Genetic Programming and Evolvable Machines (2024) 25:19
https://doi.org/10.1007/s10710-024-09492-4

A survey on dynamic populations in bio-inspired algorithms

Davide Farinati[1] · **Leonardo Vanneschi**[1]

Received: 10 March 2024 / Revised: 19 July 2024 / Accepted: 20 July 2024

Abstract

Population-Based Bio-Inspired Algorithms (PBBIAs) are computational methods that simulate natural biological processes, such as evolution or social behaviors, to solve optimization problems. Traditionally, PBBIAs use a population of static size, set beforehand through a specific parameter. Nevertheless, for several decades now, the idea of employing populations of dynamic size, capable of adjusting during the course of a single run, has gained ground. Various methods have been introduced, ranging from simpler ones that use a predefined function to determine the population size variation, to more sophisticated methods where the population size in different phases of the evolutionary process depends on the dynamics of the evolution itself and events occurring within the population during the run. The common underlying idea in many of these approaches, is similar: to save a significant amount of computational effort in phases where the evolution is functioning well, and therefore a large population is not needed. This allows for reusing the previously saved computational effort when optimization becomes more challenging, and hence a greater computational effort is required. Numerous past contributions have demonstrated a notable advantage of using dynamically sized populations, often resulting in comparable results to those obtained by the standard PBBIAs but with a significant saving of computational effort. However, despite the numerous successes that have been presented, to date, there is still no comprehensive collection of past contributions on the use of dynamic populations that allows for their categorization and critical analysis. This article aims to bridge this gap by presenting a systematic literature review regarding the use of dynamic populations in PBBIAs, as well as identifying gaps in the research that can lead the path to future works.

Keywords Population-based algorithms · Bio-inspired algorithms · Population size · Dynamic population · Adaptive population

Extended author information available on the last page of the article

Published online: 24 July 2024

1 Introduction

Population Based Bio-Inspired Algorithms (PBBIAs) are a class of computational methods inspired by the biological processes and behaviors observed in nature. These algorithms simulate the natural phenomena of evolution, social behavior, and biological systems to solve complex optimization and search problems. The core idea behind these algorithms is to maintain a population of potential solutions that evolve or adapt through mechanisms such as reproduction, mutation, and crossover in the context of evolutionary algorithms, or through the collective behavior and communication observed in swarms or flocks in the context of swarm intelligence algorithms. PBBIAs include a wide range of algorithms, each drawing inspiration from different biological processes. Some of the most prominent PBBIAs include Evolutionary Algorithms (EAs), inspired by the process of natural selection and genetics; Particle Swarm Optimization (PSO), based on the social behavior and movement dynamics of birds and fish; Ant Colony Optimization (ACO), which mimics the foraging behavior of ants; and Artificial Bee Colony (ABC) algorithms, inspired by the food foraging behavior of honey bees. The class of algorithms we refer to as PBBIAs is sometimes grouped under the broader term Evolutionary Computation (EC) in various bibliographic sources. However, it is also common practice to restrict EC to computational methods specifically inspired by Darwinian evolution. As a result, algorithms such as PSO, ACO, and ABC are often categorized under Swarm Intelligence (SI) rather than EC. For instance, [1] highlights that SI algorithms have expanded beyond the traditional scope of EC, establishing themselves as a distinct field. Therefore, we have opted to use the more inclusive term PBBIAs to encompass all biologically inspired population-based computational methods.

Since the early years of PBBIAs, the significance of population size as a critical parameter has been consistently demonstrated. For instance, different studies [2–4] show that population size significantly affects the performance of PBBIAs, and that selecting the correct value for this parameter is key for the success of the evolutionary process. In the initial implementations of PBBIAs, the population size was set as a constant value, remaining unaltered throughout successive generations. However, in the early 1990 s pioneering initiatives were launched to develop PBBIAs with dynamic population size. This choice was motivated by various reasons. For instance, different contributions [5–11] justify the use of dynamic populations in order to reduce the computational cost of PBBIAs, claiming that a steady decrease in the population size allows to save calculations while keeping a good quality of solutions. Others contributions [12–18] used dynamic populations to improve the performance of the algorithm by balancing exploration and exploitation. A different approach was the one presented in papers such as [19, 20], where dynamic populations are used to adapt the population to a dynamic modification of the environment, mainly caused by a dynamic change in the fitness function. When handling these types of optimization problems, dynamic target functions, different approaches have also been explored. These include maintaining diversity through methods like random immigrants and hypermutation, using memory schemes to retain and reintroduce good solutions, and employing multi-population approaches to ensure broad exploration.[21, 22] Adaptive parameter control and change detection mechanisms allow the algorithm to adjust its behavior dynamically. [23, 24] Together, these strategies enhance the algorithm's ability to cope with evolving optimization landscapes. A different perspective is the one presented in [25–29], where employing an adaptive population size contributes to the definition of an algorithm in which the user is relieved from the responsibility of setting the parameters (*parameter-free* algorithm). This approach aims to make the algorithms more user-friendly and robust by reducing the dependency on finely-tuned parameters, which can be difficult to set correctly without expert knowledge. Parameter-less algorithms typically employ adaptive or self-adaptive mechanisms, where parameters dynamically change based on the current state of the search process[23, 30],. For instance, some algorithms adjust mutation rates based on population diversity, ensuring a balance between exploration and exploitation [31]. Other approaches use heuristic rules or meta-optimization techniques to set parameters on-the-fly [32, 33]. These strategies not only simplify the deployment of optimization algorithms but also enhance their ability to adapt to various problem landscapes, ultimately improving the overall performance and robustness of the optimization process [34]. All dynamic population algorithms are characterized by some fundamental choices, such as determining the appropriate timing and extent of population size adjustments, as well as selecting the specific individuals to add or remove. Usually, the decisions about when and to what extent to modify the population are closely interlinked and depend on the same criterion, while the selection of the individuals to add or remove is an independent step. It is worth noting that while not all dynamic population algorithms introduced thus far contemplate the option of increasing the population size, all of them integrate a mechanism for decreasing the population.

The goal of this study is to provide an overview of the most promising dynamic population methods present in the literature, while also focusing on categorizing these studies and identifying possible gaps where future research could be conducted.

The paper is organized as follows: Sect. 2 describes the bibliographic search that lead to the collection of the contributions discussed in this study. Section 3 proposes a categorization of the existing dynamic population methods based into two axes: the methods employed for removing individuals from the population and the criterion used for selecting individuals to be removed. This classification serves as a foundation for a detailed discussion on the identified literature. Section 4 provides a further analysis on the presented studies, linking the categories presented in Sect. 3 to the most common motivation used by the authors to justify the employment of dynamic population. Finally, Sect. 5 summarizes the findings of this study, emphasizing the substantial advantages of utilizing dynamic populations in PBBIAs, contributing to a deeper understanding and furtherance of this research domain.

2 Bibliographic search methodology

The bibliographic search that provides the foundation of this work is based on two academic databases, Scopus and Web of Science. The keywords *"dynamic population"* are very common in many fields that differ from the one of this study, such as Biology and Medicine, among others. Therefore the search on the previously named databases was based on pairing *" dynamic population"* with some keywords that may address the search to our field of study, such as *" artificial intelligence"*, *" machine learning"* and more specifically, of course, *" evolutionary computation"* and *" evolutionary algorithms"*. Then, with the aim of not missing any important contribution, *" dynamic population"* was paired with the names of some of the most famous algorithms in the area, namely *" genetic algorithms"*, *" genetic programming"*, *" differential evolution"* and *" particle swarm optimization"*. To enhance the scope and variety of our exploration, we also conducted a parallel search using the keywords *" adaptive population"*. The inquiry has resulted in the identification of numerous scientific articles, which have undergone a meticulous selection process based on their alignment with the topic of this survey, as well as their overall significance. Consequently, we propose a comprehensive analysis of the 51 most consequential papers.

3 Categorization of the selected contributions

In this work, the classification of PBBIA studies incorporating dynamic populations is not constrained by a chronological timeline, as no specific pattern or trend has been discerned. While papers discussing the implementation of dynamic populations began emerging in the early 1990s, the subsequent years, especially from the 2000s onward, witnessed a consistent influx of new contributions without exhibiting any discernible trend or peak. Instead, the classification presented here is grounded in two primary factors: the criterion governing the selection of when and how many individuals are added or removed, and the criterion determining which individuals are selected for removal. The second factor of the categorization is based only on the criterion that chooses which individuals to remove and not on the addition of new individuals. This is because, out of the 51 papers used for this analysis, 14 did not implement the addition of new individuals to the population, and in the other 20 cases, the individuals added to the population were variations of those already in the current population.

