Diversity-driven Cooperating Portfolio of Metaheuristic Algorithms

Adam Żychowski
adam.zychowski@pw.edu.pl
Faculty of Mathematics and Information Science,
Warsaw University of Technology
Warsaw, Poland

Xin Yao
xinyao@ln.edu.hk
School of Data Science,
Lingnan University
Hong Kong SAR

Jacek Mańdziuk
mandziuk@mini.pw.edu.pl
Faculty of Mathematics and Information Science,
Warsaw University of Technology
Warsaw, Poland
Faculty of Computer Science,
AGH University of Krakow
Krakow, Poland

Abstract

The paper introduces a novel hybrid island-based framework in which diverse metaheuristics cooperate to effectively explore the search space. A core component of the framework is a diversity-driven migration mechanism, enabling adaptive management of the information flow between islands. Three fundamental aspects of migration - *what* to migrate, *when* to migrate, and *where* to migrate - are thoroughly analyzed, leading to the development of strategies that foster synergy between heterogeneous algorithms. These strategies balance exploration and exploitation, ensuring effective global and local search. The framework was evaluated on a set of diverse optimization benchmarks, both discrete (Traveling Salesman Problem instances) and continuous (BBOB functions). Experimental results demonstrate that the proposed approach surpasses traditional algorithms and their island-based variants in convergence speed, solution quality, and resilience to stagnation. Adaptive mechanisms dynamically adjust migration strategies during the optimization process, further enhancing the framework's effectiveness. The proposed method represents an advancement in hybrid metaheuristic systems, offering scalability and flexibility that are essential for solving complex optimization tasks.

CCS Concepts

• **Computing methodologies → Search methodologies**; *Continuous space search*; *Discrete space search*.

Keywords

Metaheuristic, Island algorithm, Migration strategies

ACM Reference Format:
Adam Żychowski, Xin Yao, and Jacek Mańdziuk. 2025. Diversity-driven Cooperating Portfolio of Metaheuristic Algorithms. In *Proceedings of GECCO'25*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

GECCO'25, July 14–18, 2025, Málaga, Spain

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Optimization problems are at the core of many scientific and industrial challenges, including machine learning model tuning, supply chain logistics, network design, and engineering system optimization. These problems are often complex, featuring high dimensionality, multimodal landscapes, and conflicting objectives. Traditional optimization methods, such as gradient-based approaches, often fall short when dealing with non-linear, discontinuous, or noisy objective functions. To overcome these limitations, population-based metaheuristics have emerged as powerful alternatives for solving such problems.

Population-based metaheuristics, including Genetic Algorithms (GAs) [4], Particle Swarm Optimization (PSO) [16], Ant Colony Optimization (ACO) [5], and Differential Evolution (DE) [25], employ a population of candidate solutions to search the problem space. These algorithms are inspired by natural or social processes, such as biological evolution, swarm behavior, or pheromone-based communication, and operate by iteratively refining the population through various mechanisms. The diversity of solutions within the population plays a critical role in enabling these methods to effectively explore the search space and avoid premature convergence to suboptimal solutions. While these algorithms have proven effective across numerous applications, each has inherent strengths and weaknesses. For example, GAs excel at global exploration but may struggle with local exploitation, while PSO offers fast convergence but is prone to stagnation in multimodal landscapes.

One approach to address the limitations of individual population is the use of *island-based evolutionary algorithms*. In island-based models the entire population is divided into subpopulations, each evolving independently on separate "islands". These islands periodically exchange individuals through a migration mechanism, enabling the transfer of solutions between subpopulations. This structure not only helps maintain diversity across the entire population but also allows subpopulations to specialize in different regions of the search space. Migration strategies (such as selecting individuals to migrate based on fitness, diversity, or other criteria) are crucial for the effectiveness of the approach. Island-based models are particularly well-suited for parallelization and distributed computing, further enhancing their scalability and efficiency. Despite these advantages, most existing implementations of island-based methods focus on a single metaheuristic (usually evolutionary algorithms) [9], limiting the potential for exploiting the complementary strengths of different algorithms.

An important consideration in population-based and island-based models is the role of diversity in the optimization process. Diversity refers to the variety of solutions within the population and is critical for both global exploration and local exploitation. Insufficient diversity can lead to premature convergence, where the algorithm stagnates in suboptimal regions of the search space, while excessive diversity may slow down convergence by weakening the focus on promising solutions. Island-based models inherently promote diversity by isolating subpopulations. Different algorithms have distinct mechanisms for generating and maintaining diversity, and combining them in an island-based framework can enhance the global search process, reduce the risk of stagnation, and improve solution quality. However, this advantage can be further amplified by migrating certain individuals between islands to enhance diversity.

This paper aims to address the gap in existing research by proposing a novel hybrid island-based optimization framework that integrates multiple population-based metaheuristics. In the proposed framework, each island employs a distinct metaheuristic, such as GA, PSO, or ACO, to independently explore the search space. These islands cooperate through a migration mechanism driven by population diversity, ensuring that promising solutions are shared while maintaining a balance between exploration and exploitation. By leveraging the complementary strengths of different algorithms, the framework creates a cooperating portfolio of metaheuristics, enhancing the overall search process.

The main contributions of this work are as follows:

- *Hybrid island-based framework:* We introduce a novel framework that combines multiple metaheuristic algorithms in an island-based setting, enabling collaborative optimization.
- *Extensive experimental research on migration strategies:* We conduct an in-depth study exploring various migration strategies, driven by three core research questions: *What* to migrate, *When* to migrate, and *Where* to migrate. This investigation evaluates multiple migration variants to optimize collaboration between islands.
- *Diversity-driven migration mechanism:* A novel adaptive migration strategy based on population diversity is proposed to dynamically adapt the flow of individuals between islands, fostering effective cooperation.
- *Empirical evaluation:* We conduct extensive experiments on a suite of benchmark optimization problems, demonstrating the framework's superiority in terms of convergence speed, robustness, and solution quality compared to application of a single-algorithm and traditional hybrid approaches.

2 Related work

2.1 Island-based algorithms

Island-based models [17], also known as distributed or multi-population models, partition the global population into subpopulations (islands) that develop independently, occasionally exchanging individuals through migration.

