Migration Timing in Hybrid Island-Based Metaheuristic Algorithms

Adam Żychowski[1][0000-0003-0026-5183], Xin Yao[2][0000-0001-8837-4442], and
Jacek Mańdziuk[1,3][0000-0003-0947-028X]

[1] Warsaw University of Technology, Faculty of Mathematics and Information Science
[2] School of Data Science, Lingnan University, Hong Kong SAR
[3] Faculty of Computer Science, AGH University of Krakow, Krakow, Poland
adam.zychowski@pw.edu.pl, xinyao@ln.edu.hk, mandziuk@mini.pw.edu.pl

Abstract. This paper investigates the critical aspect of migration timing in hybrid island-based metaheuristic algorithms. Migration timing plays a pivotal role in balancing exploration and exploitation, ensuring that the algorithm avoids premature convergence while effectively exploring the search space. We propose and evaluate several migration timing strategies, including periodic migration, fitness-based triggers, and diversity-driven approaches. Our experiments are conducted on a set of benchmark optimization problems, including both discrete (Traveling Salesman Problem) and continuous (Black-box Optimization Benchmarking) tasks. The results demonstrate that adaptive migration strategies, which dynamically adjust based on population diversity and fitness stagnation, outperform static approaches. This study provides insights into the optimal conditions for triggering migration and offers guidelines for designing more effective hybrid metaheuristic frameworks.

Keywords: hybrid metaheuristic, island algorithm, migration strategy, distributed optimization

1 Introduction

Optimization algorithms are essential tools for solving complex problems in various domains, including logistics, engineering, and artificial intelligence. Traditional optimization methods, such as gradient-based approaches, often struggle with complex, high-dimensional, and multimodal landscapes. Population-based metaheuristics, such as Genetic Algorithms (GAs) [3], Particle Swarm Optimization (PSO) [7], or Ant Colony Optimization (ACO) [4], have emerged as powerful tools for tackling these challenges. These algorithms leverage a population of candidate solutions to explore the search space, iteratively refining solutions through mechanisms inspired by natural or social processes.

One strategy for improving the performance of population-based metaheuristics is their multiplication. A common approach involves constructing a portfolio of distinct algorithms that run in parallel [1, 13]. Another possibility consists in using a sequential process that adaptively selects a metaheuristics, suitable for the current state of the process, from a portfolio of diverse metaheuristics. A recent realization of this idea is the Constrained Hybrid Metaheuristic (cHM) approach [8, 9]. Yet another widely used approach is the island model [11], in which the global population is divided into subpopulations (islands) that evolve independently. These islands periodically exchange individuals through migration, facilitating the transfer of solutions between subpopulations. This structure not only promotes diversity but also allows subpopulations to specialize in different regions of the search space [19]. Previous studies (e.g. [10, 12]) demonstrate that the effectiveness of island-based models heavily depends on the migration strategy, particularly the timing of migration [18, 17].

The timing and frequency of migration in island-based models share similarities with the communication frequency in co-evolutionary algorithms. In co-evolution, subpopulations evolve independently but periodically exchange information to collaboratively solve complex problems. The optimal communication frequency in co-evolution has been shown to significantly impact performance, with too frequent or too infrequent communication leading to suboptimal results [14].

The timing of migration is crucial for balancing exploration (searching new areas of the solution space) and exploitation (refining existing solutions) [16]. Migrating too frequently can lead to premature convergence, where the subpopulations become too similar and lose diversity. This can trap the algorithm in a local optimum and prevent it from finding the global optimum. Conversely, infrequent migration can result in stagnation, where the subpopulations evolve independently for too long, hindering the sharing of beneficial solutions. This can slow down the optimization process and limit the quality of the final solution.

The most popular strategy for timing migration in island-based models is the use of fixed intervals, where migration occurs at regular, predefined intervals regardless of the population's state. However, this approach may not always be optimal, as it fails to adapt to the current performance of the optimization process. Furthermore, the optimal migration timing usually varies throughout the optimization process. In this work, we explore alternative migration timing strategies that are dynamically adjusted based on population diversity and fitness stagnation. We aim to identify the conditions under which the migration should occur in order to maximize the algorithm's performance.