For each of the two axes, we were able to identify the most common approaches, which are presented in Tables 1 and 2 respectively, together with the count of how many times they are used in the analyzed papers.

By looking at Table 1, we observe that the prevalent approach (21 contributions) for determining when and how many individuals are added or removed from the population is "based on fitness". In other words, the decision is taken using the fitnesses of the individuals that belong to the current population. The

Table 1 Most common approaches concerning the criterion determining when and how many individuals are added or removed from the population

Approach	Count
Based on fitness	21
Regulated by an *a priori* defined function	10
Based on diversity	7
Based on the number of fitness evaluations	6
Based on individual life span	5
Encoded in the population	1

Table 2 Most common approaches concerning the criterion responsible for selecting which individuals to remove from the population

Approach	Count
Based on fitness	26
Size of the offspring population changed	15
Based on the size of the individuals	4
Based on individual life span	2
Random individuals	1
Encoded in the population	1
Based on diversity	1

central concept typically involves an attempt to discern whether the evolutionary process is advancing effectively or encountering stagnation in local optima. Based on this, decisions are made regarding the addition or removal of individuals from the population. Also, in Table 1 we can see that in 10 studies the population size is regulated by applying an *a priory* defined function. The third most used approach (7 articles) is "based on diversity", meaning that individuals are added or removed to the population according to the population diversity measured during the evolution process. In four bibliographic references, the "number of fitness evaluations" was used as a criterion to decide whether to enlarge or shrink the population. These papers are characterized by the implementation of an overall budget of evaluations for the evolutionary process, and a consequent regulation of the population size. In five articles the approach involves using a life span that regulates how long each individual survives in the population. Finally, in one article [28] the population size is directly encoded in the chromosomes representing the evolving individuals.

Observing Table 2, we can see many shared approaches with the ones presented in Table 1, such as the approaches related to fitness, diversity, and life span, in which these values determine which individuals to remove from the population. Different methods use the size of the individuals as the driver to choose which individuals to remove from the population. This approach is typical of PBBIAs with variable size representations, like Genetic Programming (GP). Finally, in one article [35] the individuals to remove are chosen at random from the population.

Considering five distinct approaches for the criterion determining when and how many individuals to add or remove, and an additional six approaches for the criterion deciding which individuals to remove, the resulting grid would encompass 30 categories. In order to manage this complexity more effectively, we opted to aggregate certain approaches into groups. Regarding the horizontal axis (criterion that selects when and how many individuals to remove), we have decided to aggregate the approaches " Regulated by an *a priori* defined function", " Based on individual life span" and " Encoded in the population". Indeed, these three categories share the common characteristic of implementing an automatic approach to select when and how many individuals to remove, which does not take into account the features of the current population. Specifically, the first approach uses a function to determine the population size at any given generation. The second approach assigns each individual a maximum lifespan to each individual, removing them from the population after that lifespan is reached. The last approach encodes the population size within the algorithm, optimizing it as part of the overall process. A similar aggregation was performed for the vertical axis (criterion that selects which individuals to remove), grouping together the approaches " Size of the offspring population changed", " Based on individual life span", " Random individuals" and " Encoded in the population". Indeed, again these four approaches implement an automatic way of selecting which individuals to remove, without looking at the features of the current individuals.

Figure 1 presents the categories in the two-dimensional space, together with the count of how many of our selected articles fall into that category.

Observing Fig. 1, it is possible to notice that there are nine different categories of studies that contain at least one article. In the continuation, we propose a more in-depth analysis of the different categories, together with a discussion of the contributions that belong to them. Concerning terminology, we will designate the categories based on the approach they adopt along both the horizontal and vertical axes of Fig. 1. For instance, the term " Fitness Based - Automatic" denotes the category

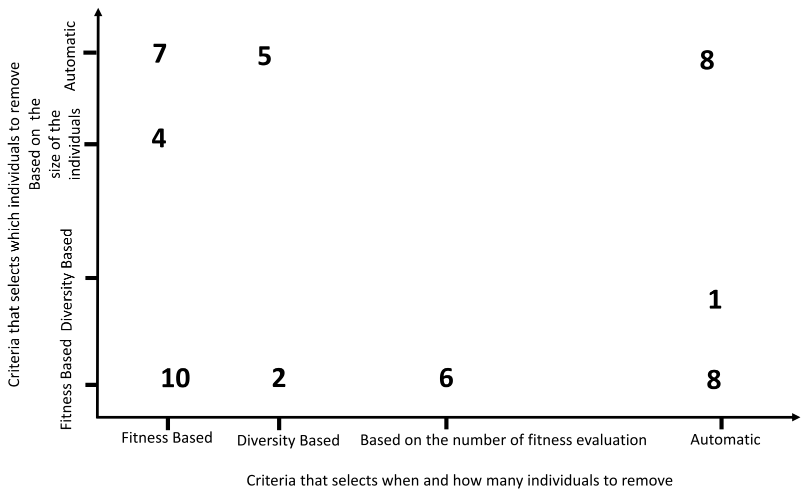

Fig. 1 Graphical representation of the two-dimensional categorization with the number of papers for each category

of studies that employ a " Fitness Based" approach for determining when and how many individuals to remove, and an " Automatic" approach for selecting which individuals to remove. The order in which the studies are discussed in each category is chronological.

3.1 Fitness based–fitness based

This category contains 10 papers, reported in Table 3, in which the criterion used to select when and how many individuals to remove and the criterion used to select which individuals to remove are both based on fitness.

The first contribution of this category, in chronological order, is represented by a study conducted by Yen et al. in 2003 [36]. This paper introduces a Multi-Objective (MO) Particle Swarm Optimization (PSO) system with a Dynamic Population (DP), termed Dynamic Particle Swarm Multiobjective Optimization (DPSMO). The removal of individuals is determined based on fitness. DPSMO regulates the population size according to the density and the number of dominated individuals in a specific neighborhood, also referred as cell rank. The density value indicates how many individuals occupy a cell in the objective space, while the cell rank reflects the relative quality of a cell based on its dominance and density. This algorithm implements a Population Growing Strategy and a Population Declining Strategy. In the first one, one crossover and mutation operations are designed to minimize rank values and maintain cell densities. The fitness assignment scheme divides the population into subpopulations responsible for rank and density optimization. Offspring are retained in the next generation if they have better fitness than their parents, ensuring exploration of unexplored cells. However, a forbidden region concept is introduced to prevent undesirable backward drifting of offspring. On the other hand, in the Population Declining Strategy, individuals are removed based on their cell rank and density values, and an age threshold is introduced to ensure each individual contributes meaningfully. This is done by assuring that any individual survives in the population for at least a number of generations equal to the age threshold.

A similar approach is presented in the work of Leong et al., published in 2008 [37], in which another MO optimization problem is tackled with the use of PSO. This algorithm uses the same DP techniques as in the first paper, while also using an Adaptive Local Archive (ALA) in order to improve the diversity within each swarm. The ALA technique optimizes multiobjective problems by maintaining diverse and high-quality solutions. It clusters solutions into local archives, dividing the objective space into adaptive cells to guide exploration. This dynamic adjustment promotes solution diversity, focusing on less crowded areas to avoid premature convergence. ALA enhances the Particle Swarm Optimization framework by ensuring a balanced exploration of the search space, leading to more efficient and effective problem-solving. A different approach is presented in [38], where Zhnag et al. apply DP to Differential Evolution (DE). In this experiment, an adaptive mechanism is applied to the DE algorithm, aimed at enhancing its performance on large-scale global optimization problems. This is done by introducing two strategies: population increasing (pop_inc) and population decreasing (pop_dec). The pop_inc

Table 3 Summary of the Papers presented in Sect. 3.1

Year	Authors	Title	Reference
2003	Yen and Lu	Dynamic population strategy assisted Particle Swarm Optimization	[36]
2008	Leong and Yen	PSO-based multiobjective optimization with dynamic population size and adaptive local archives	[37]
2009	Zhang and Zhan	Adaptive population differential evolution with dual control strategy for large-scale global optimization problems	[38]
2013	Zhu et al.	Adaptive population tuning scheme for differential evolution	[39]
2013	Wang and Zhao	Differential evolution algorithm with self-adaptive population resizing mechanism	[40]
2014	Montiel et al.	Intelligent control of dynamic population size for evolutionary algorithms	[13]
2017	Wong et al.	Continuous adaptive population reduction (CAPR) for differential evolution optimization	[29]
2017	Cui et al.	A novel artificial bee colony algorithm with an adaptive population size for numerical function optimization	[41]
2023	Farinati et al.	A study of dynamic populations in geometric semantic genetic programming	[10]
2023	Shu et al.	Multi-objective particle swarm optimization with dynamic population size	[18]

strategy allows trial solutions that are not initially better than existing solutions to be retained, potentially increasing diversity and aiding in escaping local optima. The pop_dec strategy involves removing solutions with a high " degradation value," indicating poor performance or stagnation of the algorithm, to maintain an optimal population size and computational efficiency.