Whitley et al. [32] provided one of the foundational studies by demonstrating how island models allow independent subpopulations to evolve in isolation before sharing solutions. This mechanism prevents premature convergence and supports diversity by avoiding the dominance of a single solution.

Skolicki and De Jong [30] and [29] examined the effects of migration sizes and intervals on island models, highlighting the required balance between exploration and exploitation to optimize performance. Their findings suggest that appropriate tuning of migration parameters can significantly improve the algorithm's efficiency.

Meng et al. [22] introduced a dynamic island model based on spectral clustering within GAs. Their approach adaptively organizes subpopulations, enhancing convergence rates and solution quality by aligning the island structure with the problem's landscape.

The island model framework has also been successfully applied to other heuristic methods, e.g. PSO [1], DE [2], ACO [23].

2.2 Diversity in population-based algorithms

Maintaining diversity within populations is crucial for the effectiveness of population-based algorithms, as it prevents premature convergence and ensures a comprehensive exploration of the search space [11]. The theoretical underpinnings of diversity in algorithm portfolios are reinforced by analysis of diversity in Genetic Programming [13].

Gustafson and Burke [12] presented the speciation island model, an alternative parallel Evolutionary Algorithm (EA) that promotes diversity through speciation. This model demonstrated improved performance in complex optimization tasks by maintaining diverse subpopulations. Similar approach was proposed in [15] with Hierarchical Fair Competition (HFC) model which prevents premature convergence through hierarchical subpopulations.

Gozali and Fujimura [10] introduced the DM-LIMGA algorithm, featuring a dual migration mechanism that preserves diversity by facilitating migration between both neighboring and distant islands. This approach effectively balanced exploration and exploitation, reducing the likelihood of premature convergence.

Araujo and Batista [3] developed a diversity-driven migration strategy for distributed EAs, where migration decisions are guided by diversity metrics to prevent the algorithm's stagnation.

2.3 Portfolios of metaheuristics

Algorithm portfolios combine multiple metaheuristic strategies to tackle complex optimization problems more effectively than individual algorithms [8]. By leveraging the complementary strengths of different methods, portfolios can adapt to diverse problem landscapes. The prior work was usually limited to different parameterizations of the same metaheuristic or creating algorithms portfolio which runs all instances in parallel (without communication).

Liu et al. [20] proposed an automatic construction of parallel portfolios through explicit instance grouping. Their approach tailored optimization resources to specific problem types, enhancing the overall effectiveness of metaheuristic ensembles.

Tang et al. [31] and Liu et al. [21] introduced a co-evolutionary adversarial framework for constructing algorithm portfolios, where algorithms are iteratively optimized in parallel. This method emphasized the importance of leveraging algorithm diversity to adaptively address the problem complexity.

Li and Gonsalves [18] proposed a parallel hybrid island metaheuristic algorithm, where different metaheuristics were assigned to individual islands. This approach leveraged the strengths of various optimization strategies, improving convergence speed and solution quality. Unlike Li and Gonsalves [18], which primarily focuses on hybridization of metaheuristics, our research emphasizes designing an optimal migration strategy to effectively manage population diversity across islands.

3 Proposed solution

3.1 Solution overview

In the proposed solution, we aim to combine the advantages of prior approaches and introduce a novel algorithm that leverages cooperating metaheuristics. The primary objective is to harness the synergy of diverse optimization methods to achieve stronger solutions and enhance convergence. Our approach is based on the well-established island model, but we contend that the use of different metaheuristics on each island necessitates a redefinition of the migration strategy. To address this issue, we evaluated various migration approaches and demonstrated that maintaining diversity plays a particularly critical role in this context.

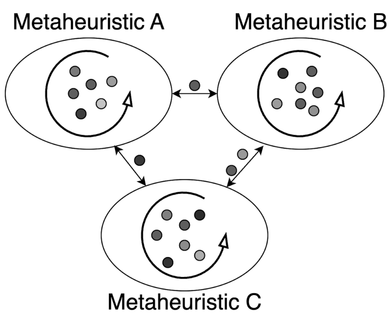

Figure 1: Overview of the proposed solution. Populations in multiple islands are developed by different metaheuristics with periodic migrations between islands.

Figure 1 provides an overview of the proposed method. In general, populations across multiple islands are developed using distinct metaheuristics, with periodic migrations occurring between islands. We consider a set of N islands, denoted as $\mathcal{I}$. Each island $I \in \mathcal{I}$ is characterized by three attributes: $I_{population}$ - representing a set of n_I candidate solutions (individuals); $I_{metaheuristic}$, the specific metaheuristic algorithm used to evolve $I_{population}$; and $I_{neighbours} \subset \mathcal{I}$, which defines the set of neighboring islands from which migration to I can occur. Algorithm 1 outlines the high-level pseudocode of the proposed method.

Initially, the population of each island is uniformly randomized. The evolution process is then carried out over a fixed number of fitness function evaluations, during which islands evolve their populations and occasionally exchange individuals through migration. Migration occurs under specific conditions (line 6): for each island, a decision on whether migration is necessary is made based on its population state and, potentially, the states of other populations. The precise conditions for migration and the definition of "population state" are elaborated in subsequent sections. After migration, the next generation of the population is produced using the metaheuristics associated with the island (line 11). This process updates the population state and consumes part of the evaluation budget (lines 12–13).

Algorithm 1 Pseudocode of the proposed method.

1: **for** each island $I \in \mathcal{I}$ **do**
2:     initialize $I_{population}$ with random individuals
3: **end for**
4: **while** *evaluation_budget > 0* **do**
5:     **for** each island $I \in \mathcal{I}$ **do**
6:         **if** needs_migration(I) **then**
7:             $I_{population} = I_{population} \cup$ migrate_from($I_{neighbours}$)
8:         **end if**
9:     **end for**
10:     **for** each island $I \in \mathcal{I}$ **do**
11:         $I_{population}$ = next_generation($I_{population}$, $I_{metaheuristic}$)
12:         evaluate($I_{population}$)
13:         *evaluation_budget = evaluation_budget* $- |I_{population}|$
14:     **end for**
15: **end while**
16: **return** the best individual from $\mathcal{I}$

3.2 Migration strategies

While the island-based concept of our method is inspired by existing works, we assert that the migration strategy is a pivotal component of the proposed island-based algorithm. Without migration, populations on different islands evolve independently; migration introduces the collaboration that defines the strength of island-based algorithms. Selecting an appropriate migration strategy becomes particularly critical in a heterogeneous setting, where different metaheuristics are applied on different islands, leading to potentially significant differences among populations.

