Migrant Selection in Island-Based Optimization

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. Island-based metaheuristics have gained significant attention in the field of optimization due to their ability to maintain population diversity and avoid premature convergence. A critical component of these algorithms is the migration strategy, which determines how individuals are exchanged between islands. This paper investigates the impact of different migration strategies on the performance of island-based metaheuristics, with a particular focus on the number of migrated individuals and the criteria for their selection. We propose several strategies for selecting individuals for migration, including random selection, fitness-based, diversity-based and hybrid approaches, and evaluate their effectiveness on a set of TSP (Traveling Salesman Problem) and BBOB (Black-box Optimization Benchmarking) problems. Our results demonstrate that the choice of migration strategy significantly affects the algorithm's performance. Specifically, selecting individuals based not only on fitness but also on their potential to increase diversity leads to better outcomes.

Keywords: Optimization metaheuristic · Island-based algorithm · Distributed optimization · Migration strategy.

1 Introduction

Population-based metaheuristics, such as Genetic Algorithms (GAs) [3], Particle Swarm Optimization (PSO) [9], or Ant Colony Optimization (ACO) [5], have been widely used to solve complex optimization problems [15]. These algorithms rely on a population of candidate solutions that evolve over time, aimed at finding high-quality solutions. However, one of the main challenges in population-based optimization is maintaining diversity within the population to avoid premature convergence to suboptimal solutions. Premature convergence can lead to the algorithm getting stuck in local optima, which is particularly harmful in complex, multi-modal optimization landscapes.

Island-based models [17] address this challenge by dividing the population into subpopulations (islands) that evolve independently. These islands periodically exchange individuals through a process called migration, which helps maintain diversity and allows subpopulations to explore distinct regions of the search space. The effectiveness of island-based models depends heavily on the migration strategy [14], which determines how individuals are selected for migration, when migration occurs, and where individuals are migrated [18].

While previous research has extensively studied the timing and topology of migration, the question of how many individuals to migrate and which individuals to select has received less attention. The number of migrated individuals and criteria for their selection can significantly impact the algorithm's performance, as they influence the balance between exploration and exploitation. Migrating too few individuals may limit the exchange of information between islands, while migrating too many of them may disrupt the search process and lead to premature convergence.

In this paper, we investigate the impact of different migration strategies on the performance of island-based metaheuristics, with a particular focus on the number of migrated individuals and the criteria for their selection. We propose several strategies for selecting individuals for migration, including fitness-based and diversity-based selections, their hybridization, as well as a randomized approach. Their effectiveness is evaluated on a set of benchmark problems, providing a guidance for designing more effective distributed optimization algorithms.

2 Related work

Island-based optimization is a well-established paradigm in evolutionary computation, wherein multiple subpopulations (islands) evolve in parallel and periodically exchange individuals through a migration mechanism. Several early works [17, 2, 1] demonstrated that island models outperform classical EAs. However, the effectiveness of these models largely depends on migration strategies, including the selection of migrants, migration frequency, and reintegration policies [14, 10, 8, 19].

Selection of individuals to migrate plays a crucial role in migration effectiveness. The most common strategies include random selection and fitness-based one [12]. Random selection, though simple and fast, may not always promote the exchange of high-quality solutions. In contrast, fitness-based approaches (e.g., best or worst individual selection) ensure that migration contributes to the overall progress, but they may also lead to premature convergence due to reducing population diversity [4]. This issue was addressed in the frequency-based fitness assignment method [16] that replaces the traditional objective function with a measure of how often solutions with the same fitness value have been found.

Recent advancements have focused on adaptive migration models based on real-time population metrics. For instance, [11] introduced self-adaptive migration schemes that modify selection rules based on diversity and convergence indicators, allowing more flexible and problem-specific optimization. Other studies [6] have explored island attractiveness as a migration destination and the impact of the island interconnection topology.

Despite significant progress, open challenges remain in understanding the optimal balance between the selection pressure and the diversity maintenance in the context of migration strategies. The impact of different migration selection schemes under various optimization landscapes is still an active area of research.