2 Proposed Solution

2.1 Baseline algorithm

The baseline solution for testing various migration strategies is the Diversity-driven Cooperating Portfolio of Metaheuristics (DdCPM) framework, which we introduced in [20]. DdCPM integrates multiple metaheuristic algorithms (GA, PSO, or ACO) in an island-based setting. Each island employs a distinct metaheuristic to independently explore the search space. Such a setup, not only promotes diversity but also enables subpopulations to specialize in different regions of the search space, leveraging the complementary strengths of different algorithms. Islands periodically exchange individuals through migration, facilitating the transfer of solutions between subpopulations.

Employing distinct metaheuristics within each island reveals better the differences, advantages, and disadvantages of individual migration strategies. The use of different algorithms allows each population to evolve its unique characteristics, potentially boosting the importance of migration compared to a more homogeneous environment.

Figure 1 and Algorithm 1 present an overview of the DdCPM baseline setup used in the experiments. The DdCPM framework consists of N islands, with each island I characterized by three attributes: $I_{population}$ - a set of n_I individuals (candidate solutions); $I_{metaheuristic}$ - a specific metaheuristic algorithm used to evolve $I_{population}$; $I_{neighbours}$ - a set of neighboring islands from which migration to I can occur.

In experiments, we used 6 islands, with 2 islands assigned to each metaheuristic algorithm: GA, PSO, and ACO. The migration topology follows a clique structure, meaning that every island is directly connected to all the others, allowing individuals to migrate between any pair of islands. An individual selected for migration is determined by a sum of its *fitness* and *diversity* contribution to the target island, both normalized to [0,1]. Each island maintains a fixed population size of 100 individuals, and the total budget for fitness function evaluation is set to 10^5 evaluations. Please refer to [20] for a more detailed description of the DdCPM framework.

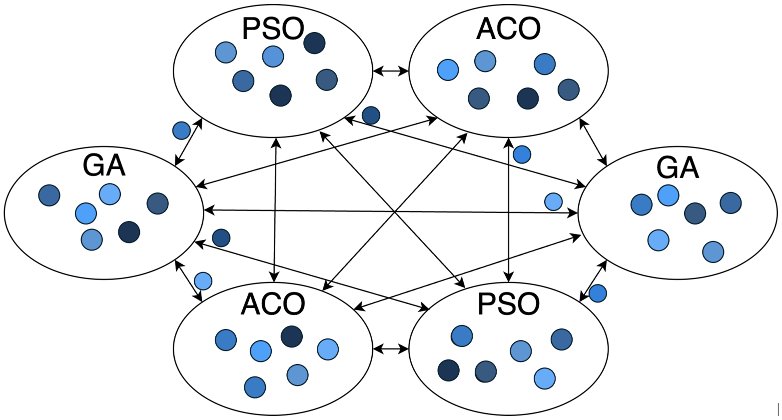

Fig. 1. Overview of the baseline DdCPM framework used for testing migration strategies. Populations in multiple islands are developed by different metaheuristics, with migrations between islands governed by certain migration triggers. A degree of shading indicates an individual's fitness value.

2.2 Migration timing strategies

At the core of population-based optimization algorithms are two fundamental driving forces: *fitness* and *diversity* [5, 2]. Fitness refers to the quality of a solution, typically measured by an objective function that the algorithm aims to optimize. High-fitness solutions guide the search towards promising regions of the search space, enabling the algorithm to exploit known good solutions. On the other hand, diversity refers to the variety of solutions within the population. Maintaining diversity is crucial for exploring

Algorithm 1 Pseudocode of the baseline DdCPM method.

```
for each island $I \in \mathcal{I}$ do
    initialize $I_{population}$ with random individuals
end for
while $evaluation\_budget > 0$ do
    for each island $I \in \mathcal{I}$ do
        if needs_migration($I$) then
            $I_{population} = I_{population} \cup$ migrate_from($I_{neighbours}$)
        end if
    end for
    for each island $I \in \mathcal{I}$ do
        $I_{population}$ = next_generation($I_{population}, I_{metaheuristic}$)
        evaluate($I_{population}$)
        $evaluation\_budget = evaluation\_budget - |I_{population}|$
    end for
end while
return the best individual from $\mathcal{I}$
```

new regions of the search space, preventing the algorithm from getting trapped in local optima.

