Tuning multi-objective multi-population evolutionary models for high-dimensional problems: The case of the migration process

Juan Carlos Gómez-López, Manuel Rodríguez-Álvarez, Daniel Castillo-Secilla, Jesús González

S0925-2312(25)01303-7
https://doi.org/10.1016/j.neucom.2025.130631
NEUCOM 130631
Neurocomputing
15 April 2024
21 May 2025
27 May 2025

Please cite this article as: J.C. Gómez-López, M. Rodríguez-Álvarez, D. Castillo-Secilla et al., Tuning multi-objective multi-population evolutionary models for high-dimensional problems: The case of the migration process, *Neurocomputing* (2025), doi: https://doi.org/10.1016/j.neucom.2025.130631.

This is a PDF file of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability, but it is not yet the definitive version of record. This version will undergo additional copyediting, typesetting and review before it is published in its final form, but we are providing this version to give early visibility of the article. Please note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

© 2025 Published by Elsevier B.V.



Highlights

Tuning Multi-objective Multi-population Evolutionary Models for High-dimensional Problems: The Case of the Migration Process*

Juan Carlos Gómez-López, Manuel Rodríguez-Álvarez, Daniel Castillo-Secilla, Jesús González

- A novel elitist probability-based migration policy.
- A new score metric to cluster statistically similar data and estimate effect size.
- An approach to find an optimal population partitioning scheme to get quality solutions efficiently.
- The number of migrants and the migration interval do not determine the procedure's efficacy.
- However, the procedure's efficiency is improved at high and low values, respectively.

Tuning Multi-objective Multi-population Evolutionary Models for High-dimensional Problems: The Case of the Migration Process

Juan Carlos Gómez-López^{a,*}, Manuel Rodríguez-Álvarez^a, Daniel Castillo-Secilla^b, Jesús González^a

^aDepartment of Computer Engineering, Automation, and Robotics, CITIC, University of Granada, Spain ^bFujitsu Technology Solutions S.A., CoE Data Intelligence, Camino del Cerro de los Gamos, 1, Pozuelo de Alarcón, Madrid, Spain.

Abstract

Multi-objective multi-population evolutionary procedures have become one of the most outstanding metaheuristics for solving problems characterized by the *curse of dimensionality*. A critical aspect of these models is the migration process, defined as the exchange of individuals between subpopulations every few iterations or generations, which has typically been adjusted according to a set of guidelines proposed more than 20 years ago, when the capacity to deal with problems was significantly less than it is today. However, the constant increase in computational power has made it possible to tackle today's complex real-world problems of great interest more plausibly, but with larger populations than before. Against this background, this paper aims to study whether these classical recommendations are still valid today, when both the magnitude of the problems and the size of the population have increased considerably, considering how this adjustment affects the performance of the procedure. In addition, the increase in the population size, coupled with the fact that multi-objective optimization is being addressed, has led to the development of a novel elitist probabilistic migration strategy that considers only the Pareto front. The results show some interesting and unexpected conclusions, in which other issues, such as the number of subpopulations or their size, should be considered when fitting multi-population models. Furthermore, some of the previously mentioned classical recommendations may not be well-suited for high-dimensional problems.

Keywords: Multi-objective Optimization, Multi-population Models, Evolutionary Algorithms, Migration Process, Energy-aware Computing, Feature Selection

1. Motivation

Over the years, many metaheuristics have been proposed to solve problems characterized by the *curse of dimensionality* [1], in which evaluating each possible solution is unmanageable with the current computational power. One metaheuristic that has gained the most recognition from the scientific community is population-based evolutionary procedures [2]. Here, a set of individuals evolves over generations to reach a solution as close as possible to the global optima.

An important consideration in the design of an evolutionary method is the type of optimization it addresses. As noted in [3], evolutionary algorithms have shown promising effectiveness in solving complex problems through multi-objective optimization, in which two or three conflicting objectives are optimized simultaneously. Therefore, the goal of the evolutionary procedure in this type of optimization is to identify a set of solutions that are better than the rest but have different satisfaction

daniel.castillosecilla@fujitsu.com (Daniel Castillo-Secilla), jesusgonzalez@ugr.es (Jesús González) trade-offs for each objective. These solutions are called Paretooptimal solutions [4].

On the other hand, regarding the population type, evolutionary procedures broadly fall into single-population and multipopulation models. Single-population models offer simplicity because only one population evolves over generations. Although this approach is widely used and has yielded good results for large-scale optimization problems [5, 6], its primary drawback is that it explores only one region of the search space, which is conditioned by the generation of the initial population [7, 8]. Multi-population models emerged to address this limitation, enabling simultaneous exploration of several search space regions and thus balancing intensification and diversification. For instance, [9] proposes Multiple Populations for Multiple Objectives (MPMO), a coevolutionary technique that offers a straightforward method to solve multi-objective problems by assigning each population to a single objective. However, most multi-population models are based on the island-based scheme, where multiple subpopulations evolve independently [10–15]. This scheme is intrinsically highly parallelizable [16] because subpopulations exchange individuals only during migration, which has allowed it to achieve optimal efficiency on large-scale problems [17].

In fact, one of the keys to the proper evolution of a multipopulation model is the migration process, defined as the move-

^{*}This research has been supported by grants PID2022-137461NB-C31 and PID2021-123930OB-C22, funded by MCIN/AEI/10.13039/501100011033 and by *ERDF, a way of making Europe.*

^{*}Corresponding author

Email addresses: goloj@ugr.es (Juan Carlos Gómez-López),

manolo@ugr.es (Manuel Rodríguez-Álvarez),

Preprint submitted to Neurocomputing

ment of individuals between subpopulations. Migration aims to maintain diversity in the search space to obtain quality solutions. Specifically, the hyperparameters that control the migration process in multi-population models are:

- *Migration interval* (*m_i*). Also known as *period* in many studies, it indicates the number of generations that must elapse between each migration process. Although it is usually set to a fixed value, some studies have considered dynamically adjusting it during the search [18–20].
- *Migration policy* (m_p) . It defines how migrants are selected from the source subpopulation and how they will be integrated into the destination.
- *Migration rate* (*m_r*). Also known as *grain*, it indicates the number of migrants each subpopulation sends when migration occurs.
- *Migration topology* (*m*_t). It determines how the different subpopulations are connected.
- *Number of subpopulations* (*n_{sp}*). Number of subpopulations that run the evolutionary procedure.
- *Subpopulation size* (*s_{sp}*). Number of individuals for each one of the subpopulations.

Historically, these hyperparameters have been adjusted considering studies such as [21-24], in which a series of guidelines were proposed. Although these works were published about 20 years ago, their indications are still being used nowadays. However, a significant issue should be considered: thanks to the exponential increase in computing power, current real-life complex problems of great interest, such as resource scheduling [25, 26], Feature Selection (FS) in high-dimensional spaces [27-30], system design [31-33], classifier optimization [34-37], intelligent transportation [38, 39], or network planning [40-42], can be tackled much more plausibly now than more than 20 years ago. In fact, multi-population models have been applied to this type of problems [43-46], following the recommendations of [21–24] but evolving a significantly higher number of individuals to maintain diversity over the search space [8, 47-50]. With this in mind, the following question arises: Are these guidelines still valid today, when the complexity of the problems and the size of the populations needed to find adequate solutions have increased considerably?

Related to the previous question, an equally relevant one arises: How does the adjustment of the migration hyperparameters affect the efficiency of the procedure? Usually, the quality of the final results is first considered. However, neither the execution time nor the energy consumption of the proposal should be neglected. In a world of limited resources and a climate change of increasing concern, the development of energyefficient approaches is essential [51, 52]. In this sense, the problem of fitting the migration hyperparameters for large populations should be tackled not only to obtain quality solutions but also to care about the execution time or the energy consumption through efficient computation. Finally, this increase in the population size, along with the adoption of multi-objective optimization to adequately address complex problems, has led to the need for a novel migration policy that considers only the Pareto optimal individuals, to favor selection pressure while reducing the number of potential migrants, which should lead to more efficient computation.

After this introduction, Section 2 presents the state-of-theart of migration adjustment for multi-population models and discusses parametric and non-parametric tests to detect statistically significant differences between the sample distributions of the results of evolutionary procedures. It also introduces the concept of effect size as a test for estimating the magnitude of such differences. These statistical methods are essential for properly tuning the migration process. Then, Section 3 introduces a novel elitist probability-based migration policy, specifically designed for multi-objective, multi-population evolutionary algorithms with large populations, which aims to increase the selection pressure towards the best solutions while reducing the number of selected migrants. The experimental study designed to fine-tune the migration process hyperparameters is discussed in Section 4, and since a huge amount of results is expected from running the evolutionary procedure with many different migration hyperparameter configurations, Section 4.5 presents a new procedure, supported by both parametric and non-parametric statistical tests, for comparing multiple results sample distributions for a given metric. Next, Section 5 applies this experimental study to optimize the migration process of a parallel multi-objective, multi-population evolutionary wrapper that solves an FS problem for several high-dimensional datasets, and finally, Section 6 concludes the paper.

2. State-of-the-art

After outlining the motivation behind this work, Section 2.1 presents the most relevant state-of-the-art related to the migration process. Furthermore, given that a study on optimizing migration hyperparameters for large populations requires robust statistical analysis, Section 2.2 outlines the most relevant literature on this matter.

2.1. Migration process

As previously mentioned, one of the keys in multi-population models is the migration process, so this section outlines issues such as the historical recommendations for adjusting the migration hyperparameters, recent studies that have endeavored to shed light on the subject of their adjustment, and, finally, an analysis of the suitability of these recommendations for large populations.

2.1.1. Classical recommendations

The first relevant approaches to adjustment of the migration process can be attributed to Cantú-Paz in the late 1990s [21–23, 53–55]. These studies aimed to provide a set of recommendations for correct adjustment of the migration process in multipopulation models since, up to that point, there was a proliferation of papers with expensive experiments, making it difficult to

establish certain generalization criteria. Cantú-Paz showed that an appropriate adjustment of the migration hyperparameters affected the final solution.

On the one hand, Cantú-Paz stated that a degradation in accuracy could occur if the different subpopulations comprised a significantly low number of individuals [54], showing the importance of s_{sp} . The author also discussed that both n_{sp} and s_{sp} had historically been given less consideration than other factors, such as the number of migrants and the migration interval. Therefore, it is expected that the population partitioning scheme (n_{sp} , s_{sp}), which usually involves the distribution of n_i individuals into n_{sp} subpopulations of size s_{sp} following the Tanese scheme [56], could be the key aspect in the adjustment of the migration process.

On the other hand, Cantú-Paz also focused on migration policies [55], specifically on *random-random* and *best-worst*, with the first term of this terminology indicating how migrants are selected in the source subpopulation, while the second one describing how the individuals to be eliminated in the destination subpopulation are chosen to accommodate the new ones. Thus, for the *random-random* policy, the individuals selected in the source subpopulation and those removed in the destination subpopulation are randomly selected to provide greater genetic diversity. In contrast, for the *best-worst* policy, the worst individuals in the destination subpopulation are replaced by the best individuals in the source. Cantú-Paz concluded that the *bestworst* policy was the best option since the selection pressure increased with the consequent acceleration in algorithm convergence.

However, m_i and m_r were not studied as thoroughly by Cantú-Paz, although he stated that 0.1 could be an appropriate value for m_r [22]. In this sense, studies such as [18, 57, 58] can be highlighted, although their results vary considerably. On the one hand, [18, 58] suggest that m_r should be set at a relatively low value (as suggested by Cantú-Paz in [22]) and m_i at a moderate value (10–30 generations). In contrast, [57] found that the influence of m_i and m_r is residual when the population is large enough, also in agreement with the conclusions of Cantú-Paz [54].

Subsequently, Tomassini presented one of the most relevant experimental works on migration tuning [24], in which some guidelines for migration hyperparameters adjustment were established. Here, classical problems such as *Even-Parity-4* or *Artificial Ant on the Santa Fe Trail* were solved, with populations of 100, 500, or 2 500 individuals distributed into up to 50 subpopulations. In summary, the main conclusions reached were as follows:

- The distribution of individuals in isolated subpopulations [53], in which migration did not occur at any time, provided better results than using a single population.
- Including migration achieved better solutions, mainly due to the injection of individuals from other subpopulations, which improved search diversity.
- The adopted topology was not a fundamental factor, although the best results were obtained with the *Ring* topol-

ogy, resulting in significant computational savings.

- Finally, a study was performed to adjust the values of m_i and m_r through a combination of different values. In this case, 5 subpopulations with 100 individuals each were used. Basically, it was concluded that m_i should take 5 or 10, while m_r should be set to 0.1 or 0.15. To a greater or lesser extent, this follows the trend previously reported in [18, 58].
- Furthermore, despite not being the best option, it could also be observed that if m_i increased, m_r should increase accordingly. If migrations occurred sporadically, more individuals were injected to promote genetic diversity. In contrast, if m_i was very low, causing migrations to occur very frequently, m_r should also be decreased accordingly to prevent the algorithm from converging prematurely. Otherwise, many individuals would migrate between subpopulations too quickly, causing all of them to start sharing the same individuals after only a few generations. This lack of diversity would negatively affect the quality of the final solutions, stagnating the algorithm at local optima.