This paper builds on prior work [18] where the authors explored various migration strategies answering three fundamental questions: "*When* to migrate?", "*What* to migrate?", and "*Where* to migrate?". In this paper, we focus on systematic analysis of another critical migration aspect: *a mechanism of selecting individual(s) for migration*. We evaluate the effects of particular selection mechanisms on population diversity and convergence, and propose enhanced selection models.

3 Strategies of migrants selection

In island-based metaheuristics, the selection of individuals for migration is a critical factor influencing the algorithm's ability to balance exploration and exploitation [4]. Traditional approaches often migrate a single individual based on fitness. However, migrating multiple individuals can enhance the exchange of diverse solutions between islands, potentially improving overall search performance. This approach opens up various strategic possibilities for selecting migrants. Both the number of individuals to migrate (K) and the criteria for their selection (S) must be carefully determined to prevent disruption of the search process or premature convergence.

We propose several strategies for selecting individuals for migration, each with its own underlying intuition and potential benefits:

S0: Random Selection: Randomly selects K individuals without considering fitness or diversity. This method serves as a baseline to benchmark the necessity of strategic selection.

S1: Top K Fitness: Selects K individuals with the highest fitness values from the donor islands. The intuition is that migrating the best solutions can help propagate high-quality solutions across islands, promoting exploitation. However,if overused this approach may reduce diversity, potentially leading to premature convergence.

S2: Top K Diversity: Selects K individuals that maximize the diversity of the receiving island's population. Diversity is measured using the average standard deviation of individuals across all dimensions. The goal is to introduce new genetic material to the receiving island, enhancing exploration. However, note that this strategy may potentially migrate individuals that are outliers, not necessarily beneficial for convergence.

S3: Top $\frac{K}{2}$ Fitness + Top $\frac{K}{2}$ Diversity: This hybrid approach selects half of the migrants based on fitness and the other half based on diversity. The idea is to improve effectiveness by combining the strengths of two previous strategies - exploiting high-quality solutions while maintaining genetic diversity.

S4: Top K (Fitness + Diversity): Selects K individuals based on a combined score of fitness and diversity. The score is computed as the sum of normalized fitness (χ) and normalized diversity (δ), where both metrics are scaled to [0,1]. Specifically, the fitness value f_i of the candidate solution i is transformed to $\chi_i \in [0, 1]$ by applying min-max normalization:

$$\chi_i = \frac{f_i - f_{min}}{f_{max} - f_{min}}$$

where f_{min} and f_{max} represent the minimum and maximum fitness values across the current population. The diversity metric is computed as described in S2 and normalized in the same way as the fitness metric. The final score in S4 is calculated as the sum of these normalized fitness and diversity metrics: $\chi_i + \delta_i$. This approach aims to balance exploration and exploitation by considering both criteria simultaneously.

S5: Weighted Random Selection: Assigns selection probabilities to individuals based on the sum of their normalized fitness and diversity scores (as in S4). Each individual i is then chosen stochastically according to its probability:

$$p(i) = \frac{\chi(i) + \delta(i)}{\sum_{j \in I}(\chi(j) + \delta(j))},$$

where $\chi(i)$ and $\delta(i)$ are the normalized fitness and diversity of the individual i respectively, and I is the set of candidate migrants. Unlike deterministic selection in S4, this method introduces stochasticity, preventing dominant individuals from always being chosen while still favoring well-rounded candidates.

S6: Weighted K (Fitness + Diversity): Selects K individuals using a dynamically weighted sum of fitness (χ) and diversity (δ). For the j-th selected individual ($j \in \{1, \ldots, K\}$), the selection maximizes the weighted sum:

$$\frac{j-1}{K-1}\chi + (1 - \frac{j-1}{K-1})\delta.$$

This means that the first selected individual ($j = 1$) is chosen solely based on diversity (as in S2), while the last one ($j = K$) is chosen solely based on fitness (as in S1). This method gradually shifts priority between selected individuals from diversity to fitness.

S7: Top 1 Fitness from K Clusters: Clusters the population into K groups using the K-means algorithm based on solution similarity. The best-fitted individual from each cluster is then selected for migration. This approach ensures that migrants are both high-quality and diverse, representing different regions of the search space.