To enhance the performance and adaptability of our proposed algorithm, we design specific strategies to answer three fundamental questions regarding migration: **What to migrate? When to migrate?** and **Where to migrate?** These aspects form a backbone of our migration mechanism and are carefully addressed to maintain diversity and cooperation among heterogeneous metaheuristics. We proposed and tested multiple strategies described below. Detailed results and further analysis are presented in Section 5.

3.2.1 When to migrate? Migration timing plays a crucial role in balancing exploration and exploitation. We propose several strategies to determine when migration should occur:

1) Periodic migration every y iterations: In this baseline strategy, migration is triggered at fixed intervals, regardless of the population's state. This is the most common approach in the literature.
2) Average fitness doesn't increase for x iterations: Migration is triggered when the average fitness of the population on an island remains stagnant for x consecutive iterations. This condition helps identify cases where the population converges prematurely to a local optimum.

3) Maximum fitness doesn't increase for x iterations: In this strategy, migration occurs if the best fitness value (maximum fitness) within an island's population does not improve for x iterations. This focuses on identifying stagnation in achieving higher-quality solutions.
4) Average and maximum fitness don't increase for x iterations: Combining the previous two criteria, migration is triggered when neither the average fitness nor the maximum fitness shows improvement over x iterations.
5) Diversity doesn't increase for x iterations: Diversity is a key indicator of a population's ability to explore the search space effectively. If diversity within an island's population does not improve for x iterations, migration is initiated to introduce new genetic material and avoid premature convergence.
6) Combination of 4 and 5: Migration occurs if either the average and maximum fitness fail to increase (as in 4) or diversity remains stagnant (as in 5). This hybrid strategy aims to address both the fitness stagnation and the lack of diversity.

3.2.2 What to migrate? Selecting the individuals for migration is critical to maintaining the balance between diversity and fitness. We propose the following strategies:

A) Individual with the best fitness: The individual with the highest fitness on the donor island is migrated. This strategy prioritizes exploitation by sharing the best solution found so far with the neighboring islands and it is the most popular approach in the existing literature.
B) Individual that increases diversity the most: The individual selected for migration is the one that contributes the greatest increase in diversity to the receiving island's population. This strategy prioritizes exploration by ensuring a more varied population.
C) Random individual that increases diversity: A randomly chosen individual from the donor island's population is selected, provided it contributes to increasing diversity on the receiving island. This introduces stochasticity while maintaining a focus on diversity.
D) Combination of diversity increment and fitness: The migrated individual is chosen based on a sum of its contribution to diversity and fitness, with both metrics normalized to a [0,1] range. This approach seeks to balance exploration and exploitation, leveraging the strengths of both.

3.2.3 Where to migrate? The topology of migration determines the structure of interaction between islands. We explore 4 primary strategies (visualized in Figure 2):

Ring – in this configuration, each island exchanges individuals only with its immediate neighbors in a predefined *Ring* structure. This localized interaction fosters gradual propagation of solutions across the system.

Clique - each island is directly connected to every other island. Individuals can migrate between any pair of islands, enabling a high level of interaction. This topology promotes faster propagation of high-quality solutions and maintains diversity by allowing individuals to exchange across the network.

Cycle - unlike the *Ring* topology, which allows bidirectional migration, the *Cycle* topology restricts the flow to one direction – each island sends individuals to the next island in a predefined cyclic order.

Star - a central hub island is designated to coordinate migrations. All other islands send individuals to the central hub, which redistributes them to other islands.

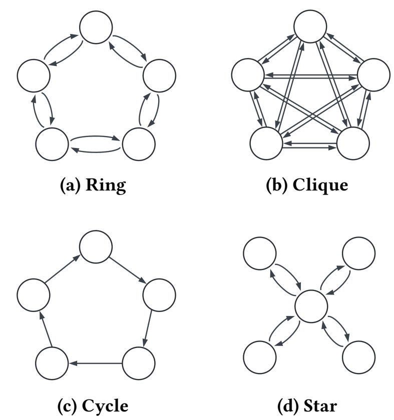

Figure 2: Tested island migration topologies.

By addressing the fundamental questions of What to migrate? When to migrate? and Where to migrate?, we aim to test various migration strategies ensuring effective cooperation and diversity management across heterogeneous metaheuristic algorithms. The interplay between these strategies is pivotal to achieving robust optimization performance. Detailed empirical evaluations of these strategies are presented in subsequent sections.

4 Experimental setup

4.1 Tested problems

We evaluated the proposed solution using two well-established benchmark problem domains: Traveling Salesman Problem (TSP) and Black-box Optimization Benchmarking (BBOB) framework.

For the TSP, we randomly selected 10 instances from the widely-used TSPLIB library [27]. These instances consist of between 400 and 724 nodes (cities), specifically: *rd400, fl417, pcb442, d493, att532, si535, u574, p654, d657*, and *u724*.

For BBOB, we utilized 12 problem instances from the COCO platform [14] dataset: *f2, f4, f6, f8, f10, f12, f14, f16, f18, f20, f22*, and *f24* [7]. These functions include all even-numbered functions, ensuring a diverse distribution across different groups characterized by varying properties. Number of dimensions for all functions was set to 20.

The TSP represents a discrete optimization problem, while BBOB consists of continuous optimization tasks. This distinction makes them fundamentally different in nature, providing a comprehensive testbed for evaluating the performance of our solution and its variants across diverse problem types.

4.2 Algorithm setup

In our experiments, we employed three distinct and widely recognized heuristic algorithms: PSO, GA, and ACO. These algorithms were chosen due to their fundamentally different properties and mechanisms of operation, ensuring diversity in the optimization approaches considered. It is important to note that the selection of specific metaheuristics is not the primary focus of this study. The main objective of this work is to introduce a general framework and evaluate the performance of various diversity-driven migration strategies. The choice of particular heuristic algorithms could be explored in greater depth in future research. At this stage, our intention was to ensure that the considered algorithms are distinct in nature to demonstrate the applicability of the proposed framework.