2.1.2. Recent studies

Although most studies set their migration processes based on the guidelines mentioned above, some works have tried to contribute their grain of sand to this subject. They can be split into two types depending on the population sizes they use.

On the one hand, works such as [59-61] can be highlighted, in which some novel metaheuristics are used for small populations. Here, m_i , m_r , n_{sp} , and s_{sp} are analyzed, with s_{sp} ranging from 60 to 100 and n_{sp} from 2 to 15, by evaluating a set of IEEE-CEC2005 unimodal and multimodal benchmark functions with different complexity. Although the trend is toward tuning m_i to intermediate values and m_r to relatively low values, as Tomassini noted, it seems that the adjustment of the migration process depends on the metaheuristic and the functions to be optimized. Anyway, due to the small size of the population and the type of problem addressed in these studies, these conclusions should not be extrapolated for large population setups. Of particular interest for m_t is [62], which divides a population of 100 individuals into 1 to 12 subpopulations to determine the relationship between migration topology and problem structure. Several topologies (including Ring, Fully connected, Grid, and others) are examined, and it is concluded that none is inherently the best in all scenarios because the relationship between the population structure and the problem is highly complex. Finally, in relation to m_p , [63] agrees with the findings of Cantú-Paz and suggests that the best-worst policy is the most appropriate approach.

On the other hand, concerning medium or large-sized populations, the work [64] can stand out, in which 2048 individuals are distributed into 10 subpopulations. Here, the best value for m_i is analyzed, concluding that small intervals should be avoided, although the remaining hyperparameters are not studied in depth. However, [65] states that a large number of indi-

viduals should migrate in short migration intervals, which conflicts with the recommendation of [64]. This could be caused by the fact that each hyperparameter is studied in isolation, leading to contradictory conclusions. Concerning m_t , Ruciński addressed the study of this hyperparameter for large populations in [66], concluding that the *Ring* topology and, more specifically, two versions of it (*Ring* + 1 + 2 and *Ring* + 1 + 2 + 3) provided the best results. Finally, in relation to m_p , the study conducted in [67], in which 2 000 individuals are divided into 10 subpopulations, also supports the conclusions derived from the work of Cantú-Paz for this hyperparameter, indicating that *best-worst* is the most appropriate as it is consistent with the natural selection mechanism for survival of the fittest.

2.1.3. Adjustment suitability for multi-objective problems and large populations

It should be noted that migration hyperparameters have not received equal attention in the literature. Most studies have focused on analyzing m_i and m_r , leaving n_{sp} and s_{sp} in a secondary role, a limitation already discussed in Section 2.1.1. However, there seems to be a consensus on the importance of properly tuning m_p and m_t .

Regarding migration policies, although several alternatives have been proposed [68-70], many are derived to some extent from the best-worst strategy. This observation supports the early recommendation of Cantú-Paz in [55], who concluded that this migration policy improves selection pressure, thus accelerating the convergence of algorithms. However, it is important to note that the effectiveness of the best-worst policy is highly problem-dependent. In the case of multi-objective optimization, it is common to find a large number of non-dominated solutions along the Pareto front. Consequently, many studies have adapted the *best-worst* approach to these problems by selecting migrants exclusively from the Pareto front. Several works [71-76] have proposed implementations that emphasize this elitist strategy. This method is especially attractive for large populations, as it increases selection pressure while reducing the number of candidate individuals.

As for the topology (m_t), the studies by Tomassini [24] and Ruciński [66] suggest that further exploration may be unnecessary. *Ring* topologies and their variants have shown strong performance in a wide range of problems [77–79], while more complex topologies have not demonstrated clear advantages despite their higher computational cost.

Finally, there is a notable lack of experimental studies that jointly consider the number and size of subpopulations, along with other migration hyperparameters. A comprehensive evaluation of their interactions, especially in the context of large populations, remains largely unexplored.

2.2. Statistical analysis

Search strategies such as evolutionary procedures are known to exhibit stochastic behavior. Furthermore, optimization of migration hyperparameters and experimental analysis of their interactions will generate a significant amount of data, thus requiring a robust statistical study to provide useful information. Although most studies originally compared their results with those of other approaches using the average of multiple runs, this methodology was considered unfair, so works such as [80-82] proposed a comparison using various parametric tests (ANOVA, t-test, etc.) based on mean and variance. These tests provide a high level of statistical rigor when comparing different results sample distributions, although their use is quite restrictive. Specifically, all results sample distributions must satisfy three conditions: independence, normality, and homoscedasticity.

The most commonly used statistical tests to assess normality are the Shapiro-Wilk and the Kolmogorov-Smirnov tests. The former is recommended when the number of samples is less than 50, while the Kolmogorov-Smirnov test is indicated otherwise. Once normality has been confirmed, homoscedasticity can be confirmed using the Barlett test [83, 84].

On the other hand, since the sample distributions of the results may be neither normal nor homoscedastic, it is particularly interesting to use non-parametric tests [85], which analyze the median of the results instead of their mean and variance. Among non-parametric tests, the Kruskal-Wallis test is one of the most relevant methods [86].

In either case, both parametric and non-parametric tests yield a *p*-value, which, if less than a certain significance level α , indicates rejection of the null hypothesis, i.e., that the compared results present statistically significant differences with a confidence level of $(1 - \alpha) \times 100\%$. For example, rejecting the null hypothesis with a *p*-value less than $\alpha = 0.05$ means that the results are different at a confidence level of 95%.

Although a *p*-value less than α reveals a statistically significant difference among several results distributions, it does not indicate the magnitude of that difference. Therefore, only a part of the statistical study is carried out, as detailed in [87–89]. In this context, a measure or index called effect size arises, which can be considered equal to or even more relevant than the *p*-value. The effect size can be used to quantify the difference between two result distributions once the *p*-value reveals that such a difference exists. Although the effect size could be considered to have been introduced more than 100 years ago [90], it was not until the late 1960s that J. Cohen standardized it with his well-known Cohen's *d* index [91].

The study of effect size has gained special relevance in countless studies, mainly due to the valuable information it provides. This fact is exemplified in [90], in which it is mentioned that 19 scientific journals of different natures require a study of the effect size to publish papers. Over the years, dozens of indexes have been proposed to quantify the effect size [92, 93], among which Hedge's g, Pearson's r, or Glass's Δ stand out. However, to date, Cohen's d, which standardizes such a practical difference or effect size as small (d = 0.2), medium (d = 0.5), large (d = 0.8), and very large ($d \ge 1.3$), remains one of the most widely used.

3. Elitist probabilistic migration: A novel migration policy

As discussed above, the *best-worst* policy has been widely used in countless studies. Thus, it is an appropriate choice for



Figure 1: Flow chart of the new elitist probabilistic migration policy.

the m_p hyperparameter. In addition, since multi-objective optimizations of complex problems are addressed in this work and the population consists of a considerable number of individuals, the aim is to select the best migrants from the source subpopulation P_i from its Pareto front $\mathcal{F}_0(P_i)$ to favor elitism and reduce the number of potential migrants. Figure 1 shows the flow chart of this novel migration policy, which is explained in more detail below.

Typically, hyperparameter m_r , which remains constant throughout the evolutionary process, determines the number of migrants chosen from the source subpopulation. However, a policy based on selecting individuals from $\mathcal{F}_0(P_i)$ depends on its variable size in each generation. To address this and streamline the process, a new hyperparameter m_l is proposed to replace m_r . It is defined as the likelihood that one Pareto-optimal solution becomes a migrant. Since all $\mathcal{F}_0(P_i)$ members have an m_l probability of being selected as migrants, the average number of selected migrants (n_m) depends on both the generation and the subpopulation. It can be estimated as:

$$n_{m_{ij}} = m_l \times |\mathcal{F}_{0_j}(P_i)| \tag{1}$$

in which $|\mathcal{F}_{0_j}(P_i)|$ is the number of Pareto-optimal solutions of the *i* – *th* subpopulation at generation *j*.

It could be the case that no migrant is selected, either due to a low value of m_l or $|\mathscr{F}_{0_j}(P_i)|$. In this improbable scenario, a randomly selected solution from the Pareto front is chosen as a migrant to avoid performance degradation caused by running the subpopulations in isolation, as discussed in Section 2.1.1.

Finally, when migrants arrive at the destination subpopulation, they are added to the population. Later, the survival selection mechanism keeps the best s_{sp} individuals for the next generation, thus eliminating the worst $n_{m_{ij}}$ from P_i .

4. Experimental study design to optimize the migration process

After presenting the novel elitist migration policy, this section discusses the proposed methodology for optimizing the migration hyperparameters of multi-objective, multi-population evolutionary procedures with large populations.

4.1. Problem and evolutionary procedure choice

Before applying the methodology proposed in the following subsections, it is important to define two points. On the one hand, a current complex real-world problem of great interest involving large populations should be selected. In particular, several instances of this problem should be addressed in order to generalize the results obtained. This approach should avoid any potential bias that might arise from focusing on a single instance. On the other hand, a multi-objective, multi-population evolutionary method, which has proven to be effective in solving this type of problem, should be chosen. Furthermore, since solving complex problems is a resource-intensive task, the chosen method should be a parallel approach to achieve efficient computation.

4.2. Hyperparameters analyzed

Note that not all hyperparameters that control the migration process require analysis. As discussed in Section 2.1.3, it is not recommended to further investigate the hyperparameters m_t and m_p . Regarding the former, an *Uni-directional Ring* topology has been selected because it is a good choice for procedures involving large populations, as it achieves an appropriate balance between diversification and intensification while presenting a low complexity. On the other hand, the novel elitist probabilistic migration policy presented in Section 3 has been chosen for m_p .

With this in mind, the hyperparameters under consideration are m_i , m_l , n_{sp} , and s_{sp} , as there is currently no existing work that addresses the study of each of them and their interactions when evolving large populations.

4.3. Metrics considered

Several metrics are obtained from each execution to allow analysis of the behavior of the evolutionary procedure with respect to its hyperparameter configuration. Concerning its effectiveness, the objectives it optimizes should be taken into account. Since multi-objective problems are solved, each run provides several Pareto optimal solutions. Thus, the average value of each objective is used for the Pareto front solutions, since in this way the average quality of the Pareto front solutions can be estimated. In addition, execution time and energy consumption should also be measured to estimate its efficiency.

4.4. Relevance of the population partitioning scheme

Once all the aspects regarding the migration hyperparameters under study, the problem, and the evolutionary procedure have been chosen, it is time to focus on the proposed steps to optimize the migration process for evolutionary processes with large populations. As discussed in Section 2.1.1, the accurate calibration of s_{sp} could be the most critical factor, while m_i and m_l should play a secondary role. Thus, the first step is to establish the set of possible values of n_{sp} as follows:

$$\mathcal{N}_{sp} = \left\{ n_{sp_j} : n_{sp_j} \in [1, n_{cores}] \cap \mathbb{N} \right\}$$
(2)

with n_{cores} being the number of physical CPU cores of the processor on which the multi-population model is executed. As stated in [94–96], the use of more threads/processes than the number of physical cores is not recommended for intensive calculations to avoid the efficiency degradation that occurs when several threads or processes demand the resources of a single core.

Next, in line with Tanese's recommendation outlined in Section 2.1.1, a population partitioning scheme can be defined for each subpopulation size n_{sp_i} as follows:

$$(n_{sp_i}, s_{sp_i}) : s_{sp_i} = \lceil n_i / n_{sp_i} \rceil \quad \forall n_{sp_i} \in \mathcal{N}_{sp}$$
(3)

On the other hand, the possible values for the migration interval (m_i) and the migration likelihood (m_i) can be defined, respectively, as follows:

$$\mathcal{M}_i = \left\{ m_{i_j} : m_{i_j} \in \mathbb{N} \right\} \tag{4}$$

$$\mathcal{M}_{l} = \left\{ m_{l_{i}} : m_{l_{i}} \in (0, 1) \right\}$$
(5)

Following Section 2.1.1, both too high m_i values and too low m_l values would lead to an isolated subpopulations-like behavior, in which migrations are practically residual. In contrast, too-low values of m_i would result in very frequent migration, while too-high values of m_l would result in a significant injection of migrants into the population, both leading to premature convergence.

Given the aforementioned considerations, while four hyperparameters are to be analyzed, only three independent variables are considered: m_i , m_l , and n_{sp} , given that s_{sp} is dependent on n_{sp} and n_i .

Once the sets of possible values for the three independent variables under consideration have been established, a grid search could be performed, evaluating a total of n_c different procedure configurations for a given problem instance:

$$n_c = |\mathcal{N}_{sp}| \times |\mathcal{M}_l| \times |\mathcal{M}_l| \tag{6}$$

in which $|\mathcal{N}_{sp}|$, $|\mathcal{M}_i|$, and $|\mathcal{M}_i|$ are the number of values for n_{sp} , m_i , and m_l , respectively. Furthermore, the procedure should be run n_e times with each different hyperparameter configuration to allow for robust statistical analysis, resulting in a total of $n_c \times n_e$ executions.