Another application of DP to DE is the one presented by Zhu et al. in [39], where redundant individuals are removed from the population according to their fitness value. This work presents an Adaptive Population Tuning Scheme (APTS), which dynamically adjusts the population size based on the current solution-searching status. APTS employs a status monitor to track the progress of individuals and two main strategies: an inferior-based population-cut strategy to remove poorly performing solutions, and an elite-based population-incremental strategy to add promising new individuals. These promising new solutions are generated by sampling trial vectors from a candidate pool, using the probability learned from its success rate. This process continuously removes redundant particles reducing the computational cost, while enhancing the optimization process by ensuring the algorithm adapts to the evolving search landscape efficiently. In an attempt at introducing a parameter-less version of DE, Wang et al. introduce a Self-adaptive population resizing mechanism named SapsDE in [40]. Population reducing or augmenting strategy are activated according to the evolutionary process's success rate, enhancing the balance between exploration and exploitation. The reduction strategy focuses on removing individuals that are considered to be performing poorly. Specifically, the approach uses an inferior-based population-cut strategy to identify and remove the worst-performing particles. The approach also uses an augmenting strategy, that introduces new individuals. Those individuals are mutations of the ones already present in the population. The approach aims to achieve faster convergence and better solution quality, particularly beneficial for large-scale and challenging optimization tasks. In [13], the concept of Mediative Fuzzy Logic is used to calculate the population size at each generation for EAs. By intelligently increasing or decreasing the population size based on the algorithm's performance and the landscape's characteristics, exploiting fuzzy logic. This method aims to enhance precision and reduce computational time compared to traditional approaches.

The last application of DP to DE presented in this section is the one introduced by Wong et al. in [29]. The paper introduces the Continuous Adaptive Population Reduction (CAPR) method for Differential Evolution (DE) optimization, focusing on dynamically reducing the population size in accordance with the optimization progress. This adjustment is based on the optimization performance gradient measured over generations. After each evaluation cycle, the average fitness values of the population are used to compute a normalized gradient value, which then determines the new population size for the next generation. This novel approach aims to enhance efficiency and convergence by continuously adjusting the population size, independent of the DE structure, allowing integration into various DE variants. This method is implemented to balance the exploration and exploitation phases of the evolutionary process. In a work presented by Cui et al. [41], DP is applied to the Artificial Bee Colony (ABC) algorithm, introducing an Adaptive Method for Population Size (AMPS). This method adjusts the population size based on the success in finding better solutions. If the algorithm excels in exploration, AMPS reduces the population by removing the less promising solutions, thus enhancing exploitation. Conversely, if the algorithm is more successful in exploitation, it increases the population by adding solutions from an external archive to boost exploration. Also in this case, this adaptive adjustment aims to maintain a balance between exploration and exploitation, improving the algorithm's performance on numerical function optimization problems.

In [10], DP strategies are adapted to a variation of Genetic Programming (GP), namely Geometric Semantic Genetic Programming(GSGP), in order to reduce the overall computational cost of the algorithm. This works adapts to GSGP a DP strategy that had initially been introduced for Genetic Algorithms (GAs), ProFIGA [17], together with one that had originally been proposed for standard GP [19]. These techniques are later described respectively in Sects. 3.2 and 3.5. A new DP method is also introduced in [10]. It differs from its predecessor, introduced in [19], by employing the Total Improvement Efficiency (TIE) criterion to assess algorithm stagnation. Unlike the method introduced in [19], which relies on fitness comparison between generations, TIE-DP-GSGP uses the Training Improvement Effectiveness (TIE) value, based on the concept of Semantic Neighborhood(SN) [42], to make decisions about population adjustments. TIE is calculated as the number of individuals present in the SN of the elite, with a better fitness than the elite itself. At each generation, the TIE value is compared to a predefined tolerance threshold. If the TIE is below this threshold, indicating minimal improvement or stagnation, a number of individuals, determined by a specific equation [19], are removed from the population to encourage exploration and avoid premature convergence. Conversely, if the TIE exceeds the threshold, suggesting potential for further exploration, the same number of individuals are added to the population, enhancing the algorithm's search capabilities. A variant with an external archive to reintroduce previously eliminated individuals is also implemented. The last work presented in this section is the study performed by Shu et al. [18], where DP is applied to PSO. This approach adapts the population size based on the archive's resources, increasing particles for better exploration when needed and using non-dominated sorting and density control to prevent excessive population growth. This dynamic adjustment aims to enhance convergence and diversity in solving complex optimization problems while maintaining a balance between exploration and exploitation.

The studies showcased in this section illustrate the efficacy of using fitness as a criterion to determine when, how many, and which individuals to remove from the population. These methods yield various outcomes, resulting in improved equilibrium between exploration and exploitation, a reduction in the algorithm's overall cost, and a possible technique for adapting to dynamic target functions. While significant progress has been made, future research should focus on addressing the limitations identified, such as the scalability of algorithms in large-scale problems and their adaptability in dynamic environments. Comparative studies involving real-world datasets across diverse domains can also offer valuable insights and drive innovation in the field.

3.2 Fitness based–automatic

This category contains 7 papers, presented in Table 4. The paper presented in this section use as criterion to select when and how many individuals to remove fitness, while having an automatic approach in choosing which to remove.

All the algorithms that belong to this category eliminate individuals by reducing the size of the offspring population, therefore the focus will be on how they choose when and how many individuals to remove. The first study, presented by Liu et al. in [43], uses a Fuzzy Logic Controller (FLC) to regulate the population size of DE at each generation. The controller adjusts the population size based on the variance changes of objective function parameters (v_{pc}) and objective function values (v_{fc}). These changes are monitored over two successive generations and are used as inputs to the FLC. If the absolute value of v_{pc} is large, indicating a significant distance from the optimal solution, it is set to be large, suggesting an increase in population size to enhance exploration. Conversely, if the absolute value of v_{fc} is small, indicating proximity to the optimal solution, it is set to be small, advocating for a reduction in population size to focus on exploitation. The aim is to enhance performance by optimizing the population size in response to the search process's current state, potentially leading to faster convergence and reduced computational load. In the work by Eiben et al. [17] PRoFIGA, a new algorithm that applies DP to GAs, is introduced. Individuals are added to the population either if the evolutionary process is working (i.e., the fitness improved since the last generation) or if it got stuck for a long time (i.e., the fitness did not improve for five generations). In all the other cases, the population size is reduced. PRoFIGA is shown to outperform the standard version of the GAs on several test problems. The effectiveness of the previous approach is also confirmed by Merchàn-Cruz et al. in [44], where PRoFIGA is applied to Robot manipulators. The benefits of employing a dynamic population are also discussed in [45], where two novel methods for adaptive parent population sizing are introduced. The first method estimates quality gain from mutation vectors on simple fitness functions like the sphere model, to dynamically adjust the parental population size for maximum quality gain. The second method, suitable for general fitness functions, uses evolutionary directional derivatives to adapt the parental population size, aiming to optimize the algorithm's path towards fitness improvement. Both methods rely on the mutation vectors and fitness values of each generation, striving for a balance between exploration and exploitation by adjusting the population size based on the current generation's performance.