We utilized standard implementations of the selected algorithms, previously proposed in the literature. Specifically, for continuous optimization, we employed the following implementations: ACO [19], GA [24], and PSO [6]. For the TSP, we used ACO [33], GA [26], and PSO [28].

For our initial experiments, the number of islands, N, was set to 6, with 2 islands allocated to each of the three metaheuristic algorithms mentioned above. The population size for each island was fixed at $n_I = 100$. Details regarding the algorithm parameterization can be found in the supplementary material.

All experiments were conducted independently 20 times using different random seeds. The results presented in the following section represent averages over all runs and 10 benchmark instances, resulting in a total of $20 \times 10 = 200$ outcomes. Statistical significance was checked according to the Wilcoxon signed-rank test with p-value ≤ 0.05.

To ensure a fair comparison, we set a uniform fitness function evaluation budget of 10^5 evaluations for all algorithms. Wall clock time was not reported, as the differences in execution times between the tested algorithm variants and parameter settings were not significant.

Following the approach outlined in [3], we calculate the diversity metrics as the average (over the problem dimensions) of the standard deviations computed for each dimension of the solution encoding vectors. Specifically, given a population $I_{population} = \{i_1, i_2, \ldots, i_{n_I}\}$ and a solution $\mathbf{x}^{i_k} = [x_1^{i_k}, x_2^{i_k}, \ldots, x_D^{i_k}]$ associated with individual i_k, the diversity is defined as:

$$\frac{1}{D} \sum_{d=1}^{D} \sigma(\{x_d^{i_1}, x_d^{i_2}, \ldots, x_d^{i_{n_I}}\}), \quad (1)$$

where $\sigma(\{x_d^{i_1}, x_d^{i_2}, \ldots, x_d^{i_{n_I}}\})$ represents the standard deviation of the values in dimension d across the population. This metric provides a measure of the spread or variability of solutions in each problem dimension, averaged over all dimensions D.

5 Results

This section presents the experimental evaluation of the proposed framework, highlighting the impact of migration strategies, hybridization of metaheuristics, and adaptive mechanisms on optimization performance across benchmark problems.

5.1 Migration - *When, What,* and *Where*

Firstly, we investigated various migration strategies as described in Section 3. Since the three key aspects of migration—when to migrate, what to migrate, and where to migrate—can influence each other, it is challenging to isolate their effects by holding two of them constant while varying the third one. To address this interdependence, we tested all possible combinations of these strategies, resulting in a Cartesian product of their configurations. This approach produced 96 unique combinations ($6 \times 4 \times 4$). For each combination, every test problem instance was executed 20 times to ensure statistical robustness. Rankings of the combinations were then determined for each problem instance, based on the average performance across these 20 runs. The frequency of periodic migrations (y) in the first "When to migrate" strategy was set to 20. For all other strategies, the condition-checking period (x) was set to 10, based on insights gained from preliminary experiments (details are provided in the supplementary material.

The best-performing combination, with the lowest average ranking across both TSP and BBOB problem sets, was (6, D, Clique) with the average ranking equal to 2.6 and 4.4, respectively. This corresponds to the sixth "When to migrate" strategy (migration occurs when either the average and maximum fitness fail to improve or population diversity stagnates), the "D" strategy for "What to migrate" (migrating the individual that maximizes the sum of normalized diversity increment and fitness), and the *Clique* topology for "Where to migrate." The next best combinations were (6, D, Ring), (4, D, Clique), and (4, D, Ring).

At the opposite end of the spectrum, the weakest combinations were (5, B, Cycle), (5, B, Star), and (2, B, Star). Due to space constraints, Tables 1, 2, and 3 only present the average results for each "When," "What," and "Where" strategy independently. Comprehensive results for all 96 combinations can be found in the supplementary material.

Table 1: Average rank and fitness values for different "When to migrate" strategies (see Sec. 3).

When variant	TSP		BBOB	
	Avg rank	Avg fitness	Avg rank	Avg fitness
1	15.9	35571	21.3	1.609
2	49.2	35605	46.8	1.635
3	11.6	35570	18.1	1.586
4	18.0	35579	20.8	**1.560**
5	29.4	35588	32.9	1.593
6	**7.2**	**35559**	**12.1**	1.570

5.1.1 *When to migrate?*

The results indicate that the best migration performance was achieved with the sixth strategy, which combines conditions for migration based on both fitness and diversity. This suggests that leveraging multiple stagnation criteria provides a more reliable indicator for migration, likely due to its ability to capture both convergence and premature stagnation phenomena. The next best strategies involved migration based on stagnation of maximum fitness (strategies 3 and 4), followed by the classical fixed-period migration (strategy 1).

In contrast, the poorest results were associated with migration criteria based solely on diversity stagnation (strategy 5) or average population fitness (strategy 2). These metrics are potentially less stable and more susceptible to noise introduced by the stochastic components of metaheuristics, such as mutation. For instance, mutation can cause fluctuations in average fitness across generations, reducing the reliability of this metric as a migration trigger.

Table 2: Average rank and fitness values for different "What to migrate" strategies (see Sec. 3).

What variant	TSP		BBOB	
	Avg rank	Avg fitness	Avg rank	Avg fitness
A	15.4	35580	41.3	1.609
B	80.2	35623	59.9	1.672
C	58.6	35610	54.8	1.646
D	**12.9**	**35571**	**25.3**	**1.584**

5.1.2 What to migrate? The choice of "What to migrate" significantly impacted the results. The best-performing strategy was to migrate the individual that maximizes the sum of normalized diversity increment and fitness (strategy D). This approach likely balances exploration (through diversity enhancement) and exploitation (through fitness maximization), leading to more effective migration. The classical strategy of migrating the fittest individual (strategy A) performed worse, as it prioritizes exploitation but may overlook the need for maintaining diversity.

Strategies focusing solely on diversity (B and C) yielded significantly poorer outcomes. These strategies appear to neglect the fitness dimension, which is crucial for guiding the search process, and may result in the migration of less useful individuals that do not contribute effectively to problem-solving.

