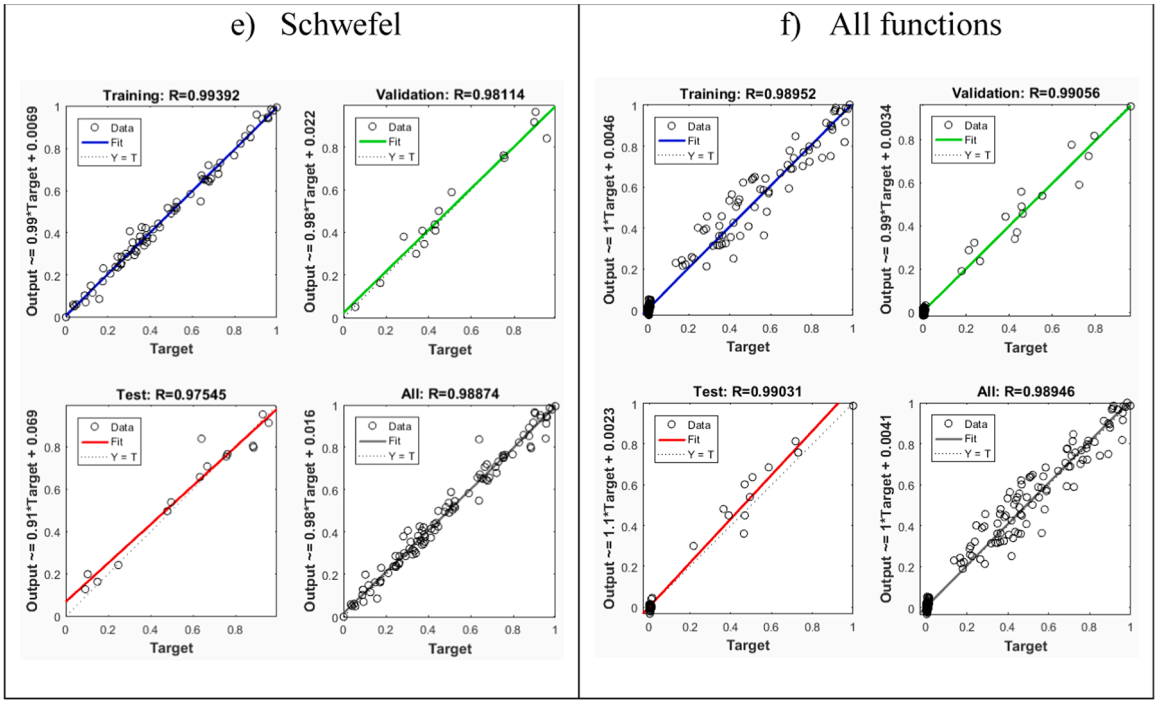


**Fig. 3.** (*continued*).

**Table 10**
Determination coefficients.

| Performance Indicator | Rosenbrock | Rotated Hyper-Ellipsoid | Zakharov | Rastrigin | Schwefel |
|---|---|---|---|---|---|
| Quadratic Regression Adjusted R-Squared | 0.824 | 0.915 | 0.912 | 0.857 | 0.89 |
| ANN R | 0.97988 | 0.98888 | 0.99343 | 0.98030 | 0.97545 |
| ANN R-Squared | 0.960164814 | 0.977883654 | 0.986903165 | 0.96098809 | 0.951502703 |

more effective in prediction. One of the advantages of the ANN is its exploratory nature, while the ability of the established models to allow the analysis of variables and interactions is an advantage of regression analysis.

## 4. Conclusion

In this study, the neighborhood-based optimization algorithm known as MBO was implemented on a distributed HPC system using multiple islands. The aim was to develop a new algorithm, referred to as MIDMBO, that would produce more successful results than MBO. Additionally, the study investigated migration parameters that could enable the distributed algorithm to yield improved outcomes. Using the results obtained from the experiments, the relationship between the migration parameters and the fitness value was modeled. Through regression analysis, different model structures were developed for each benchmark function, and based on the coefficient of determination, the Quadratic model was identified as the most successful. All constructed mathematical models were found to be statistically significant. The quadratic models provided insights into the direction and magnitude of the effects of parameters, their interactions, and their squared terms on performance.