We propose and evaluate 8 migration timing strategies (which we also refer to as *migration triggers*) most of them based on the two above-described fitness and diversity criteria. These strategies are designed to address the limitations of traditional approaches by dynamically adjusting migration timing based on the current state of the population. This allows for a more nuanced balance between exploration and exploitation, as opposed to static or randomized strategies that do not consider population health indicators.

T1: Periodic migration. In this baseline strategy, migration occurs at fixed intervals of m iterations, regardless of the state of the population. This is the most common approach in the literature and serves as a reference point for comparing more sophisticated strategies.

T2: Randomized. Migration occurs in each generation with a fixed probability of $m\%$. This stochastic approach reduces the likelihood of predictable migration patterns, promoting diversity and serving as a baseline for comparing other migration strategies.

T3: Average fitness stagnation. Migration is triggered when the average fitness of the population on an island does not improve for m consecutive iterations. This strategy aims to detect cases where the population has converged prematurely to a local optimum.

T4: Maximum fitness stagnation. Migration occurs if the best fitness value (maximum fitness) within an island's population does not improve for m iterations. This strategy focuses on identifying stagnation in achieving higher-quality solutions.

T5: Combined fitness stagnation. This strategy combines the previous two criteria. Migration is triggered when neither the average fitness nor the maximum fitness shows improvement over m iterations.

T6: Diversity stagnation. Migration is executed if the diversity within an island's population does not improve for m iterations. This strategy aims to introduce new genetic material and avoid premature convergence.

We measure diversity by calculating the average of the standard deviations across all problem dimensions. For each dimension of the solution encoding vectors, we compute the standard deviation and then average these values. Formally, given a population $I = \{i_1, i_2, \ldots, i_n\}$ and the solution vector for individual $i_k \in I$, represented as $\mathbf{x}^{i_k} = [x_1^{i_k}, x_2^{i_k}, \ldots, x_D^{i_k}]$, the diversity is defined as:

$$\frac{1}{D} \sum_{d=1}^{D} \sigma(\{x_d^{i_1}, x_d^{i_2}, \ldots, x_d^{i_n}\}), \tag{1}$$

where $\sigma(\{x_d^{i_1}, x_d^{i_2}, \ldots, x_d^{i_n}\})$ denotes the standard deviation of the values in dimension $d \in \{1, \ldots, D\}$ across all individuals in the population.

T7: Combined fitness and diversity stagnation. This hybrid strategy combines the conditions of T3 and T5. Migration occurs if either the average and maximum fitness fail to improve or population diversity stagnates for m iterations. This approach aims to simultaneously address both fitness stagnation and the lack of diversity.

T8: Sum of normalized fitness and diversity stagnation. Migration is triggered when the normalized sum of fitness and diversity remains unchanged for m consecutive iterations. Both metrics are scaled to $[0, 1]$ to balance solution quality and population diversity.

In all cases except T1, migrations are executed asynchronously. This means that a specific condition is checked for each island individually, and if the condition is met, migration to that island is triggered. As a result, the timing and frequency of migrations between islands may vary.

In each case, the individual selected for migration is the one with the highest sum of normalized fitness and normalized diversity. Specifically, for fitness, the best value (lowest in minimization problems and highest in maximization problems) is mapped to 1, while the worst is mapped to 0. Similarly, the diversity component is normalized such that the highest potential increase in diversity within the target population is mapped to 1, and the lowest to 0.

The selection of the migrated individual is made within populations of all islands that are connected to the destination island. In the case of a clique topology, where all islands are interconnected, the migration can potentially be performed from any island other than the target one.