Table 3: Average rank and fitness values for different "Where to migrate" strategies (see Sec. 3).

Where variant	TSP		BBOB	
	Avg rank	Avg fitness	Avg rank	Avg fitness
Ring	**12.9**	**35571**	**22.4**	1.609
Clique	26.4	35588	31.3	**1.605**
Cycle	46.2	35605	49.3	1.649
Star	40.9	35600	41.6	1.649

5.1.3 Where to migrate? The topology of migration also played a critical role. The *Ring* and *Clique* topologies consistently produced superior results. The *Ring* topology likely benefits from its structured but limited communication, which allows diversity to propagate without overwhelming convergence. The *Clique* topology, on the other hand, enables direct interaction between all subpopulations, facilitating rapid exchange of high-quality individuals.

In contrast, the *Star* and *Cycle* topologies underperformed. The *Star* topology's centralization may lead to bottlenecks, as all migration depends on a single central population. Similarly, the *Cycle* topology's rigid structure limits the flow of information and diversity, reducing its effectiveness compared to the more dynamic *Ring* and *Clique* configurations.

Statistical analysis revealed that the differences between all strategy pairs of "When to migrate", as well as "What to migrate" were statistically significant. However, no statistically significant differences were observed between the *Ring* and *Clique* topologies and between the *Cycle* and *Star* topologies, though *Ring* and *Clique* results were statistically significantly better than *Cycle* and *Star*.

Given that the combination (6, D, Clique) demonstrated the best performance, this configuration is adopted for subsequent experiments. The proposed method with this configuration is henceforth referred to as the **Diversity-driven Cooperating Portfolio of Metaheuristics** (**DdCPM**) algorithm.

5.2 Hybridization

We evaluated the effectiveness of our hybridization approach, which combines various metaheuristics, by comparing the performance of the DdCPM algorithm with the following baselines:

(1) Pure ACO, GA, and PSO (using the same metaheuristic implementations that were employed in the islands of DdCPM) with a population size equal to the sum of all the island populations in DdCPM.
(2) Island-based variants of these pure algorithms. In these variants, the configurations were identical to those used in DdCPM, including the same number of islands, population sizes, and migration strategies. However, all islands utilized the same algorithm (either ACO, GA, or PSO) rather than the heterogeneous approach.
(3) Island-based setup with heterogeneous algorithms, with two islands per each of the ACO, GA, and PSO algorithms, like in DdCMP setting, but without migration mechanism.

Detailed results, averaged over 20 independent runs, for each of the TSP and BBOB problems are presented in Table 4. Rows 1-3 refer to a single-population setup, rows 4-6 to homogeneous islands setup, row 7 to heterogeneous setup without migration, and the last row refers to our method.

The results confirm that DdCPM outperforms all competing methods across all instances. In 21 out of 22 problem instances (all except the easiest f2 BBOB function), the differences between DdCPM and the remaining algorithms are statistically significant. This highlights the effectiveness of the hybridization approach, which combines diverse metaheuristics to balance exploration and exploitation more effectively than any single constituent algorithm or its uniform island-based variant. The superior performance of DdCPM demonstrates its ability to leverage the strengths of diverse algorithms while mitigating their individual weaknesses. These findings support the validity and practical advantages of the proposed hybridization strategy.

5.3 Diversity vs fitness

The results presented in the previous section underscore the importance of maintaining a balance between fitness and diversity, particularly when selecting the individual to be migrated. The strategy that combined diversity improvement and fitness achieved significantly better performance. The metric m, used to determine the individual for migration, was calculated as the sum of the normalized diversity increment and fitness, both scaled to the [0,1] interval. This ensured that both factors contributed equally to the decision-making process.

However, this may not represent the optimal strategy, and it is possible that one of these components should be given more weight than the other. To investigate this, we propose using a weighted

Table 4: Comparison of hybridized metaheuristics (DdCPM) with individual methods for TSP (top) and BBOB (bottom) instances.

TSP instance	rd400	fl417	pcb442	d493	att532	si535	u574	p654	d657	u724
ACO	15562 ±16	12094 ±11	51455 ±97	35487 ±67	28102 ±49	49113 ±81	37477 ±53	35141 ±41	49571 ±87	42495 ±47
GA	15531 ±54	12095 ±33	51426 ±161	35441 ±125	28112 ±84	49098 ±155	37360 ±141	35114 ±121	49541 ±110	42472 ±136
PSO	15561 ±27	12108 ±14	51416 ±90	35518 ±58	28101 ±27	49068 ±95	37412 ±53	35164 ±63	49560 ±93	42436 ±62
ACO islands	15522 ±23	12070 ±25	51419 ±106	35489 ±46	28095 ±47	49080 ±71	37377 ±54	35103 ±69	49578 ±86	42483 ±92
GA islands	15480 ±48	12039 ±31	51332 ±126	35418 ±83	28040 ±61	49013 ±134	37373 ±94	35030 ±80	49498 ±180	42380 ±91
PSO islands	15527 ±17	12069 ±16	51391 ±106	35498 ±48	28064 ±48	49078 ±55	37431 ±52	35130 ±68	49531 ±57	42506 ±107
ACO+GA+PSO	15576 ±53	12112 ±31	51488 ±158	35497 ±122	28147 ±83	49123 ±153	37418 ±137	35162 ±118	49592 ±109	42485 ±135
DdCPM	**15448** ±23	**11996** ±27	**51294** ±107	**35365** ±38	**27987** ±29	**48984** ±69	**37305** ±87	**35012** ±37	**49418** ±62	**42366** ±52