Since the objective is to analyze the influence of the population partitioning scheme on the procedure behavior, all data obtained by the grid search for each behavioral metric M concerning the same partitioning scheme (n_{sp_j}, s_{sp_j}) should be collected to form a sampling distribution of M for each different scheme (n_{sp_j}, s_{sp_j}) . Therefore, $|\mathcal{N}_{sp}|$ sampling distributions of $|\mathcal{M}_i| \times |\mathcal{M}_l| \times n_e$ samples can be obtained for each different

metric *M*. This substantial amount of data motivates the development of a score metric that allows for a complete sorting of multiple sampling distributions for a given metric, each one associated with a different procedure configuration. This score is described in the next section.

On the other hand, although the grid search obtains a value for all the behavioral metrics each time the multi-population procedure is run, since the goal is to evaluate the effectiveness of the procedure for each population partitioning scheme, only the metrics related to the objectives to be optimized are currently considered. Premature convergence can be easily detected by identifying the population partitioning schemes whose solutions quality deteriorates significantly, especially when subpopulations have few individuals (schemes with low s_{sp_j} values), mainly due to the lack of diversity in the subpopulations.

4.5. Multiple comparison of the procedure configurations

The grid search mentioned in the previous section generates a total of $n_c \times n_e$ values for each procedure behavior metric, which can be partitioned into n_d sampling distributions, each corresponding to a different configuration for a subset of hyperparameters. The need to rank all these sampling distributions in order to sort the different hyperparameter configurations has motivated the development of a score metric, which is inspired by that presented in [97].

Let \mathcal{M} be the set:

$$\mathcal{M} = \{ M_{C_i} : i \in \mathbb{Z} \cap [0, n_d) \}$$
(7)

with M standing for one of the different metrics of the procedure behavior, while C_i denotes a concrete hyperparameter combination out of n_d possible configurations.

The first step when comparing two sampling distributions of a given metric M, M_{C_i} and M_{C_j} , obtained by two different hyperparameter configurations, C_i and C_j , is to decide whether to apply a parametric or a non-parametric test. Parametric tests can be used only if the two distributions are normal and homoscedastic, while non-parametric tests can be applied in the remaining cases. In this sense, the Shapiro-Wilk test is applied to M_{C_i} and M_{C_j} if the number of samples is less than 50. Otherwise, the Kolmogorov–Smirnov test is used. If both distributions are normal, the Bartlett test is used to detect homoscedasticity. Then, the ANOVA test is applied to M_{C_i} and M_{C_j} if both are normal and homoscedastic. Otherwise, the Kruskal-Wallis test is used.

All the aforementioned statistical tests return a *p*-value estimating the suitability of the samples to the null hypothesis. This *p*-value is then compared to a certain significance level, α , which causes the null hypothesis to be rejected if the *p*-value is less than α . The lower the α , the greater the confidence in rejecting the null hypothesis. A typical value of α is 0.05.

Once the corresponding statistical tests are applied, the comparison of distributions M_{C_i} and M_{C_j} is defined by the following relations:

$$M_{C_i} \prec M_{C_j} \iff p_{C_i,C_j} \le \alpha \land \bar{m}_{C_i} \prec \bar{m}_{C_j}$$
 (8)

$$M_{C_i} \approx M_{C_j} \quad \Leftrightarrow \quad p_{C_i,C_j} > \alpha$$
(9)

$$M_{C_i} \le M_{C_j} \quad \Leftrightarrow \quad M_{C_i} \prec M_{C_j} \quad \lor \quad M_{C_i} \approx M_{C_j}$$
 (10)

in which p_{C_i,C_j} is the *p*-value obtained by comparing the distributions M_{C_i} and M_{C_j} , \overline{m}_{C_i} and \overline{m}_{C_j} denote the means of M_{C_i} and M_{C_j} , respectively, and the \prec operator applied to scalar values such as the distribution means indicates that the left-hand expression is better than the right-hand one, i.e., it means lower for minimization and higher for maximization objectives. In fact, the set \mathcal{M} of the sampling distributions of a metric M obtained for different hyperparameter configurations can be partially sorted as follows:

$$M_{C_i} \le M_{C_i} \iff i \le j, \ \forall M_{C_i}, M_{C_i} \in (\mathcal{M}, \le)$$
(11)

That is, the lower *i*, the better the procedure behavior for metric *M*. On the other hand, the equivalence relation defined in Equation (9) allows partitioning \mathcal{M} into clusters of configurations that produce statistically similar behavior for metric *M*:

$$\frac{\mathscr{M}}{\approx} = \{ \operatorname{sim}(M_{C_i}) = [s_a] : s_a \in \mathscr{M} \}$$
(12)

Thus, the rank of a hyperparameter configuration C_i for a metric M can be calculated as the average of the positions in \mathcal{M} of all M_{C_i} that are statistically similar to M_{C_i} , i.e.:

$$r(M_{C_i}) = \frac{\sum_{M_{C_j} \in \sin(M_{C_i})} j}{|\sin(M_{C_i})|}$$
(13)

Then, the score of a hyperparameter configuration C_i for a metric M is obtained by dividing its rank by the number of hyperparameter configurations:

$$s(M_{C_i}) = \frac{r(M_{C_i})}{|\mathcal{M}|} \tag{14}$$

Although the rank defined in Equation (13) allows for the comparison of hyperparameter configurations for a given metric (the lower the rank, the better the configuration), the score proposed in Equation (14) provides better interpretability because its values are always defined in [0, 1), independently of the number of hyperparameter configurations compared.

Finally, for those configurations that yield statistically different results for a given metric, i.e., a different score, the effect size should be calculated to estimate the magnitude of such differences. The effect size obtained should correlate with the score differences obtained. Specifically, Cohen's *d* has been used in this work.

4.6. Optimization of the population partitioning scheme

On the one hand, the methodology outlined in Section 4.4 should show a tendency for the multi-population procedure to converge prematurely as the subpopulation size (s_{sp}) decreases because the number of subpopulations (n_{sp}) increases, mainly due to the lack of diversity in the subpopulations. Thus, it is

likely that there is a minimum threshold for the subpopulation size $(s_{sp_{min}})$ to avoid the problem of premature convergence. This section proposes an approach to estimate such a threshold.

On the other hand, avoiding the premature convergence issue by updating the population partitioning schemes (n_{spj}, s_{spj}) to (n_{spj}, s_{spmin}) for all $s_{spj} < s_{spmin}$ would cause the total number of individuals to increase as n_{sp} increases, which may not improve the results for large values of n_{sp} , although it would require more computational resources. Therefore, there should be a maximum threshold for the number of subpopulations (n_{spmax}) that allows the procedure to find appropriate solutions while minimizing its computational time and energy consumption. Thus, the optimal partitioning scheme would be (n_{spmax}, s_{spmin}) .

In light of the considerations above, the first step should be to identify $s_{sp_{min}}$, which should be between two successive population partitioning schemes, (n_{sp_j}, s_{sp_j}) and $(n_{sp_{j+1}}, s_{sp_{j+1}})$, with $n_{sp_j} > n_{sp_{j+1}}$ and $s_{sp_j} < s_{sp_{j+1}}$, in which the degradation begins to be noticeable, i.e., premature convergence should be observed when using a subpopulation size of s_{sp_j} , but not when using $s_{sp_{j+1}}$. Therefore, the minimum value of s_{sp} that maintains the quality of the solutions, $s_{sp_{min}}$, should be in the interval $[s_{sp_j}, s_{sp_{j+1}}]$. This interval must be searched with n_{sp_j} subpopulations because scheme (n_{sp_j}, s_{sp_j}) has shown a clear tendency to converge prematurely. The increase in the subpopulation size from s_{sp_j} to $s_{sp_{min}}$ is expected to prevent premature convergence when using n_{sp_j} subpopulations.

As an exhaustive search in the $[s_{sp_j}, s_{sp_{j+1}}]$ interval is rather costly, a binary search-based approach has been proposed, which features several modifications regarding the original algorithm [98]. However, before proceeding with a detailed examination of this approach, it is important to highlight four crucial aspects:

- The purpose of determining the value of $s_{sp_{min}}$ is to identify a minimum threshold for the subpopulation size that enables the maintenance of solution quality, which is measured by the objectives optimized by the evolutionary procedure, so their scores are considered to guide the search. Nevertheless, sample distributions of the other efficiencyrelated metrics (computing Time, *T*, and consumed Energy, *E*) are also obtained because although not used now, they will be used later to confirm the estimation of $n_{sp_{max}}$.
- Each time a new subpopulation size is evaluated, the scores of all the previously evaluated subpopulation sizes must be recalculated, since adding a new distribution of metric values affects the previous scores.
- Two score values are considered similar based on the following similarity criterion:

$$(M_{C_a}) \approx s(M_{C_b}) \Leftrightarrow |s(M_{C_a}) - s(M_{C_b})| < s_{th}$$
(15)

in which M_{C_a} and M_{C_b} are the sampling distributions of metric M for two different configurations C_a and C_b , respectively, in this case consisting of the procedure configured with two different values for s_{sp} (s_{sp_a} and s_{sp_b}), and

S

Algorithm 1: Procedure designed to find $s_{sp_{min}}$ within interval $[s_{sp_{low}}, s_{sp_{high}}]$. **1** Function findSspMin($s_{sp_{low}}, s_{sp_{high}}, w_{th}, \mathcal{M}_i, \mathcal{M}_l, n_e, n_{sp}$) **Input** : Lower extreme of the interval in which to search $s_{sp_{min}}$, $s_{sp_{low}}$ **Input** : Higher extreme of the interval in which to search $s_{sp_{min}}$, $s_{sp_{high}}$ **Input** : Threshold for the minimum width of $[s_{splow}, s_{sphigh}], w_{th}$ **Input** : Set of different values for m_i , \mathcal{M}_i **Input** : Set of different values for m_l , \mathcal{M}_l **Input** : Number of executions, n_e **Input** : Number of subpopulations, n_{sp} Output: Minimum subpopulation size $nScores \leftarrow 2$ 2 $(O_{0_{low}}, ..., O_{n_{obj}-1_{low}}, T_{low}, E_{low}) \leftarrow \texttt{gridSearch}(\mathcal{M}_{l}, \mathcal{M}_{l}, n_{e}, n_{sp}, s_{sp_{low}})$ 3 $(O_{0_{high}}, ..., O_{n_{obj}-1_{high}}, T_{high}, E_{high}) \leftarrow \texttt{gridSearch}(\mathcal{M}_i, \mathcal{M}_l, n_e, n_{sp}, s_{sp_{high}})$ 4 repeat 5 6 $s_{sp_{med}} \leftarrow \lceil s_{sp_{low}} + s_{sp_{high}} \rceil / 2$ $(O_{0_{med}},...,O_{n_{obj}-1_{med}},T_{med},E_{med}) \leftarrow \texttt{gridSearch}(\mathcal{M}_{i},\mathcal{M}_{l},n_{e},n_{sp},s_{sp_{med}})$ 7 $nScores \leftarrow nScores + 1$ 8 for $j \leftarrow 0$ to nScores - 1 do 9 $(s(O_{0_j}), ..., s(O_{n_{obj}-1_i})) \leftarrow$ Score of the n_{obj} objectives after $n_c \times n_e$ executions with $s_{sp} = s_{sp_j}$ 10 11 end if $s(O_{k_{med}}) \approx s(O_{k_{high}}) \quad \forall k \in \mathbb{Z} \cap [0, n_{obj})$ then 12 13 $(s_{sp_{high}}, O_{0_{high}}, ..., O_{n_{obj}-1_{high}}) \leftarrow (s_{sp_{med}}, O_{0_{med}}, ..., O_{n_{obj}-1_{med}})$ else $(s_{sp_{low}}, O_{0_{low}}, ..., O_{n_{obj}-1_{low}}) \leftarrow (s_{sp_{med}}, O_{0_{med}}, ..., O_{n_{obj}-1_{med}})$ d 14 15 16 end until $(s_{sp_{high}} - s_{sp_{low}} < w_{th})$ or $(s(O_{k_{low}}) \approx s(O_{k_{high}}) \quad \forall k \in \mathbb{Z} \cap [0, n_{obj}));$ 17 return s_{splow} 18 19 End

 s_{th} is a similarity threshold. As with the adjustment of α in parametric and non-parametric tests, an adjustment should be made according to the desired precision for the score similarity. In this sense, a value of 0.05 might be an appropriate choice.

• The evaluation of each possible value for $s_{sp_{min}}$ is quite costly. Specifically, the number of different procedure configurations evaluated for each population partitioning scheme are:

$$n_c = |\mathcal{M}_i| \times |\mathcal{M}_l| \tag{16}$$

which generates a total of $n_c \times n_e$ executions for each new subpopulation size to be analyzed. Therefore, the search could be stopped if the interval to be searched is too narrow, since the possible change in $s_{sp_{min}}$ is not worth the computational cost. This premature stopping criterion is regulated by the hyperparameter w_{th} , a threshold for the minimum width of the interval, which should be adjusted considering the total number of individuals (n_i) . This value should be low enough to achieve a balance between resource consumption and the desired precision for the $s_{sp_{min}}$ fit, but not too low, since analyzing very small intervals would not result in a notable change for the value of $s_{sp_{min}}$, despite the extensive number of executions required to perform a new grid search for each potential value for $s_{sp_{min}}$.