In [46], DP is applied to the Fireworks Algorithm (FWA), adjusting the population size based on the search results of the current generation. When the optimal individual is updated, the population size is decreased linearly to enhance exploitation. Conversely, if the population is trapped in local minima, the size is increased randomly beyond the initial size to aid in escaping and foster exploration. This approach aims to balance between exploitation and exploration, thereby accelerating FWA's convergence and performance, particularly in high-dimensional problems. In the work of Liu et al. [47], the adaptive PSO with dynamic population (DP-APSO) algorithm is introduced. In this algorithm, an Evolutionary State Estimation (ESE) technique is implemented to recognize the current evolutionary

Table 4 Summary of the Papers presented in Sect. 3.2

Year	Authors	Title	Reference
2003	Liu and Lampinen	Population size adaptation for differential evolution algorithm using fuzzy logic	[43]
2004	Eiben et al.	Evolutionary algorithms with on-the-fly population size adjustment	[17]
2008	Merchán-Cruz et al.	GA based trajectory planner for robot manipulators sharing a common workspace with adaptive population size	[44]
2015	LaPorte et al.	Adaptive parent population sizing in evolution strategies	[45]
2020	Yu and Takagi	Accelerating fireworks algorithm with dynamic population size strategy	[46]
2021	Liu et al.	Adaptive particle swarm optimization with dynamic population and its application to constrained engineering design optimization	[47]
2022	Zheng and Luo	Adaptive differential evolution algorithm based on fitness landscape characteristic	[48]

state of the optimization search. The evolutionary states include exploration, exploitation, convergence, and jumping out. The population size modifies according to the current state detected by the ESE, increasing to maintain diversity when the algorithm is converging and decreasing to conserve computational resources when the algorithm is 'jumping out' of local optima. In [48], DP is applied again to DE. The population is adapted according to the local fitness landscape, using a method called Fitness Landscape Adaptive Population Sizing (FL-APS). The adjustment formula is: $N_{G+1} = round(N_{init} - N_{min} x\phi + N_{min})$. Here, N_{init} and N_{min} represent the minimum and initial population sizes, respectively. The parameter ϕ reflects the local fitness landscape's characteristics, varying between 0 and 1, where values close to 0 indicate a unimodal landscape (promoting faster convergence with a smaller population) and values close to 1 indicate a multimodal landscape (requiring a larger population to maintain diversity for exploration). This approach allows the algorithm to maintain a balance between exploring new areas of the search space and exploiting known good solutions, leading to better performance on various optimization tasks.

The works presented in this section show how employing an automatic approach to select which individuals to remove paired with using fitness as criterion to choose when and how many to discard yield good results when focusing on improving the equilibrium between exploration and exploitation or on reducing the algorithm's overall cost. Even if the presented approaches yield good results, the authors think that employing a *smart* criterion to choose which individuals to remove from the population could even improve their performance, helping the DP method to improve the evolution process.

3.3 Automatic–fitness based

This category encompasses a total of eight selected papers, presented in Table 5, that have an automatic criterion to choose when and how may individuals to remove, while using fitness to choose which ones to discard.

The first study within this category is conducted by Costa et al. [49], which explores a novel resizing technique to evaluate the advantages of DP on EAs. All the strategies presented involve adjusting the population size after a set number of fitness evaluations (500 and 5000 being the values tested). The individuals selected for removal are consistently the poorest performers in the population. The randomized alteration of population size is shown to outperform the standard algorithm with fixed population size version when no prior information is available. In [25], an application of adaptive population to GAs is proposed in order to achieve a parameter-less configuration. This study, by Back et al., takes inspiration from [52], using a life span for each individual. Compared to the original work, the main formula is modified, using information concerning the current state of the search. Individuals with a better fitness than the average of the population are given a longer lifespan, making the decision of which individuals to remove from the population based on fitness.

The study conducted by Fernandez et al. [5] implements a linear reduction approach to the population size in GP. In each generation, a fixed number of

Table 5 Summary of the Papers presented in Sect. 3.3

Year	Authors	Title	Reference
1999	Costa et al.	An Experimental study on dynamic random variation of population size	[49]
2000	Bäck et al.	An empirical study on GAs "without parameters"	[25]
2003	Fernandez et al.	The effect of plagues in genetic programming: a study of variable-size populations	[5]
2007	Sun et al.	Dynamic population size based particle swarm optimization	[50]
2007	Brest and Sepesy Maucec	Population size reduction for the differential evolution algorithm	[26]
2013	Rajakumar and Aloysius	APOGA: an adaptive population pool size based genetic algorithm	[27]
2017	Cui et al.	A novel artificial bee colony algorithm with an adaptive population size for numerical function optimization	[14]
2019	Lima et al.	Designing combinational circuits using a multi-objective cartesian genetic programming with adaptive population size	[51]

individuals are eliminated from the population, aiming to decrease the computational burden of the algorithm. The individuals selected for removal are the least fit individuals in the population. The Dynamic Population Particle Swarm Optimization (DPPSO) introduced by Sun et al. extends the concept of population adjustment in PSO with a novel approach. In DPPSO [50], the population size is reduced over time with an attenuation factor, aiming to decrease computational costs as the particles converge. Concurrently, it incorporates an undulate factor that introduces new particles during certain phases to maintain diversity and prevent premature convergence. This dual approach optimizes the balance between exploration and exploitation, improving the algorithm's efficiency and efficacy on benchmark functions.

In the study by Brest et al. [26], a linear reduction method is applied to DE for population size control. The equation governing population size across generations is devised by imposing a maximum limit on fitness evaluations throughout the evolution and ensuring that only half of the individuals are retained when the population is reduced. The selection process for choosing individuals in the reduced population operates similarly to the original DE selection, favoring individuals with higher fitness values for survival. This experiment integrates dynamic population (DP) to strike a balance between exploration and exploitation, allowing for a larger population size at the onset of evolution and a smaller one towards the conclusion. In [27], the life-time algorithm presented in [25] is used, demonstrating that it outperforms the standard GA. The concept of an individual lifetime is also applied to the ABC algorithm in [14]. The usage of DP is motivated by the fact that standard ABC performs badly in exploitation. The algorithm developed by Cui et al. enhances exploitation by periodically removing the worst solutions with low success rates, while adding to the population reserved solutions to improve exploration. In [51], DP is applied to Cartesian GP (CGP) in an attempt at reducing the computational effort of the algorithm. The DP strategy employed in this work allows the population to grow up to a maximum number based on the number of non-dominated solutions, enhancing the exploration of the search space. When the population exceeds a predefined threshold, the least promising solutions are eliminated based on crowding distance, maintaining diversity and computational efficiency. This leads to the creation of efficient circuit designs.

The works presented in this section show how employing an automatic approach to select when and how many individuals to remove paired with using fitness as criterion to choose which to discard can be applied to different scenarios. These approaches were used to achieve a better balance better exploration and exploitation, to reduce the computational effort of the algorithm and to achieve a parameterless configuration of the algorithm.

3.4 Automatic–automatic

This category contains 8 contributions, reported in Table 6. The works presented in this section utilize an automatic approach in order to select when and how many individuals to remove as well as to select which individuals to discard.

Table 6 Summary of the Papers presented in Sect. 3.4

Year	Authors	Title	Reference
1994	Arabas et al.	GAVaPS–a genetic algorithm with varying population size	[52]
2003	Sean et al.	Population implosion in genetic programming	[53]
2006	Cutello et al.	Clonal selection algorithm with dynamic population size for bimodal search spaces	[54]
2008	Ma and Krings	Dynamic populations in genetic algorithms	[55]
2008	Teng et al.	Self-adaptive population sizing for a tune-free differential evolution	[28]
2018	Hu et al.	Improving monarch butterfly optimization algorithm with self-adaptive population	[56]
2019	Li	An adaptive surrogate assisted differential evolutionary algorithm for high dimensional constrained problems	[57]
2023	Liang	A dual-population constrained multi-objective evolutionary algorithm with variable auxiliary population size	[11]

In [52], the Genetic algorithm with Varying Population Size (GAVaPS) is presented. GAVaPS uses an adaptive population regulated by the individuals lifetime. The lifetime is determined using the individual fitness value at its creation. After expiring its lifetime, the individual dies off and it is removed from the population. This approach is developed trying to balance the convergence and the computational cost of the algorithm. Sean et al.[53] introduce a dynamic population technique called " population implosion," where the population size is gradually decreased throughout the genetic programming run. This method reallocates evaluations from later to earlier generations, improving efficiency by emphasizing early exploration and later exploitation. The technique consistently performed as well as or better than traditional fixed-size population layouts, which maintain a constant number of individuals throughout the run, across various problem domains and evaluation lengths, also offering the benefit of reduced memory consumption by mitigating bloat. The work by Cutello et al. [54] applied a similar concept to the Immune Algorithm (IA). Each individual has an age, that is the number of generations it has been part of the population, that is transmitted from parents to offsprings. If an individual undergoes a successful mutation, i.e. improving the fitness value, the age will be reset to 0. A maximum age is determined and when an individual reaches it, it is removed from the population. In [55], a study of six different functions to control the population size of GA through generations is presented. The functions' only input is the generation number, and the size of the population is regulated by adapting the offspring population size. The conclusion of the study conducted by Ma et al. is that there is not a universally applicable formula for sizing the population, a conclusion corroborated by the No Free Lunch Theorem [58]. In an attempt at achieving a parameter-less version of the DE algorithm, Teng et al. [28] encoded the parameters of the algorithm, including the population size, in the chromosome that represents the evolving individuals. With this approach, the population size is regulated by the evolutionary process, subject to selection, crossover and mutation. This solution was showed to have good performance among most of the studied benchmarks, saving as well the effort needed to tune the parameters.