BBOB instance	f2	f4	f6	f8	f10	f12	f14	f16	f18	f20	f22	f24
ACO	0.015 ±.000	1.558 ±.020	1.089 ±.010	2.228 ±.023	3.708 ±.067	5.046 ±.050	0.707 ±.008	0.859 ±.015	0.843 ±.010	1.613 ±.021	0.480 ±.006	1.492 ±.027
GA	0.015 ±.000	1.491 ±.053	1.050 ±.035	2.092 ±.067	3.571 ±.103	4.930 ±.181	0.739 ±.025	0.881 ±.026	0.790 ±.017	1.575 ±.038	0.466 ±.015	1.561 ±.052
PSO	0.015 ±.000	1.579 ±.024	1.060 ±.018	2.201 ±.042	3.447 ±.066	5.102 ±.071	0.720 ±.008	0.889 ±.015	0.833 ±.010	1.574 ±.024	0.466 ±.006	1.511 ±.019
ACO islands	0.015 ±.000	1.466 ±.019	1.008 ±.022	2.132 ±.040	3.466 ±.038	4.706 ±.088	0.699 ±.013	0.872 ±.018	0.817 ±.010	1.573 ±.029	0.487 ±.008	1.574 ±.028
GA islands	**0.014** ±.000	1.457 ±.050	1.041 ±.028	2.058 ±.061	3.546 ±.124	4.993 ±.122	0.683 ±.018	0.859 ±.029	0.781 ±.026	1.514 ±.047	0.468 ±.018	1.488 ±.039
PSO islands	0.015 ±.000	1.528 ±.040	1.003 ±.018	2.017 ±.050	3.669 ±.071	5.007 ±.117	0.690 ±.010	0.887 ±.015	0.817 ±.012	1.509 ±.019	0.481 ±.011	1.492 ±.016
ACO+GA+PSO	0.016 ±.000	1.495 ±.051	1.053 ±.034	2.098 ±.065	3.580 ±.101	4.945 ±.178	0.741 ±.024	0.885 ±.025	0.792 ±.016	1.580 ±.037	0.468 ±.014	1.565 ±.050
DdCPM	**0.014** ±.000	**1.411** ±.023	**0.927** ±.011	**1.886** ±.036	**3.229** ±.057	**4.639** ±.108	**0.653** ±.009	**0.818** ±.011	**0.777** ±.016	**1.450** ±.024	**0.444** ±.009	**1.421** ±.026

sum of the two components. Specifically, we define the metric as:

$$m = \alpha\xi + (1 - \alpha)\psi, \tag{2}$$

where ξ represents the normalized diversity increment (as defined in Eq. 1), and ψ is the normalized fitness value. α serves as a steering parameter that controls the balance between diversity and fitness in the selection process for migration. Table 5 presents the results of the DDMS algorithm for various values of α.

Table 5: Averaged results for the TSP and BBOB problems, showing the impact of varying the balance parameter (α) between diversity and fitness in the selection of individuals for migration (see Eq. 2). $\alpha = 0.5$ refers to the results presented in Table 4.

α	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
TSP	35558	35360	**35335**	35410	35397	35459	35594	35618	35680	35705	35742
BBOB	1.668	**1.432**	1.437	1.456	1.442	1.484	1.581	1.614	1.676	1.733	1.789

The results presented in Table 5 reveal that the optimal balance between diversity and fitness for the migration process is achieved with $\alpha = 0.2$ for the TSP and $\alpha = 0.1$ for the BBOB problems. Compared to the case of $\alpha = 0.0$ it can be seen that even a small introduction of diversity into the migration process significantly improves the algorithm's performance. This finding highlights that, while fitness remains crucial for guiding the search towards optimal solutions, introducing diversity helps maintain a broader search space, reducing the risk of premature convergence and improving the algorithm's robustness.

5.4 Adaptive migrations

Following the conclusion regarding the critical role of a careful balance between diversity and fitness in the migration process, in this section, we propose three variants of adaptive migration strategies. All of them dynamically adjust the balance between diversity and fitness during the algorithm's execution by a suitable adaptation of parameter α. The general intuition behind the need for adaptation is to allow the algorithm to dynamically shift its focus between exploitation and exploration throughout the search process. At the start of the search, the population is typically diverse, and the algorithm benefits from focusing on fitness to quickly converge towards promising regions of the solution space. This is why, at the beginning, we use smaller values of α to prioritize fitness during migration. However, as the algorithm progresses and the population becomes more homogeneous, there is a risk of stagnation. In such cases, maintaining diversity becomes increasingly important to prevent the algorithm from getting trapped in a local optima. This is why, later in the search, α is gradually increased, strengthening the influence of diversity in the selection process for migration.

We propose the following three adaptive migration strategies:

a) Linearly decreasing α with iterations

In the first variant, we set α to decrease linearly over time, according to the following formula:

$$\alpha = \frac{\text{current iteration}}{\text{total number of iterations}}.$$

The total number of iterations can be easily computed based on the number of islands, number of individuals in each island, and the fitness function evaluation budget.

In the early stages, the algorithm focuses on improving fitness to quickly converge towards promising regions of the search space. As the number of iterations increases, α decreases, shifting the balance toward diversity and encouraging exploration when the population is at risk of stagnating. This approach ensures that the migration process initially promotes exploitation, but over time, the algorithm becomes more exploratory, allowing it to avoid local optima.

b) Fitness stagnation-based adaptation

The second variant adapts α based on the fitness stagnation in the population. Starting from $\alpha = 0$, α is increased if no improvement is observed for a certain number of iterations. Specifically, the following formula is applied:

$$\alpha = \alpha + \delta \quad \text{if no improvement occurs for } x_\delta \text{ iterations,}$$

where δ and x_δ are parameters controlling the rate of adaptation. δ determines the rate at which α increases when stagnation is detected, and x_{delta} controls how long the population must stagnate before α being adapted. In the experiments reported in the paper, we set $\delta = 0.05$ and $x = 10$. The results for other selections of δ and

Table 6: Comparison of different migration adaptive strategies for TSP (top) and BBOB (bottom) instances.