With this in mind, Algorithm 1 shows the pseudocode of the binary search approach proposed. The gridSearch function is responsible for performing the grid search outlined in Section 4.4 for a given population partitioning scheme and all the possible combinations of values for m_i and m_l , generating a sample distribution of $n_c \times n_e$ samples of each behavioral metric (the average of each objective O_k of the n_{obj} objectives for the solutions found by the procedure, the computation time and the energy consumed) for that partitioning scheme. Then, once the $s_{sp_{med}}$ value has been determined (Line 6), and the new grid search using $s_{sp_{med}}$ has been performed (Line 7), all *nScores* scores must be recalculated (Lines 9 to 11), considering each one of the subpopulation sizes analyzed for s_{sp} . Next, the scores of the sample distributions of all the objectives obtained for the upper extreme of the interval $(s_{sp_{high}})$ and at the midpoint $(s_{sp_{med}})$ are compared using Equation (15). If the scores for all objectives for the upper extreme of the interval are similar to those obtained for the other half of the interval, the upper half of the

interval is rejected. Otherwise, the lower half is discarded. The algorithmic complexity of the binary search is:

$$\mathcal{O}\left(\log_2\left((s_{sp_{j+1}} - s_{sp_j} - w_{th}) \times n_c \times n_e \times f(n_o, n_i)\right)\right) \quad (17)$$

Given the minimum interval size w_{th} and the fact that the algorithm discards half of the interval at each iteration, the maximum number of iterations is limited to $\log_2(s_{sp_{j+1}} - s_{sp_j} - w_{th})$ iterations, although it could stop earlier if the scores of all objectives for the two extremes of the interval are similar. At each iteration, the binary search performs $n_c \times n_e$ executions of the evolutionary procedure (see Equation (16)), whose complexity order depends on the number of objectives (n_o) and the number of individuals (n_i) . Since the complexity order of the evolutionary procedure depends on the concrete procedure, it is denoted as $f(n_o, n_i)$.

Once the binary search is complete, the next step is to check for the absence of premature convergence, given the value obtained for the $s_{sp_{min}}$ threshold. For this purpose, all population partitioning schemes proposed in Section 4.4 must be updated as follows:

$$s_{sp_k} = \begin{cases} s_{sp_k} & \text{if } s_{sp_k} > s_{sp_{min}} \\ s_{sp_{min}} & \text{otherwise} \end{cases}$$
(18)

Then, the grid search should be repeated for all partitioning schemes whose s_{sp_k} has been replaced by $s_{sp_{min}}$, to obtain new behavioral metrics for the updated schemes. Finally, the scores for all n_{obj} objectives optimized by the evolutionary procedure should be recalculated for all partitioning schemes to consider the newly obtained distributions. There should not be statistically significant differences for any distribution, indicating that the premature convergence issue has indeed disappeared.

Estimating $n_{sp_{max}}$ is straightforward. Since all updated partitioning schemes have made the evolutionary procedure yield statistically similar results, $n_{sp_{max}}$ should be n_{sp_j} , the number of subpopulations used by the binary search to find $s_{sp_{min}}$, i.e., the minimum value of n_{sp} among all schemes whose subpopulation size was updated to $s_{sp_{min}}$. Increasing n_{sp} beyond $n_{sp_{max}}$ has not improved the results, although it probably has required more computational resources. No further runs of the evolutionary procedure are needed to confirm $n_{sp_{max}}$, since all new population partitioning schemes have already been evaluated to validate $s_{sp_{min}}$. In fact, the grid search provides distributions of all behavioral metrics for each population partitioning scheme, as discussed earlier. The values of the objectives optimized by the evolutionary process have already been used to verify $s_{sp_{min}}$. Therefore, the only remaining task is to compute and compare the values of the efficiency-related metrics. Since all partitioning schemes with $n_{sp} > n_{sp_{max}}$ now have $s_{sp} = s_{sp_{min}}$, their scores for both computation time and energy consumed should be worse than those of $(n_{sp_{max}}, s_{sp_{min}})$.

4.7. Influence of the migration interval and the number of migrants

Once the optimal population partitioning scheme $(n_{sp_{max}}, s_{sp_{min}})$ has been identified for each problem instance, it

can be used to analyze the influence of m_i and m_l . Again, no further runs of the multi-population evolutionary procedure are required because all the combinations of \mathcal{M}_i and \mathcal{M}_l values have already been evaluated for the population partitioning scheme $(n_{sp_{max}}, s_{sp_{min}})$. However, the samples of every behavioral metric must now be split into different distributions, each corresponding to each possible (m_i, m_l) combination. Then, the scores of these new sample distributions can be calculated for the behavioral metrics, which allows an assessment of the influence of m_i and m_l on the procedure's behavior.

5. Experimental results

After presenting the proposed methodology for the migration hyperparameter optimization for evolutionary procedures with large populations, this section presents the experimental results for each previously described method.

5.1. Setup

In order to study the optimization of the migration process, it is first necessary to choose an appropriate problem that involves a sufficiently large population of individuals, as discussed in Section 4.1. Therefore, among all the problems mentioned in Section 1, a high-dimensional FS problem has been chosen, since it is of great importance nowadays, mainly due to the increasing size of current datasets. In addition, since several instances of the problem should be analyzed to obtain statistically robust results, three Brain-Computer Interface (BCI) Motor Imagery (MI) datasets corresponding to three anonymized subjects, coded 104, 107, and 110, and recorded in the BCI Laboratory at the University of Essex, will be processed [101]. The datasets comprise Electroencephalogram (EEG) signals corresponding to three movements: left hand, right hand, and feet. The number of samples of each class is balanced. All are split into training and test data, consisting of 178 samples and 3 600 features each. With this in mind, these datasets meet the requirements of this experimental study since the population size in this type of problem should be set to a value close to the dataset dimensionality to cover the search space appropriately, as stated in [47].

The next step is to choose an adequate population-based algorithm to solve FS problems. Wrapper methods are commonly used for such problems. Basically, they are based on a search strategy that uses an induction method to evaluate the potential solutions for the problem [102]. Specifically, the efficient parallel multi-objective multi-population evolutionary wrapper proposed in [16], which applies a Non-dominated Sorting Genetic Algorithm II (NSGA-II) and the k-NN classifier as the induction method, has been chosen because it has previously been successfully applied to the three Essex BCI datasets [16]. This wrapper dedicates each CPU core to managing the evolution of a different subpopulation, and performs the exchange of migrants between subpopulations asynchronously, i.e., each subpopulation can send its migrants to the destination one without stalling its execution flow, which achieves a higher efficiency than synchronous communication, in which all subpopulations

Table 1: Hyperparameter values setting for all the experiments.

Hyperparameter	Value
Number of executions (n_e)	30*
Crossover probability (p_c)	0.8
Mutation probability (p_m)	0.2
Independent mutation probability for each feature (p_{ind})	0.00028^{\pm}
Maximum number of generations performed (m_g)	1 000
Number of generations analyzed to detect convergence (g_a)	15
Intra-generation convergence threshold for the standard deviation of the current population Kappa index $(s_{k_{th}})$	0.1
Inter-generation convergence threshold for the average of the current population Kappa index (\bar{k}_{th})	0.01
Migration policy (m_p)	Elitist probabilistic migration
Migration topology (m_t)	Uni-directional Ring

A value in [20, 50] is indicated as adequate regardless of the study addressed, as proposed in [99].

[‡] 1 / *Number of features*, as proposed in [100].

Table 2: Cluster node configuration.

Number of CPUs	2
CPU	Intel Xeon Silver 4214
	12 cores/ 24 threads
	2 200 Mhz, 85 W
RAM	64 GB DDR4

must synchronize, so that the slowest subpopulation is the one that determines the execution flow of the others [103–105]. Furthermore, the wrapper also provides an adaptive stop criterion regulated by hyperparameters g_a , $s_{k_{th}}$, and \overline{k}_{th} [8], which evaluates the evolution of the quality of solutions. The wrapper has been configured according to the hyperparameter values listed in Table 1. For more information about the wrapper method or its hyperparameter configuration, the reader is referred to [16].

Two points about the wrapper should be emphasized in order to properly apply the methodology proposed in Section 4:

- The wrapper is implemented in Python using the DEAP library, which requires the subpopulation size to be a multiple of 4 for all NSGA-II-based algorithms [106]. Thus, all values of s_{sp} proposed for experimentation must satisfy this constraint.
- The two objectives optimized by the wrapper are the Kappa index and the number of selected features.

Furthermore, for the application of the statistical tests described in Section 4.5, an $\alpha = 0.5$ has been used, as this is a value commonly used for this type of study.

Finally, concerning the execution platform, all experiments have been performed on a homogeneous high-performance cluster running *Rocky Linux* 8.4, whose characteristics are de-

tailed in Table 2. In addition, each node is equipped with a *Vampire* smart energy meter [107], which measures active power, voltage, current, power factor, and other energy information every 200 ms.

5.2. Assessment of the population partitioning scheme influence on the wrapper behavior

As described in Section 4.4, the first step is to determine the influence of the population partitioning scheme on the wrapper effectiveness. Regarding n_{sp} , the set of possible values for this hyperparameter has been defined as:

$$\mathcal{N}_{sp} = \{2, 4, 8, 16, 24\} \tag{19}$$

i.e., it takes increasing powers of 2 until the number of CPU cores is reached ($n_{cores} = 24$). To define the population partitioning schemes, it is necessary to fix the total population size (n_i) beforehand since, as stated in Equation (3), each s_{sp_j} depends on both its corresponding n_{sp_j} and n_i . As indicated in [7, 47, 97, 108], n_i should be close to the number of features of the dataset ($n_f = 3600$), provided that all s_{sp_j} must also be a multiple of 4 to satisfy the DEAP constraint mentioned in Section 5.1. Therefore, n_i has been set to 3 840, resulting in the following partitioning schemes:

$$\{(n_{sp_j}, s_{sp_j})\} = \{(24, 160), (16, 240), (8, 480), (20), (4, 960), (2, 1920)\}$$

Regarding m_i , the literature seems to indicate that the recommended values for the migration interval range between 5 and 20 generations, so the values proposed in this study are:

$$\mathcal{M}_i = \{5, 10, 15, 20\} \tag{21}$$

Finally, as m_r does not usually exceed 30% of s_{sp} in most papers, the values assigned to m_l are:



Figure 2: Score values for both the average test Kappa index and the average number of selected features, obtained for the three Essex datasets and different population partitioning schemes.



Figure 3: Absolute difference in scores for the average test Kappa index between different partitioning schemes for the three Essex datasets.



Figure 4: Effect size for the average test Kappa index obtained for the three Essex datasets for different population partitioning schemes.

$$\mathcal{M}_{l} = \{0.01, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3\}$$
(22)

With these probability values, an average percentage of $(m_l \times 100)\%$ of individuals from the Pareto front of each subpopulation are selected as migrants in every generation.

Considering Equation (6), the number of different configurations for the wrapper is:

$$n_c = |\mathcal{N}_{sp}| \times |\mathcal{M}_l| \times |\mathcal{M}_l| = 5 \times 4 \times 7 = 140$$
(23)

Since the wrapper was run $n_e = 30$ times for the three datasets, a total of 12 600 executions have been performed. Figure 2 shows the score values obtained for both the average test Kappa index and the average number of selected features for the three Essex datasets and every population partitioning scheme. On the one hand, Figure 2a shows a deterioration of the average test Kappa index as individuals are distributed into more subpopulations, mainly for subjects 107 and 110. The smaller the subpopulation size, the worse the average test Kappa index,



Figure 5: Absolute difference in scores for the average number of selected features between different partitioning schemes for the three Essex datasets.



Figure 6: Effect size for the average number of selected features obtained for the three Essex datasets for different population partitioning schemes.

which deteriorates significantly with 24 subpopulations. Since each partitioning scheme received a different score, statistically significant differences between each pair of schemes are appreciated. The magnitude of such differences could also be estimated by the absolute difference in the score for each pair of schemes. Figure 3 shows such differences. It can be seen that the trend of deterioration in the average test Kappa index is modest as the subpopulation size is reduced for subject 104. Figure 4a confirms this, since the effect size is insignificant for most of the comparisons of population partitioning schemes, except for the pair ((24, 160), (2, 1920)), for which a value close to 0.2 is obtained, representing a small effect size. On the other hand, the tendency to deterioration of the average test Kappa index is stronger for subjects 107 and 110, since medium effect size values are obtained in Figures 4b and 4c. Small effect sizes appear between schemes (8, 480), (4, 960) and (2, 1 920), and a medium effect size appears between schemes (24, 160) and (2, 1920). Finally, since both the absolute difference in scores and the effect size seem to follow the same trends for the three subjects, the Pearson correlation coefficient was calculated for subjects 104, 107, and 110 between the two estimates of the magnitude of the difference, obtaining values of 0.659, 0.850, and 0.935, respectively, which reinforces the hypothesis that the novel score metric introduced in this paper can detect both statistically significant differences and the magnitude of such differences.