Each strategy presents its unique properties influenced by the problem landscape and the state of the optimization process. In subsequent sections, we evaluate these strategies experimentally to assess their impact on the performance of the island-based algorithm.

In strategies that rely on diversity (S2 - S6) individuals are selected sequentially, i.e. after each individual is selected, the diversity metric is recalculated to account for the impact of the newly added individual on the remaining candidates.

A general limitation of diversity-based migrant selection strategies is their dependence on a reliable distance measure between solutions in the search space. Such measures can be challenging to define when elements of solution vectors have different units or scales. While normalization techniques can mitigate these issues, their effectiveness and appropriateness tend to be problem-specific. Thus, a careful consideration and possibly problem-specific normalization of strategies are necessary when applying the proposed strategies to particular optimization problems.

4 Experimental setup

4.1 Baseline method

The baseline method used in this study is the Diversity-driven Cooperating Portfolio of Metaheuristics (DdCPM) algorithm [18], which integrates multiple metaheuristics in an island-based framework. The algorithm employs a migration mechanism to dynamically adjust the flow of individuals between islands. DdCPM was chosen as the baseline due to the inherent diversity between islands (each of them is developed using a different metaheuristic), making it a suitable method for evaluating the impact of different migration strategies.

The pseudocode of the baseline DdCPM method is provided in Algorithm 1. Each island $I \in \mathcal{I}$ is defined by three attributes: $I_{population}$, which represents a set of n_I candidate solutions (individuals); $I_{metaheuristic}$, the specific metaheuristic algorithm used to evolve $I_{population}$; and $I_{neighbours} \subset \mathcal{I}$, the set of neighboring islands from which migration to I can occur.

Algorithm 1 Pseudocode of the baseline DdCMP 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 $\mathcal{I}$

In experiments, we used the DdCMP variant tested in [18], which consists of six islands. Each of the three metaheuristic algorithms (GA, PSO, and ACO) is assigned to two islands. The migration topology follows a clique structure, where every island is directly connected to all the others, enabling individuals to migrate freely between any pair of islands. Each island maintains a fixed population size of 100 individuals. The stopping criterion is based on a total fitness function evaluation budget of 10^5 evaluations.

4.2 Benchmark problems

To evaluate the effectiveness of the proposed migration strategies, we conducted experiments in two well-established benchmark problem domains: the Traveling Salesman Problem (TSP) and the Black-box Optimization Benchmarking (BBOB) framework.

For the TSP, we selected 10 instances from the TSPLIB library [13], ranging from 400 to 724 nodes. These instances are known for their complexity and are widely used to evaluate optimization algorithms. For the BBOB, we used all 24 problem instances from the COCO platform [7], covering a diverse range of functions, including separable functions, multi-modal functions, and functions with varying conditioning. For all functions the number of dimensions was set to 20.

The TSP represents a discrete optimization problem, while the BBOB functions are continuous optimization tasks. This distinction allows us to evaluate the performance of the proposed migration strategies across different types of optimization problems.

Each experiment was performed independently 20 times with different random seeds, and the presented results are the averages of these runs. Statistical significance was checked according to the Wilcoxon signed-rank test with p-value ≤ 0.05.

5 Results

5.1 Number of migrants

The number of individuals migrated between islands (K) plays an important role in the performance of the island-based DdCMP method. We evaluated the impact of different values of K (ranging from 1 to 24) on the algorithm's performance. The results, averaged over 20 independent runs, are presented in Tables 1 and 2 for TSP and BBOB problems respectively. For the strategies S3 and S6, the case of $K = 1$ is not applicable.

Table 1. Averaged results for various K values and different migrants selection strategies (cf. Section 3) for the TSP instances. The best results for each strategy are **bolded**. The best overall result is shaded.