Similarly to what presented in [55], the work by Hu et al. [56] uses a previously defined function to regulate the population size of the Monarch Butterfly Optimization (MBO) algorithm. The function that regulates the number of butterflies in land 1 and land 2 takes as inputs an upper and lower bound provided by the user, together with the maximum generation. This allows to control the population size and adjust it during the evolutionary process. In [57], a function that regulates the population size of DE to tackle high dimensional constrained problems is introduced. The formula that regulates the population size is based on user defined boundaries and on the current number of fitness evaluations. This approach is able to control the exploring and exploiting states of the evolution process. In [11] a dual-population constrained multi-objective evolutionary algorithm with a dynamic auxiliary population size is presented. This technique involves two populations: the main one evolves typical solutions, while the auxiliary supports it by dynamically adjusting its size based on problem complexity and evolutionary progress. Particularly, it varies in relation to the exploration of Unconstrained Pareto Fronts (UPFs). As the evolutionary process progresses and the main population's need for diverse solutions decreases, the size of the auxiliary population is reduced to allocate more computational resources to the main population, enhancing its efficiency in exploring Constrained Pareto Fronts (CPFs). This approach enhances diversity and adaptability, potentially leading to more efficient and effective problem-solving in complex multi-objective optimization scenarios.

The research presented in this section demonstrates the utilization of an automatic approach to select when, how many and which individuals to remove paired. These methodologies have been applied across diverse contexts to achieve several objectives, including optimizing the balance between exploration and exploitation, reducing computational complexity, and realizing parameterless configurations of algorithms. The presented *full-automatic* methods yield good performance, although it would be beneficial to understand why this is happening. Understanding the underlying process of these DP methods could help to formalize it and improving it.

3.5 Fitness based–based on the size of the individuals

All the contributions classified in this category apply DP to GP. This is due to the fact that the size of the individuals of GP changes through the evolution, while in other PBBIAs, like GAs and DE, it is fixed. The contributions that belong to this are category are reported in Table 7. The works that belong to this category use fitness as a criterion to select when and how many individuals to remove from the population, while using the size in order to select which individuals to remove.

The first work within this category [7] presents a variation of the algorithm described in [5] (and discussed in Sect. 3.3), where a linear reduction in population size was employed. The Rochat et al. algorithm introduces a mechanism to adjust the population size by adding or removing individuals based on the progression of fitness during evolution. When fitness stagnates, individuals are added, whereas if fitness improves, individuals are removed. The individuals that need to be removed are chosen according to both their fitness and their size. To suppress n individuals, at first the $2n$ individuals with the worst fitness are determined. Among those, the n with the largest size are then suppressed. In [8, 9], a variation of this algorithm is presented. The work of Kouchakpour et al. introduces four new different stagnation phase assessment methods, improving the adaptive population size mechanism. Similarly, Tao et al. propose a modification of the population variation scheme, that shows an acceleration of convergence. The last work presented in this section [19]

Table 7 Summary of the Papers presented in Sect. 3.5

Year	Authors	Title	Reference
2005	Rochat et al.	Dynamic size populations in distributed genetic programming	[7]
2009	Kouchakpour et al.	Dynamic population variation in genetic programming	[8]
2011	Vanneschi and Cuccu	Reconstructing dynamic target functions by means of genetic programming using variable population size	[19]
2012	Tao et al.	Genetic programming using dynamic population variation for computational efforts reduction in system modeling	[9]

utilizes DP to adapt to dynamic target functions. The resize mechanism recognizes whether the target function is changing by looking at the fitness of the population. If the target function is in a static phase, then individuals are removed from the population, with the same mechanism as the one presented in [7] and discussed above. Otherwise, new random individuals are added to the population to reinforce the exploration phase.

The works presented in this section show how employing fitness as criterion to select when and how many individual to remove, together with the size of individuals as criterion to determine which individuals to discard from the population is very effective when it comes to reducing the computational effort of the algorithm. This approach is especially focused on algorithms that utilize individuals with a varying size. It would be beneficial to extend the *non-size dependant* driver of these approaches to other algorithms, where the size of the individuals does not vary though the evolution.

3.6 Diversity based–automatic

This category contains five contributions, presented in Table 8, that all employ diversity as criterion to select when and how many individuals to remove, while having an automatic approach in choosing which individuals to discard.

In [15], the Incrementing Multiobjective Evolutionary Algorithm (IMOEA) is introduced. This approach leverages an online adaptation mechanism based on the discovered Pareto front and its desired population distribution density. By adjusting the population size in real-time, the algorithm ensures an efficient exploration of the search space and a uniform distribution of solutions along the Pareto front. This dynamic population strategy, coupled with adaptive niche induction and the Fine-Grained Boundary Local Perturbation (FBLP) method, facilitates broader neighborhood explorations and eliminates gaps along the Pareto front. The algorithm incorporates a Switching Preserved Strategy (SPS) to maintain stability and diversity among the Pareto front solutions. The work, conducted by Tan et al., shows how IMOEA performs well compared to other MO methods. Contribution [59] introduces two novel adaptive population sizing methods for the Univariate Marginal Distribution Algorithm (UMDA) in both continuous and discrete domains. These methods dynamically adjust the population size based on the algorithm's performance and the problem's complexity, aiming to optimize the balance between exploration and exploitation. The adjustment is guided by the relationship between population size and problem size, with the population size being proportional to the problem's volume. In both domains the population size for the next generation is calculated by considering the volume of the space occupied by the population. The mechanism takes into account the density function from which individuals are generated, adjusting the population size according to the spread of individuals within the problem space. Similarly, the algorithm presented in [60] adapts the GA population size to the " rate of accepting non-synonymous to synonymous genetic changes", in order to use GA for " WiMAX network planning", a dynamic optimization problem. This novel approach incorporates a population adjustment method

Table 8 Summary of the Papers presented in Sect. 3.6

Year	Authors	Title	Reference
2001	Tan et al.	Evolutionary algorithms with dynamic population size and local exploration for multiobjective optimization	[15]
2005	Hong et al.	Adaptive population size for univariate marginal distribution algorithm	[59]
2010	Hu et al.	WiMAX network planning using adaptive-population-size genetic algorithm	[60]
2014	Shi et al.	Differential evolution with adaptive population size	[35]
2015	Choi and Ahn	An adaptive population resizing scheme for differential evolution in numerical optimization	[61]

inspired by neutral theory from molecular biology, aiming to enhance search ability by dynamically modifying the population size based on the rate of accepting genetic changes. This adjustment facilitates the acceptance of new variations, especially neutral or nearly neutral ones, thereby accelerating the evolutionary process.

The last two contributions presented in this section focus on the application of DP to DE. In [35] the population size is adjusted accordingly to the evolutionary feedback, enhancing exploration and exploitation capabilities. This method considers both the standard deviation of the population together with the displacement of the center of mass, in order to efficiently adjust the population size. In particular, if the global minimum is near or outside the current population's boundary, indicating a quick movement of the population's center of mass with a large variance, the population size is increased to prevent premature convergence. Conversely, if the global minimum is close to the center of mass, resulting in slow movement and large variance, the population size is decreased to accelerate convergence. The work presented by Choi et al. [61] introduces the Adaptive Population Resizing (APR) scheme. This approach calculates the deviation of scattered individuals across search spaces, determining whether the population size needs to be decreased or increased. The deviation is calculated using the average positions of scattered individuals over each dimension. If the deviation is significantly less than the previous one, suggesting that individuals have found a promising region, the population size is decreased. Conversely, if the deviation is not sufficiently reduced, indicating a lack of population diversity and the need for further exploration, the population size is increased.

This section presents works that use diversity as criterion to choose when and how many individuals to remove pairing it with an automatic approach that determines which individuals will be discarded from the population. This approach is mostly employed to achieve a better balance between exploration and exploitation. The works presented in this section show how diversity is crucial for the evolution process. Future research should aim to develop more scalable niche-based methods and explore their application in large-scale and real-world problems. Additionally, combining niche-based approaches with other population management techniques could enhance their efficiency and effectiveness.

3.7 Based on the number of fitness evaluations–fitness Based

This category contains six contributions, reported in Table 9. The works presented in this category utilize the number of fitness evaluations in order to choose when and how many individuals to remove, while employing fitness as criterion to select which individuals to discard.