On the other hand, regarding the number of selected features, Figure 2b shows a steady improvement as the subpopulation size decreases, except for scheme (24, 160) whose score increases drastically. This trend can also be seen in Figures 5a to 5c, which show the absolute difference in the score for each pair of schemes, and also in Figures 6a to 6c, which confirm these difference magnitudes by their effect size estimation since their Pearson correlation coefficient is 0.795, 0.538, and 0.728 for subjects 104, 107, and 110, respectively. Small to medium effect sizes are observed when comparing schemes with a large difference between their scores. In short, the largest feature subsets are obtained with 2 and 24 subpopulations, while the smallest are obtained with 8 or 16 subpopulations.

In summary, after analyzing the two objectives optimized by the evolutionary procedure, it can be observed that as s_{sp} decreases due to an increase in n_{sp} , a deterioration of the average test Kappa index is observed, which, together with a decrease in the average number of features selected, confirms that the algorithm has an increasing tendency to converge prematurely to local optima, as discussed in Section 4.4.

Finally, the trend observed in the scores for both metrics has been confirmed by the subsequent analysis of the effect size, showing the potential of the score presented in Section 4.5, as it allows easy estimation of both statistical similarity and effect size.



Figure 7: Score values for both the average test Kappa index and the average number of selected features, obtained for the three Essex datasets and the population partitioning schemes analyzed in the binary search to determine the value for $s_{sp_{min}}$.

																1.0			
ູ (16, 240) ຮູ	i, 240) <mark>0</mark>					0					0								
9405 (16, 268) Li	0.006	0				0.087	0				0.089	0				0.6			
(16, 300)	0.081	0.087	0			0.142	0.055	0			0.163	0.074	0			0.0			
(16, 360)	0.107	0.113	0.026	0		0.16	0.073	0.018	0		0.178	0.089	0.015	0		.0.2			
(16, 480)	0.118	0.124	0.037	0.011	0	0.154	0.067	0.012	0.006	0	0.178	0.089	0.015	0	0	0.2			
	(16, 240)	(16, 268) Population	(16, 300) n partitionin	(16, 360) g scheme	(16, 480)	(16, 240)	(16, 268) Populatio	(16, 300) n partitionin	(16, 360) g scheme	(16, 480)	(16, 240)	(16, 268) Populatio	(16, 300) n partitionin	(16, 360) g scheme	(16, 480)	0.0			
	(a) 104							(b) 107			(c) 110								

Figure 8: Absolute difference in scores for the average test Kappa index on the comparison of the population partitioning schemes analyzed in the binary search to determine the value for $s_{sp_{min}}$ for the three Essex datasets.



Figure 9: Effect size for the average test Kappa index on the comparison of the population partitioning schemes analyzed in the binary search to determine the value for *s*_{spmin} for the three Essex datasets.

5.3. Determination of the optimal population partitioning scheme

According to the analysis in the previous section, the size of subpopulations seems to be a critical factor in multi-population models. Therefore, the next step is to determine the optimal population partitioning scheme $(n_{sp_{max}}, s_{sp_{min}})$ that prevents the search from stalling at local optima and also minimizes re-

source consumption, following the methodology proposed in Section 4.6. In this sense, since the degradation of results has started to become more evident for scheme (16, 240) and those with smaller subpopulation sizes, the value of $s_{sp_{min}}$ should probably be in the interval [240, 480], whose extremes are the subpopulation sizes of successive schemes (16, 240) and (8, 480), the former generating degraded results and the latter



Figure 10: Score values for each metric obtained for the three Essex datasets and the population partitioning schemes updated to consider sspmin.

not. Therefore, this interval has been explored with the binary search proposed in Algorithm 1 to find a value for $s_{sp_{min}}$ that does not degrade the results. In addition, the number of subpopulations has been set at 16, as suggested in Section 4.6, since this is the number of subpopulations of scheme (16, 240), the scheme that started to show noticeable degradation.

The remaining hyperparameters controlling the binary search were set as follows. Regarding w_{th} , which defines the minimum width of the search interval to stop the search as described in Section 4.6, since the total number of individuals is $n_i = 3840$, w_{th} has been set to 50 to obtain an adequate trade-off between resource consumption and the desired precision for $s_{sp_{min}}$. Finally, s_{th} , the similarity threshold for comparing scores (see Equation (15)) has been set to 0.05.

Figure 7 shows the scores for both the average test Kappa index and the average number of selected features and for all the schemes analyzed during the binary search procedure used to determine the value for $s_{sp_{min}}$. As observed in Figure 7a, the average test Kappa index remains relatively stable when $s_{sp} \ge 300$, deteriorating for scheme (16, 268) and especially for scheme (16, 240), which was expected from the results obtained in the previous section. This trend can also be observed in Figure 8, which shows the absolute difference in the aver-

age test Kappa index scores for comparison of population partitioning schemes. Again, these estimates of average test Kappa index differences are confirmed by Figure 9, which shows the effect size for the same scheme comparison, and by the Pearson correlation coefficient between the absolute difference in scores and the effect size, which is 0.553, 0.792, and 0.793 for subjects 104, 107, and 110, respectively. It can be observed that small or small-medium effect size values are obtained for schemes (16, 268) and (16, 240), especially for datasets 107 and 110, verifying the degradation of the average test Kappa index. By contrast, considering the average number of selected features, Figure 7b shows that for schemes (16, 480), (16, 360), and (16, 300), the results show some stability. In contrast, for schemes (16, 268) and (16, 240), depending on the dataset, an erratic trend starts to appear. This behavior could be a symptom of premature convergence to local optima. Absolute differences in scores and effect size are not reported in this case because the scores are relatively similar. In particular, the effect size is less than 0.2 in all cases. In this context, it seems that the value for $s_{SD_{min}}$ should be between 300 and 268 individuals. However, since it is not worth searching further because this interval is too narrow, the minimum threshold for the subpopulation size has been estimated to be $s_{sp_{min}} = 300$, which is about 10% of



Figure 11: Speedup in execution time and savings in energy consumption achieved by the (24, 300) and the (16, 300) schemes concerning the (2, 1920) scheme using the three Essex datasets.

Table 3: Effect size for the average test Kappa index and the average number of selected features when comparing schemes (24, 300) and (16, 300) for the three Essex datasets.

Metric	Dataset	Effect size value
Average test	104	0.013
Kappa indax	107	0.077
Rappa muex	110	0.063
Average number	104	0.070
of selected	107	0.106
features	110	0.131

the total number of individuals.

Now it is time to verify that the premature convergence problem has been successfully solved considering values of $s_{sp} \ge s_{sp_{min}}$. This can be confirmed in Figure 10, which shows the score obtained for each metric using the three Essex datasets for different population partitioning schemes and considering $s_{sp_{min}}$. As shown in Figure 10a, the average test Kappa index has stabilized for all schemes, so premature convergence seems to have been mitigated. Furthermore, Figure 10b indicates that the best subsets of selected features are obtained with schemes (24, 300) and (16, 300).

According to Section 4.6, the maximum threshold for the number of subpopulations should be $n_{sp_{max}} = 16$, the number of subpopulations fixed to perform the binary search of $s_{sp_{min}}$. This can also be confirmed by Figures 10c and 10d, which show that the efficiency of the wrapper improves as individuals are distributed into an increasing number of subpopulations until $n_{sp} = 16$ is reached, and then drops for scheme (24, 300), as expected. In this context, Figure 11 shows the speedup in execution time and the energy savings achieved by the (24, 300) and the (16, 300) schemes with respect to the slower scheme, i.e., (2, 1920). It can be seen that scheme (16, 300) is more efficient than (24, 300), especially in terms of energy consumption. On the other hand, the scores for both the average test Kappa index and the average number of selected features are slightly lower for scheme (24, 300) than for scheme (16, 300). However, such differences are negligible and both schemes should be considered to produce statistically similar results. This is confirmed by Table 3, which shows effect sizes below 0.2 for both metrics when comparing the schemes. Therefore, it can be confirmed that values of $n_{sp} > n_{sp_{max}}$ require additional computational time and energy that are not worth a nonsignificant improvement in the quality of solutions.

5.4. Influence of the migration interval and the number of migrants

Once the problem of premature convergence seems to have been mitigated, this section analyzes the wrapper's behavior according to hyperparameters m_i and m_l . The study has been conducted using the (16, 300) population partitioning scheme because it makes the evolutionary procedure to obtain appropriate solutions while minimizing both its execution time and energy consumption.

Every possible combination of m_i and m_l for the three Essex datasets yielded the same score value in terms of both the average test Kappa index and the average number of features selected, indicating that there are no statistically significant differences regardless of the combination of m_i and m_l used, so no further analysis is required. However, hyperparameters m_i and m_l do affect the efficiency of the procedure, as can be seen in Figure 12, which shows a heat map of the energy consumption scores obtained with different combinations of m_i and m_l using the three Essex datasets. Since the trends in energy consumption and execution time are co-dependent, as a decrease in execution time necessarily leads to an improvement in energy consumption, as noted in the previous section, only the energy consumption results are reported. The same trend for hyperparameters m_i and m_l is presented for the three datasets. An improvement in efficiency is observed for each value of m_i as the value of m_i is increased. In addition, low values of m_i seem to be more suitable to obtain the best efficiency. In this sense, the scores indicate that tuning m_i to low values and m_l to high values is the best strategy to accelerate convergence. Figure 13 shows that these differences are due to the number of generations required for convergence. As can be seen, the evolutionary

													_	-10
0.01	0.357	0.679	0.911	0.911	0.286	0.732	0.732	0.929	0.446	0.804	0.804	0.804		1.0
0.05	0.357	0.679	0.679	0.911	0.286	0.732	0.732	0.929	0.446	0.804	0.804	0.804		0.8
0.1	0.357	0.357	0.679	0.911	0.286	0.732	0.732	0.929	0.214	0.446	0.804	0.804		-0.6
NI 0.15	0.089	0.357	0.679	0.679	0.089	0.482	0.482	0.732	0.071	0.446	0.446	0.804		0.0
0.2	0.089	0.357	0.679	0.679	0.089	0.286	0.482	0.732	0.071	0.214	0.446	0.804		0.4
∑ 0.25	0.089	0.089	0.357	0.679	0.089	0.089	0.286	0.482	0.071	0.214	0.446	0.446		-0.2
0.3	0.089	0.089	0.357	0.357	0.089	0.089	0.286	0.482	0.071	0.071	0.446	0.446		
	5	10 Migratio	15 n interval	20	5	10 Migratior	15 n interval	20	5	10 Migratio	15 n interval	20	_	
		(a) 10	4			(b)	107				(c) 110			

Figure 12: Score values obtained for the energy consumption with different m_i and m_l combinations using the three Essex datasets and the population partitioning scheme (16, 300).

wrapper converges faster as m_i decreases and m_l increases for the three datasets.

5.5. Validation of the elitist probabilistic migration policy

After optimizing the migration process, it is worth comparing the performance of the novel migration policy presented in Section 3 with that of the two most commonly used policies in the literature: the best-worst and random-random policies (see Section 2.1.1). For this purpose, it is important to remember the difficulty of selecting the best individuals from a subpopulation when solving a multi-objective problem (Section 2.1.3). Indeed, this issue motivates the development of the elitist probabilistic migration proposed in this paper. However, this problem is not common to all multi-objective evolutionary methods. In the case of NSGA-II, which is the basis of the wrapper applied previously, the crowding distance [109] allows sorting of the individuals within each front of non-dominated solutions, leading to a completely ordered population. In this case, applying the best-worst policy, as originally defined by Cantú-Paz [55], is straightforward. For other multi-objective evolutionary procedures, an adaptation of the method, such as the one proposed in this paper, is required.

In light of the aforementioned considerations, the migration policy comparison is based on the same experimental conditions as in the previous section. Therefore, the same wrapper procedure was used, run on the same datasets, and configured with the previously estimated optimal hyperparameters, i.e., $n_{sp} = 16$, $s_{sp} = 300$, and $m_i = 5$. For elitist probabilistic migration, m_l was set to 0.3 as described above. This value was also used for m_r when applying the *best-worst* and *random-random* policies. The wrapper was run $n_e = 30$ times for 500 generations for each dataset and each migration policy.

Figure 14 shows the average test Kappa index, the average number of selected features, and the average number of migrants chosen by the wrapper when using the three different migration policies for the three Essex datasets. It can be observed that the new proposed migration policy and the *best-worst* policy obtain similar results for the two objectives optimized by the wrapper, which was expected since both consider the best individuals in the population. In contrast, it seems that the wrapper

experiences a degradation in its effectiveness when the *randomrandom* policy is used, which could be due to the lack of elitism of the policy. Regardless of the effectiveness of the wrapper, Figure 14c shows how the number of migrants was considerably reduced by the elitist probabilistic policy compared to the other two strategies, as expected, from a fixed number of 60 migrants selected by both the *best-worst* and *random-random* policies to about 10 migrants for the elitist probabilistic policy.