	K							
Strategy	**1**	**2**	**4**	**8**	**12**	**16**	**20**	**24**
S0	36231 ± 107	36165 ± 113	**36211** ± 123	36267 ± 117	36249 ± 110	36255 ± 102	36272 ± 109	36285 ± 115
S1	35795 ± 81	35789 ± 48	35760 ± 66	35746 ± 87	35750 ± 44	35737 ± 42	**35731** ± 52	35768 ± 56
S2	36178 ± 74	36158 ± 55	36120 ± 83	36108 ± 39	**36100** ± 50	36103 ± 61	36105 ± 68	36119 ± 72
S3	-	35737 ± 78	35703 ± 40	35701 ± 53	**35689** ± 61	35696 ± 69	35712 ± 75	35728 ± 79
S4	35622 ± 44	35572 ± 67	35523 ± 49	**35461** ± 62	35475 ± 64	35521 ± 85	35540 ± 88	35562 ± 92
S5	35920 ± 89	35885 ± 92	35894 ± 87	**35884** ± 91	35912 ± 94	35968 ± 97	35985 ± 102	36010 ± 104
S6	-	35746 ± 80	35614 ± 53	35316 ± 76	35302 ± 71	**35238** ± 45	35272 ± 58	35310 ± 63
S7	35795 ± 71	35646 ± 42	35596 ± 59	**35474** ± 82	35485 ± 63	35532 ± 47	35558 ± 64	35582 ± 69

The results indicate that the impact of increasing K on the DdCMP performance varies across strategies. For both TSP and BBOB benchmarks, the most noticeable performance improvements occur for $K \leq 12$. Beyond this point, the performance either degrades or enters a plateau with only marginal improvement. The only exception is strategy S1, which for the TSP instances report the highest results for $K = 20$. Overall, in most cases, the best performance points are accomplished for $K = 8, 12, 16$ which suggests that migrating a moderate number of individuals is sufficient to achieve a good balance between exploration and exploitation. Beyond a certain point, additional migrated individuals do not significantly contribute to the solution improvement, or may even disrupt the search process. For all strategies, the results for the best K value are not statistically significantly better than those for the second-best K. However, for each

Table 2. Averaged results for various K values and different migrants selection strategies (cf. Section 3) for the BBOB instances. The best results for each strategy are **bolded**. The best overall result is shaded .

	K							
Strategy	**1**	**2**	**4**	**8**	**12**	**16**	**20**	**24**
S0	2.413 ± 0.13	2.377 ± 0.12	2.269 ± 0.14	**2.268** ± 0.12	2.285 ± 0.13	2.418 ± 0.14	2.506 ± 0.15	2.493 ± 0.14
S1	2.205 ± 0.10	2.122 ± 0.09	2.057 ± 0.10	2.046 ± 0.09	2.054 ± 0.10	**2.040** ± 0.10	2.058 ± 0.11	2.113 ± 0.10
S2	2.598 ± 0.13	2.611 ± 0.11	2.520 ± 0.12	2.555 ± 0.10	**2.504** ± 0.11	2.508 ± 0.11	2.571 ± 0.12	2.576 ± 0.11
S3	-	2.154 ± 0.10	2.140 ± 0.09	2.115 ± 0.09	2.155 ± 0.10	**2.063** ± 0.11	2.094 ± 0.10	2.155 ± 0.11
S4	1.594 ± 0.08	1.606 ± 0.07	1.530 ± 0.07	1.468 ± 0.07	**1.445** ± 0.08	1.455 ± 0.08	1.502 ± 0.09	1.554 ± 0.08
S5	2.090 ± 0.09	2.016 ± 0.10	1.987 ± 0.09	**1.891** ± 0.09	1.956 ± 0.10	2.062 ± 0.09	2.012 ± 0.10	2.005 ± 0.11
S6	-	2.200 ± 0.10	1.964 ± 0.09	1.532 ± 0.08	1.531 ± 0.07	**1.428** ± 0.07	1.478 ± 0.08	1.468 ± 0.08
S7	2.131 ± 0.10	1.854 ± 0.09	1.707 ± 0.08	1.589 ± 0.08	**1.512** ± 0.07	1.522 ± 0.07	1.520 ± 0.08	1.576 ± 0.07

strategy, the results for the best K are statistically significantly better than those for at least half of the worst-performing K values tested.

Strategies that combine fitness and diversity, namely S4, S6, and S7, turned out to perform best. These strategies consistently outperformed simpler approaches, such as random selection (S0) or purely fitness-based selection (S1). Statistical tests confirmed that the differences among S4, S6, and S7 were not significant, but each of these strategies significantly outperformed the others (S0, S1, S2, S3, and S5). This highlights the importance of balancing fitness and diversity when selecting migrants, as strategies that focus solely on one of these aspects tend to underperform. Strategy S6 with $K = 16$ achieved the best overall result across all tested settings.