3 Results

3.1 Experimental setup

We evaluated the proposed migration timing strategies on two benchmark problem domains: the Traveling Salesman Problem (TSP) and the Black-box Optimization Benchmarking (BBOB) framework. The selection of TSP and BBOB as benchmark domains ensures the evaluation of the proposed strategies across both discrete and continuous optimization landscapes. For TSP, we used 10 instances from the TSPLIB library [15], with the number of cities ranging from 400 to 724. For BBOB, we used 12 problem instances from the COCO platform [6], covering a diverse range of function properties, including unimodal, multimodal, and composite functions. The number of dimensions for all functions was set to 20. All experiments were conducted independently 20 times using different random seeds. The results presented are averaged across these runs, and standard deviations are provided to indicate the variability in performance. Tests were run on an Intel Xeon Silver 4116 @ 2.10GHz with 256GB RAM.

3.2 Migration frequency and *m* parameter

Tables 1 and 3 present the average results across all tested problem instances for various values of parameter *m* for each timing migration strategy (T1-T8).

Table 1. Averaged results for various *m* values for different migration timing strategies (see Section 2.2.) for TSP problems. The best results for each strategy are **bolded**. The best overall result is shaded.

	Migration timing strategy							
m	**T1**	**T2**	**T3**	**T4**	**T5**	**T6**	**T7**	**T8**
1	36205 ± 47	36536 ± 59	**35832** ± 88	35758 ± 60	35683 ± 55	**35671** ± 70	35646 ± 67	35716 ± 94
2	36193 ± 81	36211 ± 53	35857 ± 67	35745 ± 64	35671 ± 79	35696 ± 58	35634 ± 94	35661 ± 72
5	36168 ± 56	35894 ± 64	35944 ± 92	35721 ± 78	35621 ± 90	35745 ± 52	**35547** ± 79	35644 ± 81
10	36093 ± 76	**35625** ± 47	36019 ± 89	**35621** ± 66	**35559** ± 59	35820 ± 53	**35547** ± 65	**35619** ± 85
15	**35634** ± 70	35866 ± 58	36056 ± 50	35721 ± 71	35708 ± 80	35795 ± 75	35658 ± 90	35715 ± 68
20	35857 ± 91	36170 ± 61	36081 ± 80	35758 ± 93	35708 ± 55	35894 ± 47	35683 ± 88	35765 ± 79
25	36118 ± 67	36182 ± 73	36106 ± 54	35770 ± 68	35708 ± 69	35857 ± 78	35696 ± 60	35714 ± 84
30	36180 ± 60	36248 ± 57	36106 ± 76	35783 ± 91	35721 ± 61	35919 ± 69	35696 ± 65	35816 ± 81
50	36242 ± 81	36286 ± 64	36143 ± 93	35783 ± 54	35733 ± 88	35919 ± 55	35708 ± 80	35778 ± 92

The optimal value of *m* varies depending on the migration strategy. For example, T3 and T6, which migrate based solely on the average fitness and diversity, respectively, perform best for $m = 1$. This is likely because these metrics tend to decrease rapidly in the early stages of the optimization process, and larger values of *m* result in migrations occurring too infrequently. This observation is supported by Tables 2 and 4, which show the migration frequency for different values of *m*. For T3 and T6, even with very small values of *m*, the migration frequency exceeds 25 generations, indicating the specified conditions for migration are rarely met.

Table 2. Averaged frequency of migrations for various *m* values for different migration timing strategies (described in Section 2.2.) for the TSP problems.

	Migration timing strategy							
m	**T1**	**T2**	**T3**	**T4**	**T5**	**T6**	**T7**	**T8**
1	1.0	98.2	26.3	9.0	10.9	32.4	9.5	9.2
2	2.0	49.5	26.9	10.1	12.1	33.3	10.9	10.4
5	5.0	19.8	30.2	12.2	13.8	37.0	13.6	14.3
10	10.0	9.9	35.4	18.8	19.0	45.5	20.3	20.2
15	15.0	6.7	40.4	21.7	27.2	42.0	26.7	25.5
20	20.0	5.0	46.9	24.9	29.0	63.0	30.5	30.5
25	25.0	4.0	51.9	33.2	31.0	51.5	35.1	36.8
30	30.0	3.4	55.0	43.5	41.0	79.0	43.8	42.4
50	50.0	2.0	75.0	57.9	66.5	74.3	51.0	51.8

Table 3. Averaged results for various *m* parameter values for different migration timing strategies (see Section 2.2.) for BBOB problems. The best results for each strategy are **bolded**. The best overall result is shaded .