The first one in chronological order is represented by the work of Silva et al. [62], in which a resource limit is imposed to the evolution process of GP. A maximum number of nodes for each generation, constant through the evolution process, is set. At each generation, the survival of the individuals is dictated primarily by the fitness, with size playing a secondary role. The selection process for the next generation starts with the offspring, followed by their parents. These groups are ranked based on fitness without considering their size. Resources (in terms of the number

Table 9 Summary of the Papers presented in Sect. 3.7

Year	Authors	Title	Reference
2005	Silva et al.	Resource-limited genetic programming: replacing tree depth limits	[62]
2005	Silva and Costa	Resource-limited genetic programming: the dynamic approach	[63]
2014	Tanabe et al.	Improving the search performance of SHADE using linear population size reduction	[64]
2018	Piotrowski	L-SHADE optimization algorithms with population-wide inertia	[65]
2018	Awad et al.	Ensemble of parameters in a sinusoidal differential evolution with niching-based population reduction	[16]
2022	Wang et al.	Evolutionary algorithm with dynamic population size for constrained multiobjective optimization	[12]

of nodes) are allocated to the queued candidates on a first-come, first-served basis. Those individuals who need more resources than what is available are bypassed (and thus do not make it to the next generation), and this process of allocation goes on until the queue is exhausted or the limits on the population size are reached.

The second paper [63] is a followup of the previously presented work. The selection process is the same as the one implemented in [62], with the only difference of the resource limit being dynamic instead of static. A flexible resource limit is applied, starting at a low threshold and increasing when it leads to enhanced average fitness across the population. This system offers previously excluded individuals a second opportunity for consideration. Each rejected individual is reevaluated for potential addition to the new generation, with acceptance contingent upon their contribution to elevating the average population fitness. This enhancement can be measured against the highest mean fitness achieved during the run, or compared to the average fitness of the preceding generation. Subsequently, Tanabe et al. [64], introduce a linear population size reduction to the Success-History based Adaptive DE (SHADE) algorithm [66]. The new algorithm (L-SHADE) implements a linear population size reduction, that decreases the population size linearly as a function of the number of fitness evaluation, removing the worst individuals. Contributions [16, 65] use the previously presented L-SHADE algorithm, adding new features to enhance its performance. The work by Piotrowski et al. [65] adds a population-wide inertia term to the mutation strategy, while the work by Awad et al. [16] introduces an ensemble sinusoidal adaptation. All three articles [16, 64, 65] show how the use of DP can be beneficial for DE. The last work of this category [12] applies DP to PBBIAs in order to solve MO constraint optimization problems. In the work, presented by Wang et al., the population size at each generation is regulated according to the current number of fitness evaluations. This allows the population to continually decrease, encouraging the exploration in the early stages and promoting exploitation in the later ones.

This section presents works that use the number of fitness evaluations as criterion to choose when and how many individuals to remove, together with fitness as criterion to determine which individuals will be discarded from the population. This approach is mostly employed to achieve a better balance between exploration and exploitation and to reduce the overall computational effort of the algorithm. The presented approaches show great improvements, especially regarding the computational cost aspect, although it would be important to keep some focus on the diversity of the population. Lack of diversity often leads to premature convergence.

3.8 Diversity based–fitness based

This category contains two articles, reported in Table 10, both of them employ diversity in order to choose to choose when and how many individuals to remove, while employing fitness as criterion to select which individuals to discard.

In [67], DP is applied to PSO, adjusting the population size automatically according to the value of diversity of the population at specific intervals, called ladders. If the diversity is above a threshold, the population size decreases by removing particles

Table 10 Summary of the Papers presented in Sect. 3.8

Year	Authors	Title	Reference
2009	DeBao and ChunXia	Particle swarm optimization with adaptive population size and its application	[67]
2019	Poláková et al.	Differential evolution with adaptive mechanism of population size according to current population diversity	[68]

with lower scores. Conversely, if the diversity is below the threshold, new particles are added through crossover to enhance diversity. The use of an adaptive population size is motivated by the fact that a large population size is not needed in the end of the evolutionary process, since the diversity of the population decreases while the generations increase. The second paper of this category applies DP to DE proposing a new mechanism based on the population diversity [68]. According to population diversity, individuals will be added or removed from the population. This approach works in a similar way to the one presented earlier in this section, adapting the diversity measure to the DE algorithm.

This section presents works that use diversity as criterion to choose when and how many individuals to remove, together with fitness as criterion to determine which individuals will be discarded from the population. This method is primarily utilized to attain an improved equilibrium between exploration and exploitation while concurrently diminishing the computational overhead of the algorithm. The DP approach based on diversity presented on both these works has great upsides, and it could be resourceful to extend it to other algorithms on top of PSO and DE.

3.9 Automatic–diversity based

This category contains only one contribution, reported in Table 11. The work presented in this section utilizes an automatic approach in order to choose when and how many individuals to remove, while employing diversity as criterion to select which individuals to discard.

In the work proposed by Fernàndez et al. [6], the population size is reduced linearly through generations removing a fixed amount at each generation. To mitigate the loss of diversity in the DP approach, a strategy is implemented where the individuals for removal are selected from distinct regions of the fitness landscape, while ensuring the retention of the best-performing ones. This approach aims to achieve a balance between eliminating less favorable individuals and preserving the diversity of the population.

Table 11 Summary of the Papers presented in Sect. 3.3

Year	Authors	Title	Reference
2003	Fernandez et al.	Saving computational effort in genetic programming by means of plagues	[6]

4 Further analysis and categorization

In order to add more details to the analysis we present two extra tables, Tables 12 and 13. The first presents the count of the number of papers included in this study for each PPBIA presented. While the latter associates each category aforementioned to the motivation that led the author to employ this strategy.

Upon examination of the former table, it is not possible to discern a clear trend linking the categories to the underlying motivations that justify them. However, we can observe that all the categories except " Fitness Based - Based on the size of the individual" and " Automatic - Diversity Based" have at least one published contribution that motivates the use of dynamic populations to balance exploration and exploitation during the evolutionary process. On the other hand, the category " Fitness Based - Based on the size of the individual" is mainly (3 cases out of 4) used to reduce the computational cost of the algorithm.

5 Conclusions

In this article, we have conducted a comprehensive literature review on the management of dynamic populations of bioinspired algorithms. Through thorough research utilizing Scopus and Web of Science, we have identified and selected 51 relevant contributions. Prior to presenting each of these publications, we undertook a classification, pinpointing two fundamental axes: the methods for removing individuals from the population and the criterion for selecting individuals to be removed. Analyzing the 51 selected publications, we opted to assign four distinct classes for each of these axes: the methods from removing individuals from the population were characterized as fitness based, diversity based, based on the number of fitness evaluations and automatic. The criterion for selecting the individuals to remove were characterized as fitness based, diversity based, based on the size of the individuals and automatic. In both cases, the term automatic was used to describe all

Table 12 Count of the number of papers included in this study for each PPBIA presented

Algorithm	#Papers
DE	14
GP	11
EA	7
GA	7
PSO	6
ABC	2
UMDA	1
MBO	1
FWA	1
IA	1

Table 13 Count of the motivations that justify the use the DP for each category

Category	Balance between exploration and exploitation	Reducing the computational cost of the algorithm	Achieving a parameterless configuration	Adapting to a dynamic target function
Fitness based–fitness based	5	3	1	1
Fitness based–automatic	6	1	–	–
Automatic-fitness based	3	2	3	–
Automatic–automatic	6	1	1	–
Fitness based–based on the size of the individual	–	3	–	1
Diversity based–automatic	4	–	–	1
Based on the number of fitness evaluations–fitness based	4	2	–	–
Diversity based–fitness based	1	1	–	–
Automatic–diversity based	–	1	–	–
Total	29	14	5	3

those approaches that depend on some criterion established *a priori*, and not on the particular events that happen dynamically in the population during the evolution. Every combination of the four categories from one axis with the four from the other determines a category of articles. The articles thus categorized were subsequently described, showcasing the numerous variations and methodologies introduced to date for the management of dynamic populations. An intriguing aspect pertains to the justifications the various selected past contributions provide for employing dynamic populations. We observe that these motivations, identified in Table 13, remain essential and current priorities for bioinspired algorithms today. In particular, the tremendous proliferation of data we witness daily, especially in certain application domains, intensifies these needs.

Our conclusion is that the use of dynamic populations in bioinspired algorithms is still a highly relevant and crucial research direction, requiring substantial effort, advancements, and progress. In a world where the training data used by modern machine learning algorithms is increasingly dynamic (consider, for example, the use of data batches), the hardware architectures used for learning also tend to be distributed and dynamic, the learning algorithms themselves become every time more dynamic, adapting their behavior over time and favoring the hybridization of classical techniques, having dynamic populations available, and consequently, efficient and effective algorithms to manage them, is a natural consequence. For this reason, we identify research in this field as an interesting hot topic in the study area of bioinspired algorithms.