The worst-performing strategies were S0 (random selection) and S2 (diversity-based selection). Random selection, while simple, does not leverage any information about the quality or diversity of individuals, leading to suboptimal performance. On the other hand, S2, which selects individuals solely based on diversity, often migrates outliers that significantly increase diversity but contribute little or nothing to the fitness.

5.2 Detailed results

Tables 3 and 4 present detailed results for each tested TSP and BBOB instance, resp. The results correspond to the values of K that yielded the best average performance for each strategy, reported in Tables 1 and 2, resp. Presented outcomes confirm our earlier observation that strategies S4, S6, and S7 generally outperform the others. At the same time, the choice of the most profitable strategy among these three depends on specific problem instance.

5.3 Migration source: *one individual from each island* vs. *a common pool*

To further investigate the impact of migration strategies in island-based optimization, we conducted experiments to compare two meta-approaches for selecting migrants:

- **One from each island**: In this approach, one individual is selected from each of the K islands (in our experiments $K = 5$, as DdCMP considers 6 islands). This ensures that migrants are drawn from diverse subpopulations, promoting exploration.

Table 3. Detailed results for tested TSP instances for different migrant selection strategies. The best results for each instance are **bolded**.

	TSP instance									
Strategy	*rd400*	*fl417*	*pcb442*	*d493*	*att532*	*si535*	*u574*	*p654*	*d657*	*u724*
S0 K=4	15825 ± 86	12304 ± 83	52315 ± 89	36112 ± 104	28578 ± 91	49925 ± 91	38080 ± 86	35754 ± 90	50423 ± 97	43244 ± 98
S1 K=20	15602 ± 93	12130 ± 94	51611 ± 95	35597 ± 96	28176 ± 97	49222 ± 98	37545 ± 99	35249 ± 100	49725 ± 93	42587 ± 102
S2 K=12	15764 ± 101	12253 ± 105	52115 ± 107	35966 ± 101	28461 ± 103	49729 ± 101	37915 ± 106	35603 ± 109	50221 ± 112	43033 ± 109
S3 K=12	15583 ± 87	12113 ± 114	51534 ± 115	35565 ± 116	28143 ± 117	49181 ± 118	37485 ± 119	35181 ± 120	49646 ± 121	42568 ± 122
S4 K=8	15475 ± 85	**11870** ± 124	51309 ± 125	35375 ± 126	28004 ± 127	48941 ± 128	37297 ± 129	35006 ± 130	**48971** ± 131	42346 ± 132
S5 K=8	15668 ± 78	12177 ± 99	51804 ± 109	35748 ± 107	28295 ± 105	49455 ± 97	37673 ± 87	35375 ± 103	49890 ± 93	42753 ± 92
S6 K=16	**15297** ± 86	12011 ± 134	**50841** ± 102	35404 ± 83	**27725** ± 84	**48477** ± 85	37327 ± 86	**34680** ± 87	49414 ± 88	**41971** ± 89
S7 K=8	15488 ± 90	12021 ± 91	51350 ± 92	**35046** ± 93	28026 ± 94	48980 ± 95	**36952** ± 96	35034 ± 97	49453 ± 98	42380 ± 99

Table 4. Detailed results for BBOB functions f1 to f12 for different migrant selection strategies. The best results for each instance are **bolded**.