Figure 15 shows the scores obtained for the average test Kappa index, the average number of selected features, and the average number of migrants chosen by the wrapper, confirming that the elitist probabilistic migration policy and the *best-worst* policy obtain comparable results for both objectives. In contrast, the *random-random* policy obtains results with a comparable average number of features, although it cannot reach the average test Kappa index obtained by the elitist policies. On the other hand, Figure 15c shows that the number of migrants is significantly reduced, receiving a score value of 0. Therefore, it can be stated that the novel proposed policy achieves appropriate solutions while minimizing the overhead of the migration process, mainly due to the significant reduction in the number of migrants.

5.6. Wrapper behavior assessment

Once the migration hyperparameters have been properly adjusted, it becomes relevant to assess the effectiveness of the wrapper used in this study, referred to as MOEW (Multi-Objective Evolutionary Wrapper), when analyzing the Essex EEG datasets, and to compare it against other state-of-the-art approaches. These alternative methods are described as follows. On the one hand, LeOCCEA is a Lexicographic Optimization Cooperative Co-Evolutionary Algorithm proposed in [97]. On the other hand, MEW, a Multi-objective Evolutionary Wrapper method described in [47], has also been applied to the same datasets using different classifiers. Specifically, four classification approaches have been used: k-NN, a Naive Bayes Classifier (NBC), and the application of Linear Discriminant Analysis (LDA) before trying with either k-NN or NBC, thus providing four wrapper approaches: MEW-KNN, MEW-NBC, MEW-LDA-KNN, and MEW-LDA-NBC. Finally, PGA-CNN is a hyperparameter optimization procedure based on Genetic



best-worst random-random Elitist probabilistic 0.7 0.6 0.5 0.4 104 107 110 Dataset (a) Average test Kappa index





Figure 13: Average number of generations needed to achieve convergence for different m_i and m_l combinations using the three Essex datasets.

Algorithms for Convolutional Neural Networks (CNNs) proposed in [110]. The rest of the hyperparameters have been adjusted as previously described.

Figure 16 depicts the average test Kappa index and the average number of selected features obtained by the different wrapper methods for the Essex EEG datasets. As can be observed, MOEW may appear to obtain competitive subsets of features. This is further supported by Figure 17, which shows the score

Figure 14: Average test Kappa index, average number of selected features, and average number of migrants chosen by the wrapper for the different migration policies and the three Essex datasets.

values obtained for the average test Kappa index and average number of selected features for the different wrapper alternatives, confirming the effectiveness of MOEW.

6. Conclusions

An experimental methodology has been proposed to determine the influence of the most relevant migration hyperpa-



Figure 15: Score values for the average test Kappa index, average number of selected features, and average number of migrants for the different migration policies and the three Essex datasets.

rameters on the performance of a multi-objective and multipopulation evolutionary method applied to complex problems requiring large populations, to confirm whether the guidelines proposed 20 years ago by authors such as Cantú-Paz or Tomassini are still valid today, when both the dimensionality of the problems and the size of the population required to solve them have increased considerably. In particular, hyperparam-



(b) Average number of selected features

Figure 16: Average test Kappa index and average number of selected features obtained by the different wrapper alternatives for the Essex EEG datasets.

eters m_i , m_l , and n_{sp} are analyzed. Tanese's guideline is used to determine all population partitioning schemes (n_{sp}, s_{sp}) according to the total number of individuals n_i and each possible value for n_{sp} . In addition to the objectives optimized by the evolutionary procedure, its execution time and energy consumption are also considered metrics to be optimized. Furthermore, the large number of individuals, and the fact that multi-objective optimization problems are considered, have led to the development of a new elitist probability-based migration policy, in which migrants are selected from the Pareto front, aiming to reduce the number of migrants while favoring elitism. Since the size of the Pareto front is variable in each generation, the hyperparameter m_l is proposed to set the probability that each Pareto optimal individual will become a migrant. If no migrant is finally selected, an individual is randomly selected from the Pareto front.

First, the influence of the population partitioning scheme is analyzed by a grid search over all possible migration hyperparameter combinations to find a narrow interval for the subpopulation size at which premature convergence becomes apparent. Then, this interval is explored with a binary search procedure to find an adequate estimate for $s_{sp_{min}}$, the minimum threshold



Figure 17: Score values for the average test Kappa index and average number of selected features for the different wrapper alternatives for the Essex EEG datasets.

for the subpopulation size. Next, $n_{sp_{max}}$, the maximum threshold for the number of subpopulations is determined, resulting in $(n_{sp_{max}}, s_{sp_{min}})$, the optimal population partitioning scheme for the problem. Finally, this scheme is used to determine the influence of m_i and m_l on the behavior of the procedure.

This experimental study is supported by a novel rankingbased score that combines both parametric and non-parametric statistical tests to split the results sample distributions of all the experiments performed into clusters of migration hyperparameter combinations that provide similar results. These clusters can be ranked and assigned a scalar score that allows multiple comparison of the results obtained by the evolutionary procedure with different configurations. The lower the score, the better the parameterization.

This methodology was validated by an NSGA-II-based wrapper applied to three high-dimensional FS datasets related to a BCI motor imagery problem. The results showed a high probability of premature convergence as the size of the subpopulations decreases, consistent with the conclusions of Cantú-Paz in [54]. The optimal population partitioning scheme ($n_{sp_{max}}$, $s_{sp_{min}}$) was then determined for the three datasets, showing that the use of 16 subpopulations yields results comparable to those of the use of 24 but is more computationally efficient. A priori, this behavior might be unexpected, as it has been assumed over the years that increasing the number of subpopulations is the best guideline to follow [69, 111, 112]. However, it seems that issues such as the appropriate values for hyperparameters n_{sp} and s_{sp} , with $s_{sp} \ge s_{sp_{min}}$ and $n_{sp} = n_{sp_{max}}$, should be considered to achieve the best performance of a multi-population model. Next, the influence of m_i and m_l on the evolutionary procedure was analyzed, and it was found that they only affect the efficiency of the procedure, causing it to converge faster as m_i decreases and m_i increases. This may contradict the guidelines given by Tomassini [24], who recommended the use of average values for both hyperparameters. In fact, Tomassini reported a degradation in the quality of solutions when these hyperparameters were tuned in this way. Perhaps this degradation could have been avoided by properly adjusting n_{sp} and s_{sp} . Finally, regarding the new elitist probabilistic migration policy, it has shown comparable results to the classical best-worst policy while requiring significantly fewer migrants, thus reducing the migration overhead.

It is worth mentioning the extent of the experimental study carried out to obtain the results presented in this work. The relationship between the population partitioning scheme and the likelihood of premature convergence was confirmed by applying the wrapper to three high-dimensional datasets, using five population partitioning schemes, four values for the migration interval, and seven values for m_l . Each different combination of these hyperparameters was used to run the wrapper 30 times, resulting in 12 600 runs of the wrapper. Then, the binary search used to estimate the optimal population partitioning scheme tried another four additional population partitioning schemes, requiring 10080 executions, and finally, the validation of the optimal population partitioning scheme needed another 2 520 runs of the wrapper. Thus, only the experimental part of this work took more than a year to complete, despite using six highperformance computing nodes.

Regarding the score metric proposed to allow multiple comparison of the results sampling distributions of the huge amount of data generated, it has proven useful to detect statistically significant differences among distributions and also allows the clustering of hyperparameter combinations that yield comparable results. Its most relevant aspect, however, is that it can also be used to estimate the effect size of differences in results, taking into account the high correlation between the score difference and the Cohen's d index obtained for all the experiments performed.

Acknowledgment

We would like to thank the BCI laboratory of the University of Essex, especially Dr. John Q. Gan, for allowing us to use their datasets.

Availability of data and materials

All data used for this work are available in the respective references given in this article. Moreover,

source code is available in the vegetto public repository: https://github.com/efficomp/vegetto

References

- R. E. Bellman, Adaptive Control Processes: A Guided Tour, Princeton University Press, 1961.
- [2] C. Blum, A. Roli, Metaheuristics in combinatorial optimization: Overview and conceptual comparison, ACM Computing Surveys 35 (2003) 268–308. doi:10.1145/937503.937505.
- [3] Y. Tian, L. Si, X. Zhang, R. Cheng, C. He, K. C. Tan, Y. Jin, Evolutionary large-scale multi-objective optimization: A survey, ACM Computing Surveys 54 (8) (2021). doi:10.1145/3470971.
- [4] V. Pareto, Cours d'économie politique, Vol. 1 and 2, F. Rouge, Lousane, 1896.
- [5] S. Liu, Q. Lin, K. C. Wong, Q. Li, K. C. Tan, Evolutionary largescale multiobjective optimization: Benchmarks and algorithms, IEEE Transactions on Evolutionary Computation 27 (2023) 401–415. doi: 10.1109/TEVC.2021.3099487.
- [6] S. Liu, Q. Lin, J. Li, K. C. Tan, A survey on learnable evolutionary algorithms for scalable multiobjective optimization, IEEE Transactions on Evolutionary Computation 27 (2023) 1941–1961. doi:10.1109/ TEVC.2023.3250350.
- [7] J. C. Gómez-López, D. Castillo-Secilla, J. González, Improving the performance of ea-based multi-population models for feature selection problems by reducing the individual size in the initial population, in: Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO'2024, Association for Computing Machinery, Melbourne, Australia, 2024, pp. 335–338. doi:10.1145/ 3638530.3654424.
- [8] J. C. Gómez-López, J. J. Escobar, A. F. Díaz, M. Damas, F. Gil-Montoya, J. González, Boosting the convergence of a GA-based wrapper for feature selection problems on high-dimensional data, in: Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO'2022, Association for Computing Machinery, Boston, Massachusetts. EEUU, 2022, pp. 431–434. doi:10.1145/3520304. 3528800.
- [9] Z. Zhan, J. Li, J. Cao, J. Zhang, H. Chung, Y. Shi, Multiple populations for multiple objectives: A coevolutionary technique for solving multiobjective optimization problems, IEEE Transactions on Cybernetics 43 (2013) 445–463. doi:10.1109/TSMCB.2012.2209115.
- [10] J. J. Grefenstette, Parallel adaptive algorithms for function optimization, Vanderbilt University, Nashville, TN, Tech. Rep. CS-81-19 (1981).
- [11] R. Tanese, Parallel genetic algorithm for a hypercube, in: Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms, Hillsdale, NJ: L. Erlhaum Associates, 1987., The Massachusetts Institute of Technology, Cambridge, MA, Boston, USA, 1987, pp. 177–183.
- [12] S. Lin, W. F. Punch, E. D. Goodman, Coarse-grain parallel genetic algorithms: Categorization and new approach, in: Proceedings of 1994 6th IEEE Symposium on Parallel and Distributed Processing, IPDPS'94, IEEE, Dallas, TX, USA, 1994, pp. 28–37. doi:10.1109/SPDP.1994. 346184.
- [13] D. Whitley, S. Rana, R. B. Heckendorn, The island model genetic algorithm: On separability, population size and convergence, Journal of Computing and Information Technology 7 (1999) 33–47.
- [14] D. Andre, J. R. Koza, Parallel Genetic Programming: A Scalable Implementation Using the Transputer Network Architecture, Vol. 2, MIT Press, Cambridge, MA, USA, 1996, Ch. 16, pp. 317–337.
- [15] P. B. Grosso, Computer Simulations of Genetic Adaptation: Parallel Subcomponent Interaction in a Multilocus Model, University of Michigan, 1985.
- [16] J. C. Gómez-López, D. Castillo-Secilla, D. Kimovski, J. González, Boosting NSGA-II-based wrappers speedup for high-dimensional data: Application to EEG classification, in: Advances in Computational Intelligence, IWANN'23, Springer, Ponta Delgada, Azores, Portugal, 2023, pp. 80–91. doi:10.1007/978-3-031-43085-5_7.
- [17] S. Liu, Q. Lin, L. Feng, K. C. Wong, K. C. Tan, Evolutionary multitasking for large-scale multiobjective optimization, IEEE Transactions on Evolutionary Computation 27 (2023) 863–877. doi:10.1109/TEVC. 2022.3166482.