	Migration timing strategy							
m	**T1**	**T2**	**T3**	**T4**	**T5**	**T6**	**T7**	**T8**
1	2.808 ± .12	2.871 ± .09	**2.830** ± .11	1.605 ± .05	1.716 ± .07	**2.585** ± .10	1.515 ± .04	1.580 ± .06
2	2.808 ± .11	2.792 ± .08	2.875 ± .12	1.605 ± .06	1.716 ± .08	2.607 ± .09	1.515 ± .05	1.529 ± .07
5	2.786 ± .10	2.690 ± .07	2.986 ± .13	1.582 ± .05	1.659 ± .06	2.674 ± .11	1.493 ± .04	1.509 ± .05
10	2.697 ± .09	**2.241** ± .06	3.075 ± .14	**1.560** ± .04	**1.649** ± .07	2.674 ± .10	**1.449** ± .03	**1.497** ± .05
15	**2.229** ± .08	2.452 ± .07	3.120 ± .15	1.605 ± .05	1.716 ± .08	2.741 ± .12	1.515 ± .04	1.553 ± .06
20	2.451 ± .10	2.721 ± .09	3.142 ± .16	1.605 ± .06	1.738 ± .07	2.741 ± .11	1.515 ± .05	1.579 ± .07
25	2.719 ± .11	2.808 ± .10	3.165 ± .17	1.627 ± .05	1.738 ± .08	2.786 ± .13	1.519 ± .04	1.563 ± .06
30	2.786 ± .12	2.854 ± .11	3.187 ± .18	1.627 ± .06	1.738 ± .09	2.741 ± .14	1.525 ± .05	1.601 ± .07
50	2.853 ± .13	2.872 ± .12	3.231 ± .19	1.627 ± .07	1.738 ± .10	2.808 ± .15	1.522 ± .06	1.537 ± .08

For all other strategies, the optimal tested value of *m* belongs to {5, 10, 15}. Notably, for the majority of migration strategies, the optimal *m* value (in terms of the best results from Tables 1 and 3) corresponds to a similar migration frequency, ranging between approximately 10 and 20 generations. This finding suggests that there is a “sweet spot” for migration frequency that effectively balances exploration and exploitation. Migrating too frequently can lead to premature convergence, as the islands may homogenize and lose diversity. Conversely, migrating too rarely can result in stagnation, as the islands are unable to share beneficial solutions with one another.

The optimal migration frequency strikes a balance between these two extremes, facilitating both the exploration of new solutions and the exploitation of existing ones. A similar conclusion has been drawn in [14] regarding co-evolutionary algorithms.

3.3 Performance of migration strategies

Tables 5 and 6 present detailed performance results of the tested migration timing strategies for all evaluated TSP and BBOB instances, resp. For each strategy, the optimal *m*

Table 4. Averaged frequency of migrations for various m values for different migration timing strategies (described in Section 2.2.) for the BBOB problems.

	Migration timing strategy							
m	**T1**	**T2**	**T3**	**T4**	**T5**	**T6**	**T7**	**T8**
1	1.0	99.7	25.9	9.1	10.5	31.9	9.8	9.8
2	2.0	49.8	27.0	10.3	12.0	33.5	10.4	10.4
5	5.0	20.0	29.9	12.8	15.3	38.6	14.7	14.3
10	10.0	9.9	35.0	19.4	19.0	38.8	19.8	18.9
15	15.0	6.6	41.0	24.9	22.6	50.6	21.4	20.7
20	20.0	5.0	43.8	26.4	27.2	52.4	24.8	25.5
25	25.0	4.0	49.4	35.0	39.6	68.7	30.1	29.2
30	30.0	3.3	52.5	35.3	40.0	72.8	34.0	32.9
50	50.0	2.0	74.4	55.1	59.2	81.7	59.4	60.0

value, determined based on the analysis presented in the previous section, was employed.

Table 5. Detailed results for tested TSP instances for different migration timing strategies. The best results for each instance are **bolded**.