	BBOB function											
Strategy	*f1*	*f2*	*f3*	*f4*	*f5*	*f6*	*f7*	*f8*	*f9*	*f10*	*f11*	*f12*
S0 K=8	**0.000** ± .000	0.017 ± .001	2.845 ± .142	2.212 ± .110	1.876 ± .094	1.437 ± .071	3.542 ± .177	3.031 ± .151	6.124 ± .306	5.273 ± .264	8.235 ± .412	7.176 ± .359
S1 K=16	**0.000** ± .000	0.016 ± .001	2.532 ± .127	2.022 ± .101	1.723 ± .086	1.318 ± .066	3.128 ± .156	2.653 ± .132	5.346 ± .267	4.537 ± .226	7.584 ± .379	6.568 ± .328
S2 K=12	**0.000** ± .000	0.016 ± .001	3.124 ± .156	2.479 ± .124	2.053 ± .103	1.588 ± .079	4.127 ± .206	3.493 ± .175	6.742 ± .337	5.838 ± .292	8.953 ± .448	7.770 ± .389
S3 K=16	**0.000** ± .000	**0.015** ± .001	2.578 ± .129	2.048 ± .102	1.746 ± .087	1.336 ± .067	3.321 ± .166	2.805 ± .140	5.642 ± .282	4.873 ± .244	7.612 ± .381	6.589 ± .329
S4 K=12	**0.000** ± .000	**0.015** ± .001	**1.872** ± .094	1.515 ± .076	**1.247** ± .062	**0.968** ± .048	**2.415** ± .121	**2.036** ± .102	**4.127** ± .206	3.501 ± .175	**5.246** ± .262	**4.485** ± .224
S5 K=8	**0.000** ± .000	**0.015** ± .001	2.453 ± .123	1.961 ± .098	1.685 ± .084	1.304 ± .065	3.027 ± .151	2.548 ± .127	5.128 ± .256	4.354 ± .218	7.124 ± .356	6.147 ± .307
S6 K=16	**0.000** ± .000	**0.015** ± .001	1.895 ± .095	**1.420** ± .071	1.263 ± .063	0.971 ± .049	2.527 ± .126	2.124 ± .106	4.215 ± .211	**3.384** ± .169	5.724 ± .286	4.902 ± .245
S7 K=12	**0.000** ± .000	**0.015** ± .001	2.014 ± .101	1.589 ± .079	1.387 ± .069	1.081 ± .054	2.493 ± .125	2.111 ± .105	4.352 ± .218	3.635 ± .182	5.635 ± .282	4.816 ± .241

Table 5. Detailed results for BBOB functions f13 to f24 for different migrant selection strategies. The best results for each instance are **bolded**.

	BBOB function											
Strategy	*f13*	*f14*	*f15*	*f16*	*f17*	*f18*	*f19*	*f20*	*f21*	*f22*	*f23*	*f24*
S0 K=8	1.235 ± .062	1.046 ± .052	1.427 ± .071	1.288 ± .064	1.356 ± .068	1.184 ± .059	2.573 ± .129	2.246 ± .112	0.825 ± .041	0.703 ± .035	2.452 ± .123	2.147 ± .107
S1 K=16	1.072 ± .054	0.903 ± .045	1.246 ± .062	1.120 ± .056	1.157 ± .058	0.996 ± .050	2.215 ± .111	1.930 ± .096	0.728 ± .036	0.621 ± .031	2.267 ± .113	1.980 ± .099
S2 K=12	1.312 ± .066	1.108 ± .055	1.512 ± .076	1.363 ± .068	1.472 ± .074	1.293 ± .064	2.935 ± .147	2.570 ± .128	0.886 ± .044	0.754 ± .038	2.764 ± .138	2.413 ± .121
S3 K=16	1.087 ± .054	0.922 ± .046	1.284 ± .064	1.153 ± .058	1.195 ± .060	1.036 ± .052	2.412 ± .121	2.103 ± .105	0.742 ± .037	0.630 ± .032	2.285 ± .114	1.994 ± .100
S4 K=12	**0.804** ± .040	0.682 ± .034	**0.925** ± .046	**0.811** ± .041	**0.872** ± .044	0.754 ± .038	**1.726** ± .086	**1.483** ± .074	**0.543** ± .027	0.467 ± .023	**1.675** ± .084	**1.459** ± .073
S5 K=8	1.024 ± .051	0.870 ± .043	1.235 ± .062	1.112 ± .056	1.142 ± .057	0.990 ± .049	2.135 ± .107	1.860 ± .093	0.672 ± .034	0.568 ± .028	2.218 ± .111	1.940 ± .097
S6 K=16	0.815 ± .041	**0.652** ± .033	0.942 ± .047	0.823 ± .041	0.884 ± .044	**0.751** ± .037	1.784 ± .089	1.549 ± .078	0.562 ± .028	**0.439** ± .022	1.724 ± .086	1.493 ± .075
S7 K=12	0.857 ± .043	0.717 ± .036	0.987 ± .049	0.860 ± .043	0.946 ± .047	0.818 ± .041	1.874 ± .094	1.627 ± .081	0.584 ± .029	0.489 ± .024	1.765 ± .088	1.533 ± .077