Consequently, this study statistically demonstrates the significance of migration parameters used in island-based distributed metaheuristic algorithms and their relationship to algorithm performance. It establishes that changes in migration parameters have a highly significant impact on algorithm performance. As a secondary modeling approach, an ANN analysis was also conducted. The high coefficients of determination observed in the ANN models between the target and the fitness value further support the effectiveness of the proposed models. Moreover, the determination coefficients obtained from the ANN analysis were higher than those from the regression analysis. The consistency of these findings across all different benchmark functions supports one another. These results also indicate that the success of the developed MIDMBO algorithm is not random but statistically significant, highly correlated with the parameters, and exhibits consistent performance across various benchmarking functions.

## Acknowledgement

Some of the computations presented in this study were carried out at TÜBİTAK ULAKBİM, High Performance and Grid Computing Center (TRUBA resources). We would like to express our sincere thanks to all those who contributed to the establishment and maintenance of the TRUBA infrastructure.

## Data availability

No data was used for the research described in the article.

## References

[1] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, GSA: A gravitational search algorithm, Inf. Sci. 179 (13) (2009) 2232–2248.
[2] O. Erol, I. Eksin, A new optimization method: Big Bang–Big Crunch, Adv. Eng. Softw. 37 (2) (2006) 106–111.
[3] H. Noor, Z. Mostafa, M. Rammohan, N. Ponnuthurai, An improved differential evolution algorithm using efficient adapted surrogate model for numerical optimization, Inf. Sci. 451 (2018) 326–347.
[4] K. Manoela, M. Marley, T. Ricardo, PSO+: A new particle swarm optimization algorithm for constrained problems, Appl. Soft. Comput. 85 (2019) 105865.
[5] M.M. Özyetkin, H. Birdane, The processes with fractional order delay and PI controller design using particle swarm optimization, Int. J. Optim. Control: Theor. Appl. 13 (1) (2023) 81–91.
[6] S. Mirjalili, A. Lewis, The whale optimization algorithm, Adv. Eng. Softw. 95 (2016) 51–67.
[7] A.N. Mat, O. İnan, M. Karakoyun, An application of the whale optimization algorithm with Levy flight strategy for clustering of medical datasets, Int. J. Optim. Control: Theor. Appl. 11 (2) (2021) 216–226.
[8] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Softw. 69 (2014) 46–61.
[9] D. Karaboga, B. Basturk, On the performance of artificial bee colony (ABC) algorithm, Appl. Soft Comput. 8 (1) (2008) 687–697.
[10] D. Aydın, Y. Özcan, M. Sulaiman, G. Yavuz, Z. Halim, Elitist artificial bee colony with dynamic population size for multimodal optimization problems, Swarm Intell. 17 (4) (2023) 305–334.
[11] A. Kaveh, S. Talahatari, Charged system search for optimal design of frame structures, Appl. Soft Comput. 12 (1) (2012) 382–393.
[12] E. Duman, M. Uysal, A. Alkaya, Migrating Birds Optimization: A new metaheuristic approach and its performance on quadratic assignment problem, Inf. Sci. 217 (2012) 65–77.
[13] D. Goldberg, Genetic algorithms in search, optimization & machine learning, Addison Wesley Publishing Company, USA, 1989.
[14] A. Skakovski, P. Jędrzejowicz, An island-based differential evolution algorithm with the multi-size populations, Expert Syst. Appl. 126 (2019) 308–320.
[15] M. Rebaudengo, M. Reorda, An experimental analysis of the effects of migration in parallel genetic algorithms, in: 1993 Euromicro Workshop on Parallel and Distributed Processing, Gran Canaria, Spain, 1993, pp. 232–238.
[16] E. Alba, J. Troya, Improving flexibility and efficiency by adding parallelism to genetic algorithms, Stat. Comput. 12 (2) (2002) 91–114.
[17] T. Hong, L. Huang, W. Lin, Y. Liu, G. Chakraborty, Dynamic migration in multiple ant colonies, in: IEEE 2nd International Conference on Cybernetics (CYBCONF), Gdynia, Poland, 2015, 24-26 June.