	Migration timing strategy							
Instance	**T1**	**T2**	**T3**	**T4**	**T5**	**T6**	**T7**	**T8**
rd400	15557 ± 87	15549 ± 92	15631 ± 105	15544 ± 78	15486 ± 95	15594 ± 89	**15485** ± 83	15535 ± 102
fl417	12108 ± 76	12087 ± 81	12222 ± 98	12085 ± 72	12036 ± 88	12113 ± 84	**12025** ± 79	12084 ± 91
pcb442	51420 ± 112	51434 ± 105	51576 ± 117	51406 ± 98	51375 ± 110	51494 ± 107	**51333** ± 102	51395 ± 115
d493	35497 ± 95	35468 ± 89	35783 ± 110	35478 ± 92	**35406** ± 97	35525 ± 101	35413 ± 88	35474 ± 104
att532	28111 ± 82	28080 ± 78	28281 ± 99	28083 ± 85	28027 ± 91	28155 ± 87	**28005** ± 84	28088 ± 96
si535	49069 ± 108	49083 ± 112	49177 ± 119	49054 ± 105	**48994** ± 115	49081 ± 110	49006 ± 107	49060 ± 118
u574	37425 ± 97	37418 ± 92	37529 ± 110	37405 ± 94	37338 ± 99	37425 ± 101	**37323** ± 96	37427 ± 108
p654	35116 ± 89	35119 ± 85	35474 ± 105	35104 ± 91	35068 ± 93	35169 ± 97	**35044** ± 88	35114 ± 102
d657	49552 ± 110	49545 ± 108	49752 ± 120	49516 ± 112	49467 ± 115	49556 ± 118	**49441** ± 107	49526 ± 119
u724	42485 ± 102	42469 ± 98	42895 ± 111	42446 ± 105	42398 ± 110	42593 ± 112	**42394** ± 108	42489 ± 115

Strategy T7, relying on the combined stagnation of fitness and diversity, yielded superior performance across the majority of tested instances. Statistically significant improvement (according to paired t-test with p-value $\leq$ 0.05) was observed for T7 compared to all other strategies in 5 out of 10 TSP instances and 7 out of 12 BBOB functions. Conversely, no other strategy demonstrated statistically significant superiority over T7 in any instance. These results underscore the efficacy of the combined fitness and diversity stagnation criterion for triggering migration. This likely stems from the fact that both fitness stagnation (indicating the lack of promising new individuals) and diversity stagnation (indicating the lack of exploration of the search space) serve as valuable indicators for migration. Relying on only one of these criteria appears to be suboptimal.

Strategy T4 (maximum fitness stagnation) and strategy T8 (normalized sum of fitness and diversity stagnations) yielded slightly lower, yet still competitive, perfor-

Table 6. Detailed results for tested BBOB functions for different migration timing strategies. The best results for each function are **bolded**.

	Migration timing strategy							
Function	**T1**	**T2**	**T3**	**T4**	**T5**	**T6**	**T7**	**T8**
f2	0.015 ± .001	0.015 ± .002	0.015 ± .001	**0.014** ± .001	0.015 ± .001	0.015 ± .001	**0.014** ± .001	0.015 ± .001
f4	1.712 ± .150	1.580 ± .140	2.253 ± .200	1.501 ± .130	1.568 ± .135	2.051 ± .180	**1.411** ± .125	1.509 ± .132
f6	1.315 ± .120	1.264 ± .115	1.341 ± .125	1.031 ± .095	1.019 ± .090	1.226 ± .110	**0.927** ± .085	1.000 ± .092
f8	2.721 ± .250	2.602 ± .240	3.772 ± .350	2.023 ± .180	2.150 ± .190	3.520 ± .320	**1.886** ± .170	1.921 ± .175
f10	6.098 ± .550	5.605 ± .500	5.786 ± .520	3.373 ± .300	3.741 ± .330	5.465 ± .490	**3.229** ± .290	3.439 ± .310
f12	7.447 ± .670	7.357 ± .660	9.904 ± .620	5.005 ± .450	5.092 ± .460	10.085 ± .902	**4.639** ± .420	4.673 ± .425
f14	0.816 ± .075	0.815 ± .074	1.059 ± .095	0.702 ± .065	0.742 ± .068	0.966 ± .088	**0.653** ± .060	0.689 ± .063
f16	1.099 ± .100	1.007 ± .092	1.248 ± .115	0.873 ± .080	0.913 ± .083	1.190 ± .110	0.818 ± .075	**0.810** ± .074
f18	1.337 ± .120	1.318 ± .118	1.630 ± .150	0.806 ± .070	0.840 ± .075	1.610 ± .145	0.777 ± .068	**0.769** ± .067
f20	1.812 ± .165	1.806 ± .163	2.259 ± .200	1.521 ± .140	1.593 ± .145	2.104 ± .190	**1.450** ± .135	1.458 ± .136
f22	0.818 ± .075	0.758 ± .070	0.971 ± .090	0.471 ± .045	0.499 ± .048	0.875 ± .080	**0.444** ± .042	0.457 ± .043
f24	1.947 ± .175	1.783 ± .160	2.138 ± .190	1.439 ± .130	1.522 ± .135	1.979 ± .180	**1.421** ± .128	1.438 ± .130