- [18] W. Y. Lin, T. P. Hong, S. M. Liu, On adapting migration parameters for multi-population genetic algorithms, in: 2004 IEEE International Conference on Systems, Man and Cybernetics, Vol. 6, IEEE, The Hague, Netherlands, 2004, pp. 5731–5735.
- [19] K. Y. Szeto, Y. Guo, Multi-population genetic algorithm for locating multi-optima in noisy complex landscape, Communications in Statistics - Theory and Methods 40 (2011) 3029–3048. doi:10.1080/ 03610926.2011.562789.
- [20] W. L. Lin, T. P. Hong, S. M. Liu, J. H. Lin, Revisiting the design of adaptive migration schemes for multipopulation genetic algorithms, in: 2012 Conference on Technologies and Applications of Artificial Intelligence, IEEE, Tainan, Taiwan, 2012, pp. 338–343. doi:10.1109/TAAI.2012. 41.
- [21] E. Cantú-Paz, A survey of parallel genetic algorithms, Calculateurs Paralleles, Eeseaux et Systems Repartis 10 (1998) 141–171.
- [22] E. Cantú-Paz, Topologies, migration rates, and multi-population parallel genetic algorithms, in: Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation - Volume 1, GECCO'99, Orlando, Florida, 1999, pp. 91—98.
- [23] E. Cantú-Paz, D. E. Goldberg, Efficient parallel genetic algorithms: Theory and practice, Computer Methods in Applied Mechanics and Engineering 186 (2000) 221–238. doi:10.1016/S0045-7825(99) 00385-0.
- [24] M. Tomassini, Spatially Structured Evolutionary Algorithms: Artificial Evolution in Space and Time, Springer, 2006.
- [25] G. Ewald, W. Kurek, M. A. Brdys, Grid implementation of a parallel multiobjective genetic algorithm for optimized allocation of chlorination stations in drinking water distribution systems: Chojnice case study, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 38 (2008) 497–509. doi:10.1109/TSMCC.2008. 923864.
- [26] S. Nesmachnow, H. Cancela, E. Alba, A parallel micro evolutionary algorithm for heterogeneous computing and grid scheduling, Applied Soft Computing 12 (2012) 626–639. doi:10.1016/j.asoc.2011. 09.022.
- [27] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, H. Liu, Feature selection: A data perspective, ACM Comput. Surv. 50 (2017) 1–45. doi:10.1145/3136625.
- [28] B. Venkatesh, J. Anuradha, A review of feature selection and its methods, Cybernetics and Information Technologies 19 (1) (2019) 3–26. doi:doi:10.2478/cait-2019-0001.
- [29] J. C. Gómez-López, D. Castillo-Secilla, J. González, L. J. Herrera, I. Rojas, Towards the identification of multiclass lung cancer-related genes: An evolutionary and intelligent procedure, in: Advances in Computational Intelligence, IWANN'23, Springer, Ponta Delgada, Azores, Portugal, 2023, pp. 553–562. doi:10.1007/978-3-031-43085-5_44.
- [30] C. Zhang, B. Jiang, Z. Wang, J. Yang, Y. Lu, X. Wu, W. Sheng, Efficient multi-view semi-supervised feature selection, Information Sciences 649 (2023) 119675. doi:10.1016/j.ins.2023.119675.
- [31] J. Lienig, A parallel genetic algorithm for performance-driven vlsi routing, IEEE Transactions on Evolutionary Computation 1 (1997) 29–39. doi:10.1109/4235.585890.
- [32] H. Pierreval, J. L. Paris, Distributed evolutionary algorithms for simulation optimization, IEEE Transactions on Systems, Man, and Cybernetics
 Part A: Systems and Humans 30 (2000) 15–24. doi:10.1109/3468.
 823477.
- [33] X. Zhuang, Y. Tian, H. Wang, S. A. Ali, Neural network adaptive observer design for nonlinear systems with partially and completely unknown dynamics subject to variable sampled and delay output measurement, Neurocomputing 561 (2023) 126865. doi:10.1016/j.neucom. 2023.126865.
- [34] C. Anglano, M. Botta, Now g-net: Learning classification programs on networks of workstations, IEEE Transactions on Evolutionary Computation 6 (2002) 463–480. doi:10.1109/TEVC.2002.800882.
- [35] Y. Nojima, S. Mihara, H. Ishibuchi, Ensemble classifier design by parallel distributed implementation of genetic fuzzy rule selection for large data sets, in: IEEE Congress on Evolutionary Computation, 2010, pp. 1–8. doi:10.1109/CEC.2010.5586393.
- [36] N. Zeng, H. Qiu, Z. Wang, W. Liu, H. Zhang, Y. Li, A new switching-delayed-PSO-based optimized svm algorithm for diagnosis of alzheimer's disease, Neurocomputing 320 (2018) 195–202. doi:

10.1016/j.neucom.2018.09.001.

- [37] H. Geng, F. Song, J. Shen, J. Li, A classification and regression assisted optimization algorithm for high-dimensional expensive many-objective problems, Neurocomputing 586 (2024) 127629. doi:10.1016/j. neucom.2024.127629.
- [38] L. Kattan, B. Abdulhai, Sensitivity analysis of an evolutionary-based time-dependent origin/destination estimation framework, IEEE Transactions on Intelligent Transportation Systems 13 (2012) 1442–1453. doi:10.1109/TITS.2012.2189437.
- [39] B. Yu, Z. Yang, X. Sun, B. Yao, Q. Zeng, E. Jeppesen, Parallel genetic algorithm in bus route headway optimization, Applied Soft Computing 11 (2011) 5081–5091. doi:10.1016/j.asoc.2011.05.051.
- [40] J. C. Creput, A. Koukam, T. Lissajoux, A. Caminada, Automatic mesh generation for mobile network dimensioning using evolutionary approach, IEEE Transactions on Evolutionary Computation 9 (2005) 18– 30. doi:10.1109/TEVC.2004.837923.
- [41] Y. Chen, B. Wang, W. S. Lin, Y. Wu, K. J. R. Liu, Cooperative peer-topeer streaming: An evolutionary game-theoretic approach, IEEE Transactions on Circuits and Systems for Video Technology 20 (2010) 1346– 1357. doi:10.1109/TCSVT.2010.2077490.
- [42] P. Chen, J. Pei, W. Lu, M. Li, A deep reinforcement learning based method for real-time path planning and dynamic obstacle avoidance, Neurocomputing 497 (2022) 64–75. doi:10.1016/j.neucom.2022. 05.006.
- [43] A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay, C. A. Coello, A survey of multiobjective evolutionary algorithms for data mining: Part I, IEEE Transactions on Evolutionary Computation 18 (2014) 4–19. doi:10.1109/TEVC.2013.2290086.
- [44] A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay, C. A. Coello, Survey of multiobjective evolutionary algorithms for data mining: Part II, IEEE Transactions on Evolutionary Computation 18 (2014) 20–35. doi:10. 1109/TEVC.2013.2290082.
- [45] B. Xue, M. Zhang, W. N. Browne, X. Yao, A survey on evolutionary computation approaches to feature selection, IEEE Transactions on Evolutionary Computation 20 (2016) 606–626. doi:10.1109/TEVC. 2015.2504420.
- [46] M. Sheng, Z. Wang, W. Liu, X. Wang, S. Chen, X. Liu, A particle swarm optimizer with multi-level population sampling and dynamic p-learning mechanisms for large-scale optimization, Knowledge-Based Systems 242 (2022) 108382. doi:10.1016/j.knosys.2022.108382.
- [47] J. González, J. Ortega, M. Damas, P. Martín-Smith, J. Q. Gan, A new multi-objective wrapper method for feature selection – accuracy and stability analysis for BCI, Neurocomputing 333 (2019) 407–418. doi:10.1016/j.neucom.2019.01.017.
- [48] A. M. Turky, S. Abdullah, A multi-population harmony search algorithm with external archive for dynamic optimization problems, Information Sciences 272 (2014) 84–95. doi:10.1016/j.ins.2014.02.084.
- [49] H. Kwasnicka, M. Przewozniczek, Multi population pattern searching algorithm: A new evolutionary method based on the idea of messy genetic algorithm, IEEE Transactions on Evolutionary Computation 15 (2011) 715–734. doi:10.1109/TEVC.2010.2102038.
- [50] J. E. Fieldsend, Running up those hills: Multi-modal search with the niching migratory multi-swarm optimiser, in: 2014 IEEE Congress on Evolutionary Computation (CEC), 2014, pp. 2593–2600. doi: 10.1109/CEC.2014.6900309.
- [51] E. Oró, V. Depoorter, A. Garcia, J. Salom, Energy efficiency and renewable energy integration in data centres. strategies and modelling review, Renewable and Sustainable Energy Reviews 42 (2015) 429–445. doi:10.1016/j.rser.2014.10.035.
- [52] T. Mastelic, A. Oleksiak, H. Claussen, I. Brandic, J. M. Pierson, A. V. Vasilakos, Cloud computing: Survey on energy efficiency, ACM Comput. Surv. 47 (2014). doi:10.1145/2656204.
- [53] E. Cantú-Paz, D. E. Goldberg, Modeling idealized bounding cases of parallel genetic algorithms, in: J. Koza, K. Deb, M. Dorigo, D. Fogel, M. Garzon, H. Iba, R. Riolo (Eds.), Genetic Programming 1997: Proceedings of the Second Annual Conference, Citeseer, Morgan Kaufmann, 1997, pp. 353—361.
- [54] E. Cantú-Paz, Efficient and Accurate Parallel Genetic Algorithms, Springer Science & Business Media, 2000.
- [55] E. Cantú-Paz, Migration policies, selection pressure, and parallel evolutionary algorithms, Journal of Heuristics 7 (2001) 311–334. doi:

10.1023/A:1011375326814.

- [56] R. Tanese, Distributed genetic algorithms, in: Proceedings of the 3rd International Conference on Genetic Algorithms, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1989, pp. 434—439. doi: 10.5555/645512.657245.
- [57] T. Hiroyasu, M. Miki, M. Negami, Distributed genetic algorithms with randomized migration rate, in: IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No.99CH37028), Vol. 1 of SMC'99, IEEE, Tokyo, Japan, 1999, pp. 689–694. doi:10.1109/ICSMC.1999.814175.
- [58] Z. Skolicki, K. A. D. Jong, The influence of migration sizes and intervals on island models, in: Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation, GECCO'05, Association for Computing Machinery, Washington DC, USA, 2005, pp. 1295–1302. doi:10.1145/1068009.1068219.
- [59] M. A. Al-Betar, M. A. Awadallah, Island bat algorithm for optimization, Expert Systems with Applications 107 (2018) 126–145. doi:10.1016/ j.eswa.2018.04.024.
- [60] M. A. Al-Betar, M. A. Awadallah, I. A. Doush, A. I. Hammouri, M. Mafarja, Z. A. A. Alyasseri, Island flower pollination algorithm for global optimization, The Journal of Supercomputing 75 (2019) 5280–5323. doi:10.1007/s11227-019-02776-y.
- [61] M. A. Awadallah, M. A. Al-Betar, A. L. Bolaji, I. A. Doush, A. I. Hammouri, M. Mafarja, Island artificial bee colony for global optimization, Soft Computing 24 (2020) 13461–13487. doi:10.1007/ s00500-020-04760-8.
- [62] I. Arnaldo, I. Contreras, D. Millán-Ruiz, J. Hidalgo, N. Krasnogor, Matching island topologies to problem structure in parallel evolutionary algorithms, Soft Computing 17 (2013) 1209–1225. doi:10.1007/ s00500-013-0995-9.
- [63] J. Kushida, A. Hara, T. Takahama, A. Kido, Island-based differential evolution with varying subpopulation size, in: 2013 IEEE 6th International Workshop on Computational Intelligence and Applications (IW-CIA), 2013, pp. 119–124. doi:10.1109/IWCIA.2013.6624798.
- [64] F. Neumann, P. S. Oliveto, G. Rudolph, D. Sudholt, On the effectiveness of crossover for migration in parallel evolutionary algorithms, in: Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation, GECCO'11, Association for Computing Machinery, 2011, pp. 1587—1594. doi:10.1145/2001576.2001790.
- [65] X. Zhou, R. Zhao, F. You, EFSM-based test data generation with multipopulation genetic algorithm, in: 2014 IEEE 5th International Conference on Software Engineering and Service Science, ICSESS'14, IEEE, Beijing, China, 2014, pp. 925–928. doi:10.1109/ICSESS.2014. 6933716.
- [66] M. Ruciński, D. Izzo, F. Biscani, On the impact of the migration topology on the island model, Parallel Computing 36 (2010) 555–571. doi: 10.1016/j.parco.2010.04.002.
- [67] K. Ono, Y. Hanada, M. Kumano, M. Kimura, Island model genetic programming based on frequent trees, in: 2013 IEEE Congress on Evolutionary Computation, 2013, pp. 2988–2995. doi:10.1109/CEC. 2013.6557933.
- [68] L. Falahiazar, M. Teshnehlab, A. Falahiazar, Parallel genetic algorithm based on a new migration strategy, in: 2012 International Conference on Recent Advances in Computing and Software Systems, IEEE, Chennai, India, 2012, pp. 37–41. doi:10.1109/RACSS.2012.6212694.
- [69] Y. J. Gong, W. N. Chen, Z. H. Zhan, J. Zhang, Y. Li, Q. Zhang, J. J. Li, Distributed evolutionary algorithms and their models: A survey of the state-of-the-art, Applied Soft Computing 34 (2015) 286–300. doi: 10.1016/j.asoc.2015.04.061.
- [70] L. Araujo, J. J. Merelo, Diversity through multiculturality: Assessing migrant choice policies in an island model, IEEE Transactions on Evolutionary Computation 15 (2011) 456–469. doi:10.1109/TEVC.2010. 2064322.
- [71] W. Guo, L. Wang, Q. Wu, Numerical comparisons of migration models for multi-objective biogeography-based optimization, Information Sciences 328 (2016) 302–320. doi:https://doi.org/10.1016/j. ins.2015.07.059.
- [72] A. M. Mora, P. García-Sánchez, J. J. Merelo, P. A. Castillo, Paretobased multi-colony multi-objective ant colony optimization algorithms: An island model proposal, Soft Computing 17 (2013) 1175–1207. doi: 10.1007/s00500-013-0993-y.