5.1 Potential areas for further study

From Fig. 1, we can notice that, in addition to the 9 categories described in Sect. 3, there are 7 other categories that have not been explored in the literature. These unexplored categories indicate direction for possible future studies. Among them is " Diversity Based - Diversity Based", which likely remains unexplored because using only diversity as criterion to regulate the population size can lead to a loss of performance, which is the ultimate goal of any algorithm. Although Diversity, both phenotypic and genotypic, is very important for the evolution process, in fact lack of diversity can lead to early convergence. For this reason future research should investigate methods to balance diversity without compromising performance. Other two unexplored categories are " Based on the number of Fitness Evaluations - Automatic" and " Based on the number of Fitness Evaluations - Diversity Based". As discussed in Sect. 3.7, the primary focus when using the number of fitness evaluations as a criterion for deciding when and how many individuals to remove is mainly to limit the resources used by the algorithm. The two aforementioned categories, which would use diversity and and automatic approach as criteria for selecting which individuals to remove from the population are likely to result in the discard of well-performing solutions in an attempt at reducing the computational cost of the algorithm. Furthermore we can identify three categories, " Diversity Based - Based on the size of the individual", " Based on the number of Fitness Evaluations - Based on the size of the individual" and " Automatic - Based on the size of the individual". These categories would use as criterion for selecting which individual to remove from the population the size of those. These categories likely have not been studied since not all PBBIAs have a varying individual size, limiting the potential algorithms to only a few, i.e. Genetic Programming(GP). Being the size of the individual a driver of the computational cost, these three categories present potential when the goal is to reduce the computational cost of the algorithm. The final unexplored category is " Fitness Based - Diversity Based", which is the only unexplored category among those that use fitness as a criterion to determine when and how many individuals to remove from the population. This category is worth exploring, as different studies [5, 6] have shown how decreasing the population size can lead to a loss in diversity. This can be important to avoid a premature convergence. Using diversity as criterion to select which individuals to remove from the population, specifically targeting those that are very similar to others, could help reducing the loss of diversity.

Funding Open access funding provided by FCT|FCCN (b-on). This work was supported by national funds through FCT (Fundação para a Ciência e a Tecnologia), under the project - UIDB/04152/2020 - Centro de Investigação em Gestão de Informação (MagIC)/NOVA IMS (https://doi.org/10.54499/UIDB/04152/2020).

Data availability All the data used for this work can be accessed through Scopus and Web of Science.

Declarations

Confict of interest The authors declare through the submission of this document that they have no Conflict of interest.

References

1. X.-S. Yang, X. He, In: Yang, X.-S. (ed.) Swarm Intelligence and Evolutionary Computation: Overview and Analysis, 1–23. Springer, Cham (2015)
2. A.P. Piotrowski, Review of differential evolution population size. Swarm Evol. Comput. **32**, 1–24 (2017). https://doi.org/10.1016/j.swevo.2016.05.003
3. A.P. Piotrowski, J.J. Napiorkowski, A.E. Piotrowska, Population size in particle swarm optimization. Swarm Evol. Comput. **58**, 100718 (2020). https://doi.org/10.1016/j.swevo.2020.100718
4. C.M. Fernandes, N. Fachada, J.L.J. Laredo, J.J. Merelo, A.C. Rosa, Population sizing of cellular evolutionary algorithms. Swarm Evol. Comput. **58**, 100721 (2020). https://doi.org/10.1016/j.swevo.2020.100721
5. F. Fernandez, L. Vanneschi, M. Tomassini, The effect of plagues in genetic programming: a study of variable-size populations, in *Genetic Programming*. ed. by C. Ryan, T. Soule, M. Keijzer, E. Tsang, R. Poli, E. Costa (Springer, Berlin, Heidelberg, 2003), pp.317–326

6. F. Fernandez, M. Tomassini, L. Vanneschi, Saving computational effort in genetic programming by means of plagues. In: The 2003 Congress on Evolutionary Computation, 2003. CEC '03., **3**, 2042–20493 (2003). https://doi.org/10.1109/CEC.2003.1299924
7. D. Rochat, M. Tomassini, L. Vanneschi, Dynamic size populations in distributed genetic programming, in *Genetic Programming*. ed. by M. Keijzer, A. Tettamanzi, P. Collet, J. Hemert, M. Tomassini (Springer, Berlin, Heidelberg, 2005), pp.50–61
8. P. Kouchakpour, A. Zaknich, T. Braunl, Dynamic population variation in genetic programming. Inf. Sci. **179**, 1078–1091 (2009). https://doi.org/10.1016/j.ins.2008.12.009
9. Y.-Y. Tao, J. Cao, M.-L. Li, Genetic programming using dynamic population variation for computational efforts reduction in system modeling. J. Shanghai Jiaotong Univ. (Sci.) (2012). https://doi.org/10.1007/s12204-012-1251-7
10. D. Farinati, I. Bakurov, L. Vanneschi, A study of dynamic populations in geometric semantic genetic programming. Inform. Sci. (2023). https://doi.org/10.1016/j.ins.2023.119513
11. J. Liang, Z. Chen, Y. Wang, X. Ban, K. Qiao, K. Yu, A dual-population constrained multi-objective evolutionary algorithm with variable auxiliary population size. Complex Intell. Syst. **9**(5), 5907–5922 (2023). https://doi.org/10.1007/s40747-023-01042-2
12. W. Bingchuan, Z.-Y. Shui, Y. Feng, Z. Ma, Evolutionary algorithm with dynamic population size for constrained multiobjective optimization. Swarm Evol. Comput. **73**, 101104 (2022). https://doi.org/10.1016/j.swevo.2022.101104
13. O. Montiel Ross, O. Castillo, P. Melin, R. Sepúlveda, Intelligent control of dynamic population size for evolutionary algorithms., 551–557 (2006)
14. L. Cui, G. Li, Z. Zhu, Q. Lin, Z. Wen, N. Lu, K.-C. Wong, J. Chen, A novel artificial bee colony algorithm with an adaptive population size for numerical function optimization. Inform. Sci. (2017). https://doi.org/10.1016/j.ins.2017.05.044
15. K.C. Tan, T.H. Lee, E.F. Khor, Evolutionary algorithms with dynamic population size and local exploration for multiobjective optimization. IEEE Transact. Evolut. Comput. **5**, 565–588 (2002). https://doi.org/10.1109/4235.974840
16. N. Awad, M. Ali, P. Suganthan, Ensemble of parameters in a sinusoidal differential evolution with niching-based population reduction. Swarm Evolut. Comput. (2017). https://doi.org/10.1016/j.swevo.2017.09.009
17. A.E. Eiben, E. Marchiori, V.A. Valkó, Evolutionary algorithms with on-the-fly population size adjustment, in *Parallel Problem Solving from Nature - PPSN VIII*. ed. by X. Yao, E.K. Burke, J.A. Lozano, J. Smith, J.J. Merelo-Guervós, J.A. Bullinaria, J.E. Rowe, P. Tiňo, A. Kabán, H.-P. Schwefel (Springer, Berlin, Heidelberg, 2004), pp.41–50
18. X. Shu, Y. Liu, J. Liu, M. Yang, Q. Zhang, Multi-objective particle swarm optimization with dynamic population size. J. Comput. Design Eng. **10**(1), 446–467 (2022). https://doi.org/10.1093/jcde/qwac139
19. L. Vanneschi, G. Cuccu, Reconstructing Dynamic Target Functions by Means of Genetic Programming Using Variable Population Size **343**, 121–134 (2011). https://doi.org/10.1007/978-3-642-20206-3_8
20. C. Fernandes, V. Ramos, A. Rosa, Varying the population size of artificial foraging swarms on time varying landscapes, 311–316 (2005). https://doi.org/10.1007/11550822_49
21. J. Branke, Evolutionary Optimization in Dynamic Environments. Springer, ??? (2002)
22. J.J. Grefenstette, Genetic algorithms for changing environments. In: Proceedings of Parallel Problem Solving from Nature, 137–144 (1992)
23. A.E. Eiben, R. Hinterding, Z. Michalewicz, Parameter control in evolutionary algorithms. IEEE Trans. Evol. Comput. **3**(2), 124–141 (1999)
24. Y. Jin, J. Branke, Evolutionary optimization in uncertain environments-a survey. IEEE Trans. Evol. Comput. **9**(3), 303–317 (2005)
25. T. Bäck, A.E. Eiben, N.A.L. Vaart, An emperical study on gas without parameters, in *Parallel Problem Solving from Nature PPSN VI*. ed. by M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J.J. Merelo, H.-P. Schwefel (Springer, Berlin, Heidelberg, 2000), pp.315–324
26. A. Zamuda, J. Brest, E. Mezura-Montes, Structured population size reduction differential evolution with multiple mutation strategies on cec 2013 real parameter optimization. In: 2013 IEEE Congress on Evolutionary Computation, 1925–1931 (2013). https://doi.org/10.1109/CEC.2013.6557794
27. A. George, B.R. Rajakumar, Apoga: an adaptive population pool size based genetic algorithm. AASRI Procedia **4**, 288–296 (2013). https://doi.org/10.1016/j.aasri.2013.10.043