mance. The performance of T8 further supports the benefits of incorporating both fitness and diversity into the migration trigger. The relatively strong performance of T4 suggests that maximum fitness stagnation is also a useful indicator, likely because it reflects a point in the search where the population has converged to a local optimum and further exploration is needed.

Strategies employing solely diversity (T6) or average fitness (T3) as migration criteria exhibited significantly weaker performance. This observation, consistent with the findings presented in Section 3.2, reinforces the conclusion that these individual metrics lead to insufficiently frequent migrations.

The traditional approach of employing a fixed migration interval (T1), commonly found in the literature, outperformed T3 and T6, but performed significantly worse than the other adaptive strategies, particularly T7. The performance of the randomized migration strategy (T2) was comparable to that of T1. This similarity likely arises from the fact that neither approach incorporates information about the population's state or behavior to inform the migration timing, thereby triggering migrations independently of any emergent population dynamics.

3.4 Computation overhead

Table 7 presents the average computation time for different migration strategies (with the optimal m value set according to experiments discussed in Section 3.2). The computational cost of monitoring fitness and diversity for adaptive migration timing is significantly smaller compared to the cost of fitness function evaluations. Fitness stagnation checks involve simple comparisons, while diversity tracking requires standard deviation calculations, both of which are lightweight operations.

In our experiments, the time spent on these monitoring tasks was significantly smaller than the time required for fitness evaluations. Thus, the minor overhead introduced by adaptive strategies does not impact the overall time efficiency.

Table 7. Average computational time (in sec) for different migration strategies.

	Migration timing strategy							
Problem	**T1**	**T2**	**T3**	**T4**	**T5**	**T6**	**T7**	**T8**
TSP	45.2 ± 0.8	46.1 ± 0.9	47.5 ± 1.0	47.0 ± 0.9	48.3 ± 1.1	48.9 ± 1.2	49.8 ± 1.3	49.2 ± 1.2
BBOB	62.8 ± 1.1	63.5 ± 1.2	65.4 ± 1.3	64.7 ± 1.2	66.1 ± 1.4	66.9 ± 1.5	68.0 ± 1.6	67.3 ± 1.5

4 Conclusions

This study explores the critical role of migration timing in hybrid island-based metaheuristic algorithms. We propose and evaluate various migration strategies, including periodic, randomized, fitness-based, diversity-driven, and combined approaches, to identify optimal methods to balance exploration and exploitation.

Our findings show that adaptive migration strategies, particularly those that combine fitness and diversity metrics, outperform static methods. Traditional approaches, such as fixed-interval migration and randomized migration, while simpler to implement, lack the adaptability needed to respond to changes in population dynamics. In contrast, adaptive strategies dynamically adjust migration timing based on multiple population health indicators, leading to more effective exploration and exploitation. This highlights the importance of dynamic, context-aware migration. These findings align with the observations made regarding co-evolutionary algorithms, where the frequency of communication among subpopulations has been shown to significantly impact performance [14]. This suggests that insights from one area can guide the design of optimization frameworks in the other, highlighting the broader applicability of migration timing strategies in distributed optimization.