TSP instance	rd400	fl417	pcb442	d493	att532	si535	u574	p654	d657	u724
$\alpha = 0.2$	15408 ±18	11945 ±12	51159 ±57	35234 ±39	27845 ±32	48894 ±97	37244 ±65	34827 ±38	49285 ±49	42159 ±55
iterative decay	**15319** ±27	**11865** ±12	**50813** ±68	**35083** ±46	27739 ±51	48665 ±50	**37077** ±52	**34684** ±37	49138 ±59	**42047** ±50
stagnation based	15359 ±21	11908 ±21	50872 ±51	35000 ±38	27746 ±51	48738 ±80	37146 ±49	34724 ±54	**48991** ±50	41872 ±74
fitness based	15368 ±19	11875 ±18	50920 ±75	35103 ±40	**27669** ±34	**48600** ±90	37041 ±57	34612 ±48	49132 ±78	41948 ±60

BBOB instance	f2	f4	f6	f8	f10	f12	f14	f16	f18	f20	f22	f24
$\alpha = 0.1$	0.014 ±.000	1.392 ±.010	0.904 ±.005	1.857 ±.011	3.183 ±.045	4.459 ±.045	0.633 ±.005	0.801 ±.006	0.747 ±.007	1.428 ±.015	0.439 ±.005	1.404 ±.009
iterative decay	**0.013** ±.000	1.350 ±.012	**0.855** ±.007	**1.728** ±.018	3.053 ±.020	4.359 ±.057	0.608 ±.004	0.771 ±.011	**0.702** ±.007	1.377 ±.017	0.428 ±.004	**1.313** ±.014
stagnation based	**0.013** ±.000	1.349 ±.017	0.870 ±.009	1.789 ±.020	3.113 ±.022	**4.322** ±.063	**0.599** ±.004	0.773 ±.008	0.706 ±.008	1.340 ±.011	0.425 ±.003	1.316 ±.008
fitness based	**0.013** ±.000	**1.303** ±.008	0.885 ±.010	1.788 ±.024	**3.022** ±.032	4.350 ±.038	0.609 ±.008	**0.755** ±.005	0.722 ±.005	**1.331** ±.019	**0.415** ±.002	1.317 ±.017

x_δ are presented in the supplementary material. On a general note, the results were found to be relatively insensitive to the choice of these parameters, meaning the strategy is robust to reasonable variations in their values.

This strategy aims to dynamically adjust the migration process based on the population's behavior. When the population's fitness stagnates over a given number of iterations, it signals that the search is becoming less effective, and the population may be converging prematurely. Increasing α enhances the role of diversity in the migration process, helping to diversify the solutions and encourage the exploration of new regions in the solution space.

c) Fitness ratio-based adaptation

The third variant adapts α based on the relative fitness of the population within the island and among neighboring islands. Specifically, α is calculated as follows:

$$\alpha = \frac{\max \text{fitness}_{\text{local}}}{\max \text{fitness}_{\text{global}}},$$

where max $\text{fitness}_{\text{local}}$ refers to the maximum fitness value within the population of the island to which an individual is to be migrated, and max $\text{fitness}_{\text{global}}$ refers to the maximum fitness among individuals in neighboring islands from which migration can occur.

This strategy adjusts the migration parameter based on how the island population's fitness compares to the global fitness across neighboring islands. If the local population has a relatively high fitness compared to the global population, it suggests that the island is well-converged, and migration should place more emphasis on diversity to avoid stagnation. On the other hand, if the local population fitness is low compared to the global fitness, it implies that the island needs to focus more on fitness improvement, and migration should prioritize individuals that strengthen the fitness.

Table 6 present results for the adaptation methods proposed above. For the reference, we provide results for a fixed α value (the best selection from Table 5).

The results clearly demonstrate the superior performance of the three adaptive migration methods over the fixed value of α across all problem instances (in 61 out of 66 cases the differences are statistically significant). However, while all three adaptive methods show improved performance, the differences between them are statistically insignificant. This suggests that the key factor in achieving superior performance lies in the general ability to adjust α in response to the current state of the solution process, rather than the specific adaptation mechanism employed.

6 Conclusions

In this paper, we propose a novel hybrid optimization framework in which diverse metaheuristics cooperate to effectively explore the search space in the island-based setup. The proposed framework successfully integrates multiple metaheuristic algorithms within an island-based architecture, leveraging diversity-driven migration strategies to enhance optimization process. By addressing the questions of *when*, *what*, and *where* to migrate, the framework achieves a balance between exploration and exploitation, significantly improving solution quality across diverse optimization problems.

Extensive experiments on 22 instances of two popular problems: TSP (discrete optimization) and BBOB (continuous optimization) demonstrate the efficacy of DdCPM in outperforming traditional single-algorithm approaches, as well as conventional island-based implementations.

It is important to highlight that our focus is not on competing with the state-of-the-art optimization methods that are highly specialized and tailored to specific problem instances. Instead, our aim is to propose a wide perspective of a general framework that integrates various methods and demonstrate how this integration enhances the performance. The performed experiments indicate that the use of a hybrid portfolio of metaheuristics allows the framework to exploit the complementary strengths of different algorithms, leading to superior convergence and robustness. The introduction of adaptive migration strategies further enhances performance by dynamically adjusting the balance between diversity and fitness throughout the search process.

The findings highlight the importance of diversity as a cornerstone of robust optimization, providing critical insights for future research in hybrid metaheuristic frameworks. The scalability and flexibility of DdCPM make it a promising tool for solving complex optimization problems in both discrete and continuous domains.

Future work will focus on extending the framework to incorporate additional metaheuristics and problem types. We believe that the continued development of hybrid metaheuristic frameworks holds significant potential for advancing the state-of-the-art in optimization research.

Acknowledgments

Adam Żychowski was funded by the Warsaw University of Technology within the Excellence Initiative: Research University (IDUB) program. Xin Yao's work is partially supported by an internal seed grant from Lingnan University. Jacek Mańdziuk's work was partially supported by the National Science Centre, Poland, grant number 2023/49/B/ST6/01404.

References

[1] Houda Abadlia, Nadia Smairi, and Khaled Ghedira. 2017. Particle swarm optimization based on dynamic island model. In *2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, 709–716.

[2] Javier Apolloni, Guillermo Leguizamón, José García-Nieto, and Enrique Alba. 2008. Island based distributed differential evolution: an experimental study on hybrid testbeds. In *2008 eighth international conference on hybrid intelligent systems*. IEEE, 696–701.

[3] Jean NR Araujo and Lucas S Batista. 2023. A diversity-driven migration strategy for distributed evolutionary algorithms. *Swarm and Evolutionary Computation* 82 (2023), 101361.

[4] Thomas Bäck and Hans-Paul Schwefel. 1993. An overview of evolutionary algorithms for parameter optimization. *Evolutionary computation* 1, 1 (1993), 1–23.