– **Common pool**: In this approach, all individuals from all islands are combined into a single pool, and K individuals are selected from this pool. This allows for a more centralized selection process, potentially favoring higher-quality or more diverse individuals. This approach was also employed in all previous experiments, as Dd-CMP implements a clique island migration topology.

The goal of these experiments is to determine whether selecting migrants from a common pool (*global perspective*) or one from each island (*local perspective*) leads to better performance. This distinction is particularly important for strategies that rely on global information, such as diversity-based or hybrid approaches, as the source of migrants can significantly influence the final algorithm performance of the overall algorithm.

We tested both approaches for strategies S0 (Random Selection), S1 (Top K Fitness), S2 (Top K Diversity), S4 (Top K (Fitness + Diversity)), and S5 (Weighted Random Selection). The remaining strategies were excluded from this analysis because they inherently rely on a global perspective (they require selection of $K > 1$ individuals from one population to compute meaningful scores or clusters).

Table 6. Performance of migration strategies with different migration sources one from each island vs. common pool ($K = 5$). The best results for each strategy are **bolded**. Statistically significant differences between migration sources is shaded

	TSP		BBOB	
Strategy	**One from each island**	**Common pool**	**One from each island**	**Common pool**
S0	**36245** ± 102	36312 ± 108	**2.436** ± 0.15	2.505 ± 0.16
S1	**35737** ± 42	35812 ± 47	**2.049** ± 0.10	2.108 ± 0.11
S2	36103 ± 61	**36078** ± 65	**2.534** ± 0.11	2.5516 ± 0.12
S4	35521 ± 85	**35412** ± 89	1.522 ± 0.08	**1.495** ± 0.09
S5	35968 ± 97	**35845** ± 101	2.014 ± 0.10	**2.001** ± 0.11

The results averaged over all instances are presented in Table 6. For S0 and S1, the *one from each island* approach performs better than the *common pool* approach. This is probably because selecting one individual from each island introduces diversity into the target island, which helps maintain exploration and prevents premature convergence. In contrast, the common pool approach may lead to over-concentration of similar individuals from one island.

For S2 there is no significant difference between both approaches. S2 focuses solely on diversity, and selecting one individual from each island already ensures a diverse set of migrants. The common pool approach does not provide additional benefits in this case, as the diversity metric is effectively preserved in both scenarios. However, this approach still lacks consideration of certain fitness-related information and tends to select mostly outlier individuals for migration, which is not particularly beneficial.

For S4 and S5, the common pool approach performs better. These strategies benefit from a global perspective when combining fitness and diversity scores or assigning selection probabilities. The common pool approach ensures that the selection process is driven by the information from the full population, leading to more effective migration decisions.

In conclusion, the choice of migration source is strategy-independent. For a random strategy and those that focus solely on fitness or randomness, selecting one individual from each island is preferable, as it introduces diversity and promotes exploration. For strategies that rely on global information, such as diversity-based or hybrid approaches, the common pool approach is more effective.

5.4 Computation time

Table 7 presents the average computation time for each strategy using $K = 8$. The results show that the computation time does not differ significantly between strategies, with the exception of S7, which requires additional time to perform clustering. The simplest strategy, S0 (random selection), had the lowest computation time. The differences between the other strategies (S1-S6) were within the range of standard deviation, indicating that the selection process itself is not a significant contributor to the overall computation time.

Table 7. Average computational time (in seconds) for different migrant selection strategies.