- [73] N. Melab, M. Mezmaz, E.-G. Talbi, Parallel hybrid multi-objective island model in peer-to-peer environment, in: 19th IEEE International Parallel and Distributed Processing Symposium, IPDPS'05, IEEE, Denver, CO, USA, 2005, p. 9. doi:10.1109/IPDPS.2005.327.
- [74] A. Ouyang, K. Li, X. Fei, X. Zhou, M. Duan, A novel hybrid multiobjective population migration algorithm, International Journal of Pattern Recognition and Artificial Intelligence 29 (2015). doi:10.1142/ S0218001415590016.
- [75] A. Sallam, K. Li, A multi-objective virtual machine migration policy in cloud systems, The Computer Journal 57 (2014) 195–204. doi:10. 1093/comjnl/bxt018.
- [76] Y. An, X. Chen, Y. Li, Y. Han, J. Zhang, H. Shi, An improved nondominated sorting biogeography-based optimization algorithm for the (hybrid) multi-objective flexible job-shop scheduling problem, Applied Soft Computing 99 (2021) 106869. doi:10.1016/j.asoc.2020. 106869.
- [77] D. Izzo, M. Ruciński, C. Ampatzis, Parallel global optimisation metaheuristics using an asynchronous island-model, in: 2009 IEEE Congress on Evolutionary Computation, Trondheim, Norway, 2009, pp. 2301– 2308. doi:10.1109/CEC.2009.4983227.
- [78] M. D. Nasir, S. Das, D. Maity, S. Sengupta, U. Halder, P. N. Suganthan, A dynamic neighborhood learning based particle swarm optimizer for global numerical optimization, Information Sciences 209 (2012) 16–36. doi:10.1016/j.ins.2012.04.028.
- [79] N. S. Jaddi, S. Abdullah, A. R. Hamdan, Multi-population cooperative bat algorithm-based optimization of artificial neural network model, Information Sciences 294 (2015) 628–644. doi:10.1016/j.ins.2014. 08.050.
- [80] A. Czarn, C. MacNish, K. Vijayan, B. Turlach, R. Gupta, Statistical exploratory analysis of genetic algorithms, IEEE Transactions on Evolutionary Computation 8 (2004) 405–421. doi:10.1109/TEVC.2004. 831262.
- [81] B. Ozcelik, T. Erzurumlu, Comparison of the warpage optimization in the plastic injection molding using anova, neural network model and genetic algorithm, Journal of Materials Processing Technology 171 (2006) 437-445. doi:10.1016/j.jmatprotec.2005.04.120.
- [82] I. Rojas, J. . González, H. Pomares, J. J. Merelo, P. A. Castillo, G. Romero, Statistical analysis of the main parameters involved in the design of a genetic algorithm, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 32 (1) (2002) 31–37. doi:10.1109/TSMCC.2002.1009128.
- [83] D. Hatchavanich, A comparison of type i error and power of bartlett's test, levene's test and o'brien's test for homogeneity of variance tests, Southeast Asian Journal of Sciences 3 (2014) 181–194.
- [84] M. Jamshidian, S. Jalal, Tests of homoscedasticity, normality, and missing completely at random for incomplete multivariate data, Psychometrika 75 (2010) 649–674. doi:10.1007/s11336-010-9175-3.
- [85] S. García, D. Molina, M. Lozano, F. Herrera, A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: A case study on the cec'2005 special session on real parameter optimization, Journal of Heuristics 15 (2009) 617–644. doi: 10.1007/s10732-008-9080-4.
- [86] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm and Evolutionary Computation 1 (2011) 3–18. doi:10.1016/j.swevo.2011.02.002.
- [87] G. M. Sullivan, R. Feinn, Using effect size—or why the p value is not enough, Journal of Graduate Medical Education 4 (2012) 279–282. doi:10.4300/JGME-D-12-00156.1.
- [88] A. K. Akobeng, Understanding type i and type ii errors, statistical power and sample size, Acta Paediatrica 105 (2016) 605–609. doi:10.1111/ apa.13384.
- [89] A. Bakker, J. Cai, L. English, G. Kaiser, V. Mesa, W. V. Dooren, Beyond small, medium, or large: Points of consideration when interpreting effect sizes, Educational Studies in Mathematics 102 (2019) 1–8. doi:10. 1007/s10649-019-09908-4.
- [90] C. J. Huberty, A history of effect size indices, Educational and Psychological Measurement 62 (2002) 227–240. doi:10.1177/ 0013164402062002002.
- [91] J. Cohen, Statistical Power Analysis for the Behavioral Sciences, Academic Press, 2013.

- [92] R. E. Kirk, Practical significance: A concept whose time has come, Educational and Psychological Measurement 56 (1996) 746—759. doi: 10.1177/0013164496056005002.
- [93] C. Ialongo, Understanding the effect size and its measures, Biochemia Medica 26 (2016) 150–163. doi:10.11613/BM.2016.015.
- [94] R. A. T. Leng, J. Hsieh, V. Mashayekhi, R. Rooholamini, An empirical study of hyper-threading in high performance computing clusters, Linux HPC Revolution 45 (2002).
- [95] L. Bononi, M. Bracuto, G. D'Angelo, L. Donatiello, Exploring the effects of hyper-threading on parallel simulation, in: 2006 Tenth IEEE International Symposium on Distributed Simulation and Real-Time Applications, DS-RT, Málaga, Spain, 2006, pp. 257–260. doi:10.1109/ DS-RT.2006.18.
- [96] X. Tian, A. Bik, M. Girkar, P. Grey, H. Saito, E. Su, Intel® openmp c++/fortran compiler for hyper-threading technology: Implementation and performance., Intel Technology Journal 6 (1) (2002).
- [97] J. González, J. Ortega, J. J. Escobar, M. Damas, A lexicographic cooperative co-evolutionary approach for feature selection, Neurocomputing 463 (2021) 59–76. doi:10.1016/j.neucom.2021.08.003.
- [98] D. E. Knuth, The Art of Computer Programming, Vol. 3, Pearson Education, 1997.
- [99] M. A. Goulet, D. Cousineau, The power of replicated measures to increase statistical power, Advances in Methods and Practices in Psychological Science 2 (2019) 199–213. doi:10.1177/ 2515245919849434.
- [100] H. Muhlenbein, D. Schlierkamp-Voosen, Optimal interaction of mutation and crossover in the breeder genetic algorithm, in: Proceedings of the Fifth International Conference on Genetic Algorithms, Citeseer, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993, p. 648.
- [101] J. Asensio-Cubero, J. Q. Gan, R. Palaniappan, Multiresolution analysis over simple graphs for brain computer interfaces, Journal of Neural Engineering 10 (4) (2013) 21–26. doi:10.1088/1741-2560/10/4/ 046014.
- [102] R. Kohavi, G. H. John, Wrappers for feature subset selection, Artificial Intelligence 97 (1-2) (1997) 273–324. doi:https://doi.org/10. 1016/S0004-3702(97)00043-X.
- [103] C. Sanhueza, F. Jiménez, R. Berretta, P. Moscato, Pasmoqap: A parallel asynchronous memetic algorithm for solving the multi-objective quadratic assignment problem, in: 2017 IEEE Congress on Evolutionary Computation (CEC), Donostia, Spain, 2017, pp. 1103–1110. doi:10.1109/CEC.2017.7969430.
- [104] M. Märtens, D. Izzo, The asynchronous island model and nsga-ii: Study of a new migration operator and its performance, in: Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation, GECCO '13, Association for Computing Machinery, 2013, pp. 1173—-1180. doi:10.1145/2463372.2463516.
- [105] O. Rice, R. E. Smith, R. Nyman, Parallel multi-objective genetic algorithm: Gpu accelerated asynchronously distributed nsga ii, in: Theory and Practice of Natural Computing: Second International Conference, Cáceres, Spain, 2013, pp. 217–227. doi:10.1007/ 978-3-642-45008-2_18.
- [106] F. A. Fortin, F. M. de Rainville, M. A. Gardner, M. Parizeau, C. Gagné, Tournament selection based on dominance, https: //deap.readthedocs.io/en/master/api/tools.html#deap. tools.selTournamentDCD, accessed: 2023-09-14 (2024).
- [107] A. F. Díaz, B. Prieto, J. J. Escobar, T. Lampert, Vampire: A smart energy meter for synchronous monitoring in a distributed computer system, Journal of Parallel and Distributed Computing 184 (2024) 104794. doi:10.1016/j.jpdc.2023.104794.
- [108] J. González, I. Rojas, J. Ortega, H. Pomares, F. J. Fernandez, A. F. Diaz, Multiobjective evolutionary optimization of the size, shape, and position parameters of radial basis function networks for function approximation, IEEE Transactions on Neural Networks (2003) 1478–1495doi: 10.1109/TNN.2003.820657.
- [109] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: Nsga-ii, IEEE Transactions on Evolutionary Computation 6 (2) (2002) 182–197. doi:10.1109/4235.996017.
- [110] J. León, J. J. Escobar, A. Ortiz, J. Ortega, J. González, P. Martín-Smith, J. Q. Gan, M. Damas, Deep learning for eeg-based motor imagery classification: Accuracy-cost trade-off, PLOS ONE 15 (2020) 1–

- 30. doi:10.1371/journal.pone.0234178.
 [111] J. G. Qi, G. R. Burns, D. K. Harrison, The application of parallel multipopulation genetic algorithms to dynamic job-shop scheduling, The International Journal of Advanced Manufacturing Technology 16 (2000) 609–615. doi:10.1007/s001700070052.
 [112] J. P. Cohoon, S. U. Hegde, W. N. Martin, D. Richards, Punctuated equilibring. A parallel agarting algorithm in: Genetic Algorithms and Their
- libria: A parallel genetic algorithm, in: Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms, Hillsdale, NJ: L. Erlhaum Associates, 1987., The Massachusetts Institute of Technology, Cambridge, MA, Boston, USA, 1987, pp. 148-154.

Multi-objective multi-population evolutionary procedures have become one of the most outstanding metaheuristics for solving problems characterized by the curse of dimensionality. A critical aspect of these models is the migration process, defined as the exchange of individuals between subpopulations every few iterations or generations, which has typically been adjusted according to a set of guidelines proposed more than 20 years ago, when the capacity to deal with problems was significantly less than it is today. However, the constant increase in computational power has made it possible to tackle today's complex real-world problems of great interest more plausibly, but with larger populations than before. Against this background, this paper aims to study whether these classical recommendations are still valid today, when both the magnitude of the problems and the size of the population have increased considerably, considering how this adjustment affects the performance of the procedure. In addition, the increase in the population size, coupled with the fact that multi-objective optimization is being addressed, has led to the development of a novel elitist probabilistic migration strategy that considers only the Pareto front. The results show some interesting and unexpected conclusions, in which other issues, such as the number of subpopulations or their size, should be considered when fitting multi-population models. Furthermore, some of the previously mentioned classical recommendations may not be well-suited for high-dimensional problems.

Juan Carlos Gómez-López received the M.Sc. in Computer Engineering from the University of Granada, Spain, in 2016. He is a Ph.D. student in the Department of Computer Engineering, Automation, and Robotics at the University of Granada. His research interests include evolutionary computation, machine learning, and energy-aware computing.

Manuel Rodríguez-Álvarez received the B.Sc. degree in Physics (Electronics) in 1986 and the Ph.D. degree in Physics (with distinction) in 2002, both from the University of Granada (Granada, Spain). He is currently Associate Professor at the Department of Computer Engineering, Automation and Robotics at the University of Granada. He is a member of the CASIP (Circuits and Systems for Information Processing) research group, he has been the director of 3 Ph.D. theses and principal investigator in 3 national research projects. He also collaborates with research facilities such as SKA and IFMIF-DONES involved in sub-nanosecond time transfer solutions based on White Rabbit. His current research interests include deterministic timing over optical fiber networks.

Daniel Castillo-Secilla received the Ph.D. degree in Communication and Information Technologies, with a specialization in Bioinformatics, from the University of Granada, Spain, in 2020. After his postdoctoral phase, he is a Senior Bioinformatician in the Center of Excellence of Data Intelligence at Fujitsu Technology Solutions. His research interests include cancer biomarker detection by using machine learning, feature selection optimization, and applied image pattern detection.

Jesús González received the M.Sc. and Ph.D. degrees in Computer Engineering from the University of Granada, Spain, in 1998 and 2001, respectively, both with distinction. He is a full professor in the Department of Computer Engineering, Automation, and Robotics, at the University of Granada. His research interests include evolutionary computation, machine learning, and embedded systems.

1

Daniel



Manuel



Juan Carlos



Jesus



There is not any conflict of interest.