Future work will focus on extending the set of metaheuristic algorithms and diverse problem types tested within the DdCPM framework. We also plan to investigate the impact of varying the number of islands and utilizing other island topologies (such as ring or star structures) on the performance of the migration strategies.

Acknowledgment

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 was supported by the National Science Centre, grant number 2023/49/B/ST6/01404.

References

1. Akay, R., Basturk, A., Kalinli, A., Yao, X.: Parallel population-based algorithm portfolios: An empirical study. Neurocomputing **247**, 115–125 (2017)
2. Araujo, J.N., Batista, L.S.: A diversity-driven migration strategy for distributed evolutionary algorithms. Swarm and Evolutionary Computation **82**, 101361 (2023)
3. Bäck, T., Schwefel, H.P.: An overview of evolutionary algorithms for parameter optimization. Evolutionary computation **1**(1), 1–23 (1993)

4. Dorigo, M., Birattari, M., Stutzle, T.: Ant colony optimization. IEEE computational intelligence magazine **1**(4), 28–39 (2006)
5. Gupta, D., Ghafir, S.: An overview of methods maintaining diversity in genetic algorithms. International journal of emerging technology and advanced engineering **2**(5), 56–60 (2012)
6. Hansen, N., Auger, A., Ros, R., Mersmann, O., Tušar, T., Brockhoff, D.: Coco: A platform for comparing continuous optimizers in a black-box setting. Optimization Methods and Software **36**(1), 114–144 (2021)
7. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of International Conference on Neural Networks. pp. 1942–1948 (1995)
8. Kowalski, P.A., Kucharczyk, S., Mańdziuk, J.: Constrained hybrid metaheuristic algorithm for probabilistic neural networks learning (2025), `https://arxiv.org/abs/2501.15661`
9. Kowalski, P.A., Kucharczyk, S., Mańdziuk, J.: Evaluating the effectiveness and stability of the constrained hybrid metaheuristic algorithm in probabilistic neural networks training. In: [under review] (2025)
10. Lardeux, F., Maturana, J., Rodriguez-Tello, E., Saubion, F.: Migration policies in dynamic island models. Natural Computing **18**, 163–179 (2019)
11. Leitão, A., Pereira, F.B., Machado, P.: Island models for cluster geometry optimization: how design options impact effectiveness and diversity. Journal of Global Optimization **63**, 677–707 (2015)
12. Li, J., Gonsalves, T.: Parallel hybrid island metaheuristic algorithm. IEEE Access **10**, 42268–42286 (2022)
13. Liu, S., Tang, K., Yao, X.: Automatic construction of parallel portfolios via explicit instance grouping. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 1560–1567 (2019)
14. Lu, X., Menzel, S., Tang, K., Yao, X.: Cooperative co-evolution-based design optimization: A concurrent engineering perspective. IEEE Transactions on Evolutionary Computation **22**(2), 173–188 (2017)
15. Reinelt, G.: TSPLIB — A traveling salesman problem library. ORSA journal on computing **3**(4), 376–384 (1991)
16. da Silveira, L.A., de Lima, T.A., de Barros, J.B., Soncco-Álvarez, J.L., Llanos, C.H., Ayala-Rincón, M.: On the behavior of parallel island models. Applied Soft Computing **148**, 110880 (2023)
17. Skolicki, Z.: An analysis of island models in evolutionary computation. In: Proceedings of the 7th annual workshop on Genetic and evolutionary computation. pp. 386–389 (2005)
18. Skolicki, Z., De Jong, K.: The influence of migration sizes and intervals on island models. In: Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation. pp. 1295–1302 (2005)
19. Whitley, D., Rana, S., Heckendorn, R.B.: The island model genetic algorithm: On separability, population size and convergence. Journal of Computing and Information Technology **7**(1), 33–47 (1999)
20. Żychowski, A., Yao, X., Mańdziuk, J.: Diversity-driven cooperating portfolio of metaheuristic algorithms. In: [under review] (2025)