[18] M.A. Al-Betar, M.A. Awadallah, A.T. Khader, Z.A. Abdalkareem, Island-based harmony search for optimization problems, Expert Syst. Appl. 42 (4) (2015) 2026–2035.
[19] Y. Maeda, M. Ishita, O. Li, Fuzzy adaptive search method for parallel genetic algorithm with island combination process, Int. J. Approx. Reason. 41 (1) (2006) 59–73.
[20] A. Abdelhafez, E. Alba, G. Luque, Performance analysis of synchronous and asynchronous distributed genetic algorithms on multiprocessors, Swarm Evol. Comput. 49 (2019) 147–157.
[21] N. Adar, G. Kuvat, Parallel genetic algorithms with dynamic topology using cluster computing, Adv. Electr. Comput. Eng. 16 (3) (2016) 73–80.
[22] M. Sulaiman, Z. Halim, M. Lebbah, M. Waqas, S. Tu, An evolutionary computing-based efficient hybrid task scheduling approach for heterogeneous computing environment, J. Grid Comput. 19 (1) (2021) 11.
[23] L. Asadzadeh, A parallel artificial bee colony algorithm for the job shop scheduling problem with a dynamic migration strategy, Comput. Ind. Eng. 102 (2016) 359–367.
[24] A. Baştürk, R. Akay, Performance analysis of the coarse-grained parallel model of the artificial bee colony algorithm, Inf. Sci. 253 (2013) 34–55.
[25] S. Aslan, Deployment in wireless sensor networks by parallel and cooperative parallel artificial bee colony algorithms, Int. J. Optim. Control: Theor. Appl. 9 (1) (2019) 1–10.
[26] A. Bevilaqua, D. Bevilaqua, K. Yamanaka, Parallel island based memetic algorithm with Lin–Kernighan local search for a real-life two-echelon heterogeneous vehicle routing problem based on Brazilian wholesale companies, Appl. Soft Comput. 76 (2019) 697–711.
[27] Z. Xuejun, G. Xiangmin, Z. Yanbo, L. Jiaxing, Strategic flight assignment approach based on multi-objective parallel evolution algorithm with dynamic migration interval, Chin. J. Aeronaut. 28 (2) (2015) 556–563.
[28] B.H. Abed-alguni, N.A. Alawad, Distributed Grey Wolf Optimizer for scheduling of workflow applications in cloud environments, Appl. Soft Comput. 102 (2021) 107113.
[29] M.S. Turgut, O.E. Turgut, D.T. Eliiyi, Island-based crow search algorithm for solving optimal control problems, Appl. Soft Comput. 90 (2020) 106170.
[30] D.R. Penas, J.R. Banga, P. González, R. Doallo, Enhanced parallel differential evolution algorithm for problems in computational systems biology, Appl. Soft Comput. 33 (2015) 86–99.
[31] T. Harada, E. Alba, Parallel genetic algorithms: a useful survey, ACM Comput. Surv. 53 (4) (2020) 39.
[32] P. Lissaman, C. Shollenberger, Formation flight of birds, Science (1970) 1003–1005.
[33] H. Makas, Parallel and collaborative applications of the recent optimization algorithms and their performance analyses, Doctoral Thesis, Sakarya University Institute of Natural Science, Computer and Information Engineering, 2015.
[34] A. Tülek, Implementation of migrating birds optimization algorithm on parallel computers, Msc Thesis, Balıkesir University Institute of Science, Electrical and Electronics Engineering, 2019.
[35] V. Tongur, E. Ülker, The analysis of migrating birds optimization algorithm with neighborhood operator on traveling salesman problem, Intell. Evol. Syst. 5 (2015) 227–237.
[36] S. Niroomand, A. Hadi-Vencheh, R. Şahin, B. Vizvári, Modified migrating birds optimization algorithm for closed loop layout with exact distances in flexible manufacturing systems, Expert Syst. Appl. 42 (19) (2015) 6586–6597.
[37] D. Öz, An improvement on the migrating birds optimization with a problem-specific neighboring function for the multi-objective task allocation problem, Expert Syst. Appl. 67 (2017) 304–311.
[38] H. Makas, N. Yumuşak, System identification by using migrating birds optimization algorithm: a comparative performance analysis, Turk. J. Electr. Eng. Comput. Sci. 24 (2016) 1879–1900.
[39] H. Makas, N. Yumuşak, Balancing exploration and exploitation by using sequential execution cooperation between artificial bee colony and migrating birds optimization algorithms, Turk. J. Electr. Eng. Comput. Sci. 24 (6) (2016) 4935–4956.