Strategy	TSP	BBOB	Strategy	TSP	BBOB
S0	**45.5** ± 0.7	**62.1** ± 1.0	**S4**	48.6 ± 1.0	65.7 ± 1.5
S1	45.9 ± 1.0	63.3 ± 1.3	**S5**	49.1 ± 1.3	67.2 ± 1.4
S2	47.2 ± 1.1	65.8 ± 1.2	**S6**	49.4 ± 1.2	67.6 ± 1.7
S3	47.8 ± 0.8	64.4 ± 1.1	**S7**	51.7 ± 1.5	70.2 ± 1.8

Furthermore, the experiments showed that the computation time for each strategy was largely independent of the value of K, with the exception of S7. This suggests that the selection of migrants is not a computationally intensive process and does not significantly impact the overall runtime of the algorithm. Therefore, in practical use, one can focus on selecting the most effective migration strategy without worrying about its computational cost.

6 Conclusion

This paper investigates the impact of migration strategies on the performance of island-based metaheuristics, with a focus on the selection of individuals for migration. We propose and evaluate several strategies, ranging from simple random selection to hybrid approaches that balance fitness and diversity. The results demonstrate that the choice of migration strategy significantly influences the quality of the solutions found.

A key finding is that increasing the number of migrants generally improves performance, but the benefits decrease as the number of them grows. Migrating a moderate number of individuals is sufficient to achieve strong performance, while avoiding the potential downsides of the excessive migration (e.g. homogenization of the populations). Strategies that combine fitness and diversity consistently outperform simpler approaches, highlighting the importance of balancing these two factors.

In conclusion, this study highlights the importance of carefully designing migration strategies in island-based metaheuristics, and provide practical guidance for improving the effectiveness of island-based models in solving complex optimization problems. In future research, we plan to perform comprehensive benchmarking of the proposed approach against state-of-the-art metaheuristic algorithms, and extend the algorithm evaluation to a wider set of metaheuristics, higher number of islands, and other problem domains. Another direction of research is the exploration of more sophisticated migration strategies, for example self-adaptive migration or frequency-based fitness assignment [16].

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. Alba, E., Tomassini, M.: Parallelism and evolutionary algorithms. IEEE transactions on evolutionary computation **6**(5), 443–462 (2002)
3. Bäck, T., Schwefel, H.P.: An overview of evolutionary algorithms for parameter optimization. Evolutionary computation **1**(1), 1–23 (1993)
4. Cantú-Paz, E.: Migration policies, selection pressure, and parallel evolutionary algorithms. Journal of heuristics **7**, 311–334 (2001)
5. Dorigo, M., Birattari, M., Stutzle, T.: Ant colony optimization. IEEE computational intelligence magazine **1**(4), 28–39 (2006)
6. Duarte, G., Lemonge, A., Goliatt, L.: A dynamic migration policy to the island model. In: 2017 IEEE congress on evolutionary computation (CEC). pp. 1135–1142. IEEE (2017)
7. 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)
8. Harada, T., Alba, E.: Parallel genetic algorithms: a useful survey. ACM Computing Surveys (CSUR) **53**(4), 1–39 (2020)
9. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of International Conference on Neural Networks. vol. 4, pp. 1942–1948. IEEE (1995)
10. 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)
11. Lin, W.Y., Hong, T.P., Liu, S.M., Lin, J.H.: Revisiting the design of adaptive migration schemes for multipopulation genetic algorithms. In: 2012 Conference on Technologies and Applications of Artificial Intelligence. pp. 338–343. IEEE (2012)

12. Luque, G., Alba, E.: Parallel genetic algorithms: Theory and real world applications, vol. 367. Springer Science & Business Media (2011)
13. Reinelt, G.: TSPLIB—A traveling salesman problem library. ORSA journal on computing **3**(4), 376–384 (1991)
14. 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)
15. Talbi, E.: Metaheuristics: From design to implementation. John Wiley & Sons google schola **2**, 268–308 (2009)
16. Weise, T., Wan, M., Wang, P., Tang, K., Devert, A., Yao, X.: Frequency fitness assignment. IEEE Transactions on Evolutionary Computation **18**(2), 226–243 (2013)
17. 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)
18. Żychowski, A., Yao, X., Mańdziuk, J.: Diversity-driven cooperating portfolio of metaheuristic algorithms. In: [under review] (2025)
19. Żychowski, A., Yao, X., Mańdziuk, J.: Migration timing in hybrid island-based metaheuristic algorithms. In: International Conference on Artificial Intelligence and Soft Computing [accepted] (2025)