[5] Marco Dorigo, Mauro Birattari, and Thomas Stutzle. 2006. Ant colony optimization. *IEEE computational intelligence magazine* 1, 4 (2006), 28–39.

[6] Mohammed El-Abd and Mohamed S Kamel. 2009. Black-box optimization benchmarking for noiseless function testbed using particle swarm optimization. In *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers*. 2269–2274.

[7] Steffen Finck, Nikolaus Hansen, Raymond Ros, and Anne Auger. 2010. *Real-parameter black-box optimization benchmarking 2009: Presentation of the noiseless functions*. Technical Report. Citeseer.

[8] Carla P Gomes and Bart Selman. 2001. Algorithm portfolios. *Artificial Intelligence* 126, 1-2 (2001), 43–62.

[9] Yue-Jiao Gong, Wei-Neng Chen, Zhi-Hui Zhan, Jun Zhang, Yun Li, Qingfu Zhang, and Jing-Jing Li. 2015. Distributed evolutionary algorithms and their models: A survey of the state-of-the-art. *Applied Soft Computing* 34 (2015), 286–300.

[10] Alfian Akbar Gozali and Shigeru Fujimura. 2019. DM-LIMGA: dual migration localized island model genetic algorithm—a better diversity preserver island model. *Evolutionary Intelligence* 12, 4 (2019), 527–539.

[11] Deepti Gupta and Shabina Ghafir. 2012. An overview of methods maintaining diversity in genetic algorithms. *International journal of emerging technology and advanced engineering* 2, 5 (2012), 56–60.

[12] Steven Gustafson and Edmund K Burke. 2006. The speciating island model: An alternative parallel evolutionary algorithm. *J. Parallel and Distrib. Comput.* 66, 8 (2006), 1025–1036.

[13] Steven Matt Gustafson. 2004. *An analysis of diversity in genetic programming*. Ph. D. Dissertation. University of Nottingham.

[14] Nikolaus Hansen, Anne Auger, Raymond Ros, Olaf Mersmann, Tea Tušar, and Dimo Brockhoff. 2021. COCO: A platform for comparing continuous optimizers in a black-box setting. *Optimization Methods and Software* 36, 1 (2021), 114–144.

[15] Jianjun Hu, Erik Goodman, Kisung Seo, Zhun Fan, and Rondal Rosenberg. 2005. The hierarchical fair competition (HFC) framework for sustainable evolutionary algorithms. *Evolutionary Computation* 13, 2 (2005), 241–277.

[16] James Kennedy and Russell Eberhart. 1995. Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks*, Vol. 4. ieee, 1942–1948.

[17] António Leitão, Francisco Baptista Pereira, and Penousal Machado. 2015. Island models for cluster geometry optimization: how design options impact effectiveness and diversity. *Journal of Global Optimization* 63 (2015), 677–707.

[18] Jiawei Li and Tad Gonsalves. 2022. Parallel hybrid island metaheuristic algorithm. *IEEE Access* 10 (2022), 42268–42286.

[19] Tianjun Liao, Daniel Molina, Thomas Stutzle, Marco A Montes de Oca, and Marco Dorigo. 2012. An ACO algorithm benchmarked on the BBOB noiseless function testbed. In *Proceedings of the 14th annual conference companion on Genetic and evolutionary computation*. 159–166.

[20] Shengcai Liu, Ke Tang, and Xin Yao. 2019. Automatic construction of parallel portfolios via explicit instance grouping. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 1560–1567.

[21] Shengcai Liu, Ke Tang, and Xin Yao. 2020. Generative adversarial construction of parallel portfolios. *IEEE transactions on cybernetics* 52, 2 (2020), 784–795.

[22] Qinxue Meng, Jia Wu, John Ellis, and Paul J Kennedy. 2017. Dynamic island model based on spectral clustering in genetic algorithm. In *2017 international joint conference on neural networks (IJCNN)*. IEEE, 1724–1731.

[23] Antonio Miguel Mora, Pablo García-Sánchez, JJ Merelo, and Pedro A Castillo. 2013. Pareto-based multi-colony multi-objective ant colony optimization algorithms: an island model proposal. *Soft Computing* 17 (2013), 1175–1207.

[24] Miguel Nicolau. 2009. Application of a simple binary genetic algorithm to a noiseless testbed benchmark. In *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers*. 2473–2478.

[25] Kenneth V Price. 2013. Differential evolution. In *Handbook of optimization: From classical to modern approach*. Springer, 187–214.

[26] Noraini Mohd Razali, John Geraghty, et al. 2011. Genetic algorithm performance with different selection strategies in solving TSP. In *Proceedings of the world congress on engineering*, Vol. 2. International Association of Engineers Hong Kong, China, 1–6.

[27] Gerhard Reinelt. 1991. TSPLIB—A traveling salesman problem library. *ORSA journal on computing* 3, 4 (1991), 376–384.

[28] Xiaohu H Shi, Yanchun Chun Liang, Heow Pueh Lee, C Lu, and QX Wang. 2007. Particle swarm optimization-based algorithms for TSP and generalized TSP. *Information processing letters* 103, 5 (2007), 169–176.

[29] Zbigniew Skolicki. 2005. An analysis of island models in evolutionary computation. In *Proceedings of the 7th annual workshop on Genetic and evolutionary computation*. 386–389.

[30] Zbigniew Skolicki and Kenneth De Jong. 2005. The influence of migration sizes and intervals on island models. In *Proceedings of the 7th annual conference on Genetic and evolutionary computation*. 1295–1302.

[31] Ke Tang, Shengcai Liu, Peng Yang, and Xin Yao. 2021. Few-shots parallel algorithm portfolio construction via co-evolution. *IEEE Transactions on Evolutionary Computation* 25, 3 (2021), 595–607.

[32] Darrell Whitley, Soraya Rana, and Robert B Heckendorn. 1999. The island model genetic algorithm: On separability, population size and convergence. *Journal of computing and information technology* 7, 1 (1999), 33–47.

[33] Jinhui Yang, Xiaohu Shi, Maurizio Marchese, and Yanchun Liang. 2008. An ant colony optimization method for generalized TSP problem. *Progress in natural science* 18, 11 (2008), 1417–1422.