[40] T.D. Diep, K.T. Pham, K. Fürlinger, N. Thoai, A time-stamping system to detect memory consistency errors in MPI one-sided applications, Parallel Comput. 86 (2019) 36–44.
[41] N. Sultana, M. Rüfenacht, A. Skjellum, I. Laguna, K. Mohror, Failure recovery for bulk synchronous applications with MPI stages, Parallel Comput. 84 (2019) 1–14.
[42] H. Zhou, J. Gracia, N. Zhou, R. Schneider, Collectives in hybrid MPI+ MPI code: Design, practice and performance, Parallel Comput. 99 (2020) 102669.
[43] P.S. Pacheco, Parallel Programming with MPI, Morgan Kaufmann Publishers, San Francisco, California, 1997. An Imprint of Elsevier.
[44] V. Tongur, E. Ülker, PSO-based improved multi-flocks migrating birds optimization (IMFMBO) algorithm for solution of discrete problems, Soft Comput. 23 (14) (2019) 5469–5484.
[45] M. Hacibeyoglu, K. Alaykiran, A.M. Acilar, V. Tongur, E. Ulker, A comparative analysis of metaheuristic approaches for multidimensional two-way number partitioning problem, Arab. J. Sci. Eng. 43 (12) (2018) 7499–7520.
[46] G. Kuvat, N. Adar, The analysis of the interaction of the migration diversity and permeability in parallel genetic algorithms, Int. J. Optim. Control: Theor. Appl. (IJOCTA) 6 (1) (2016) 23–31, https://doi.org/10.11121/ijocta.01.2016.00262, 3366508.
[47] K.T. Perry, A critical examination of the use of preliminary tests in two-sample tests of location, J. Mod. Appl. Stat. Methods 2 (2) (2003) 5.
[48] D. Rasch, K.D. Kubinger, K. Moder, The two-sample t test: pre-testing its assumptions does not pay off, Stat. Pap. 52 (2011) 219–231.
[49] D. Maulud, A.M. Abdulazeez, A review on linear regression comprehensive in machine learning, J. Appl. Sci. Technol. Trends 1 (2) (2020) 140–147.
[50] T. Brambor, W.R. Clark, M. Golder, Understanding Interaction Models: Improving Empirical Analyses, Polit. Anal. 14 (1) (2006) 63–82.
[51] Y. Kim, H. Oh, Comparison between multiple regression analysis, polynomial regression analysis, and an artificial neural network for tensile strength prediction of BFRP and GFRP, Materials 14 (17) (2021) 4861.
[52] G. Di Franco, M. Santurro, Machine learning, artificial neural networks and social research, Qual. Quant. 55 (3) (2021) 1007–1025.
[53] S. Hosseini, A new machine learning method consisting of GA-LR and ANN for attack detection, Wirel. Netw. 26 (6) (2020) 4149–4162.
[54] F.E. Jalal, M. Iqbal, W.A. Khan, A. Jamal, K. Onyelowe, Lekhraj, ANN-based swarm intelligence for predicting expansive soil swell pressure and compression strength, Sci. Rep. 14 (1) (2024) 14597.
[55] R.K. Gupta, R.C. Singh, Optimizing high-speed rotating shaft vibration control: Experimental investigation of squeeze film dampers and a comparative analysis using artificial neural networks (ANN) and response surface methodology (RSM), Expert Syst. Appl. 249 (2024) 123800.
[56] L. Hasimi, D. Zavantis, E. Shakshuki, A. Yasar, Cloud computing security and deep learning: An ANN approach, Procedia Comput. Sci. 231 (2024) 40–47.
[57] D.R. Kumar, W. Wipulanusat, M. Kumar, S. Keawsawasvong, P. Samui, Optimized neural network-based state-of-the-art soft computing models for the bearing capacity of strip footings subjected to inclined loading, Intell. Syst. Appl. 21 (2023) 200314.
[58] R. El Chaal, M.O. Aboutafail, Comparing artificial neural networks with multiple linear regression for forecasting heavy metal content, Acadlore Trans. Geosci. 1 (1) (2022) 2–11, https://doi.org/10.56578/atg010102.
[59] J. Stangierski, D. Weiss, A. Kaczmarek, Multiple regression models and artificial neural network (ANN) as prediction tools of changes in overall quality during the storage of spreadable processed Gouda cheese, Eur. Food Res. Technol. 245 (11) (2019) 2539–2547.