28. N.S. Teng, J. Teo, M.H.A. Hijazi, Self-adaptive population sizing for a tune-free differential evolution. Soft Comput. **13**(7), 709–724 (2009). https://doi.org/10.1007/s00500-008-0344-6
29. I. Wong, W. Liu, C.-M. Ho, X. Ding, Continuous adaptive population reduction (capr) for differential evolution optimization. SLAS Technol. **22**, 2472630317690318 (2017). https://doi.org/10.1177/2472630317690318
30. G. Karafotias, M. Hoogendoorn, A.E. Eiben, Parameter control in evolutionary algorithms: trends and challenges. IEEE Trans. Evol. Comput. **19**(2), 167–187 (2015)
31. J.E. Smith, Self-adaptation in evolutionary algorithms for combinatorial optimization. Eur. J. Oper. Res. **185**(3), 1396–1414 (2008)
32. P.J. Angeline, Adaptive and self-adaptive evolutionary computations. In: Computational Intelligence: A Dynamic Systems Perspective, 152–163. IEEE Press, ??? (1995)
33. Á. Fialho, M. Schoenauer, M. Sebag, Toward comparison-based adaptive operator selection. In: Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, 767–774 (2010)
34. S. Yang, X. Yao, Experimental study on population-based incremental learning algorithms for dynamic optimization problems. Soft. Comput. **9**(11), 815–834 (2005)
35. E.C. Shi, F.H.F. Leung, B.N.F. Law, Differential evolution with adaptive population size. In: 2014 19th International Conference on Digital Signal Processing, 876–881 (2014). https://doi.org/10.1109/ICDSP.2014.6900794
36. G.G. Yen, H. Lu, Dynamic population strategy assisted particle swarm optimization. In: Proceedings of the 2003 IEEE International Symposium on Intelligent Control, 697–702 (2003). https://doi.org/10.1109/ISIC.2003.1254720
37. W.-F. Leong, G.G. Yen, Pso-based multiobjective optimization with dynamic population size and adaptive local archives. IEEE Transact. Syst. Man Cybernet Part B (Cybernet.) **38**(5), 1270–1293 (2008). https://doi.org/10.1109/TSMCB.2008.925757
38. X. Zhang, Z.-H. Zhan, J. Zhang, Adaptive population differential evolution with dual control strategy for large-scale global optimization problems. In: 2020 IEEE Congress on Evolutionary Computation (CEC), 1–7 (2020). https://doi.org/10.1109/CEC48606.2020.9185854
39. W. Zhu, Y. Tang, J.-A. Fang, W. Zhang, Adaptive population tuning scheme for differential evolution. Inf. Sci. **223**, 164–191 (2013). https://doi.org/10.1016/j.ins.2012.09.019
40. J. Brest, M.S. Maučec, B. Bošković, Self-adaptive differential evolution algorithm with population size reduction for single objective bound-constrained optimization: algorithm j21. In: 2021 IEEE Congress on Evolutionary Computation (CEC), 817–824 (2021). https://doi.org/10.1109/CEC45853.2021.9504782
41. L. Cui, G. Li, Z. Zhu, Q. Lin, Z. Wen, N. Lu, K.-C. Wong, J. Chen, A novel artificial bee colony algorithm with an adaptive population size for numerical function optimization. Inform. Sci. (2017). https://doi.org/10.1016/j.ins.2017.05.044
42. I. Gonçalves, S. Silva, C.M. Fonseca, M. Castelli, Unsure when to stop? In: Proceedings of the Genetic and Evolutionary Computation Conference. ACM, ??? (2017). https://doi.org/10.1145/3071178.3071328
43. R. Poláková, P. Bujok, Adaptation of population size in differential evolution algorithm: an experimental comparison. In: 2018 25th International Conference on Systems, Signals and Image Processing (IWSSIP), 1–5 (2018). https://doi.org/10.1109/IWSSIP.2018.8439374
44. E.A. Merchán-Cruz, G. Urriolagoitia-Sosa, J. Ramírez-Gordillo, R. Rodríguez-Cañizo, I.Y. Campos-Padilla, J.J. Muñoz-César, E. Lugo-González, Ga based trajectory planner for robot manipulators sharing a common workspace with adaptive population size. In: 2008 Electronics, Robotics and Automotive Mechanics Conference (CERMA '08), 520–525 (2008). https://doi.org/10.1109/CERMA.2008.65
45. T.-Y. Huang, Y.-Y. Chen, Parental population sizing in evolutionary strategies. In: Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546), **2**, 1351–13582 (2001). https://doi.org/10.1109/CEC.2001.934348
46. J. Yu, H. Takagi, Accelerating fireworks algorithm with dynamic population size strategy. In: 2020 Joint 11th International Conference on Soft Computing and Intelligent Systems and 21st International Symposium on Advanced Intelligent Systems (SCIS-ISIS), 1–6 (2020). https://doi.org/10.1109/SCISISIS50064.2020.9322693
47. Z.-H. Zhan, J. Zhang, Y. Li, H.S.-H. Chung, Adaptive particle swarm optimization. IEEE Transact. Syst. Man Cybernet. Part B (Cybernet.) **39**(6), 1362–1381 (2009). https://doi.org/10.1109/TSMCB.2009.2015956

48. L. Zheng, S. Luo, Adaptive differential evolution algorithm based on fitness landscape characteristic. Mathematics **10**, 1511 (2022). https://doi.org/10.3390/math10091511
49. J.C. Costa, R. Tavares, A. Rosa, An experimental study on dynamic random variation of population size. In: IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No.99CH37028), **1**, 607–6121 (1999). https://doi.org/10.1109/ICSMC.1999.814161
50. W.-F. Leong, G.G. Yen, Dynamic population size in pso-based multiobjective optimization. In: 2006 IEEE International Conference on Evolutionary Computation, 1718–1725 (2006). https://doi.org/10.1109/CEC.2006.1688515
51. L.S. Lima, H.S. Bernardino, H.J.C. Barbosa, Designing combinational circuits using a multi-objective cartesian genetic programming with adaptive population size. In: Machine Learning, Optimization, and Data Science: 5th International Conference, LOD 2019, Siena, Italy, September 10-13, 2019, Proceedings, 592–604. Springer, Berlin, Heidelberg (2019). https://doi.org/10.1007/978-3-030-37599-7_49
52. J. Arabas, Z. Michalewicz, J. Mulawka, Gavaps-a genetic algorithm with varying population size. In: Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence, 73–781 (1994). https://doi.org/10.1109/ICEC.1994.350039
53. S. Luke, G.C. Balan, L. Panait, Population implosion in genetic programming, in *Genetic and Evolutionary Computation—GECCO 2003*. ed. by E. Cantú-Paz, J.A. Foster, K. Deb, L.D. Davis, R. Roy, U.-M. O'Reilly, H.-G. Beyer, R. Standish, G. Kendall, S. Wilson, M. Harman, J. Wegener, D. Dasgupta, M.A. Potter, A.C. Schultz, K.A. Dowsland, N. Jonoska, J. Miller (Springer, Berlin, Heidelberg, 2003), pp.1729–1739
54. V. Cutello, D. Lee, S. Leone, G. Nicosia, M. Pavone, Clonal selection algorithm with dynamic population size for bimodal search spaces **4221**, 949–958 (2006). https://doi.org/10.1007/11881070_125
55. Z. Ma, A. Krings, Dynamic populations in genetic algorithms **12**, 1807–1811 (2008). https://doi.org/10.1145/1363686.1364119
56. H. Hu, Z. Cai, S. Hu, Y. Cai, J. Chen, S. Huang, Improving monarch butterfly optimization algorithm with self-adaptive population. Algorithms **11**, 71 (2018). https://doi.org/10.3390/a11050071
57. E. Li, An adaptive surrogate assisted differential evolutionary algorithm for high dimensional constrained problems. Appl. Soft Comput. (2019). https://doi.org/10.1016/j.asoc.2019.105752
58. D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization. IEEE Trans. Evol. Comput. **1**(1), 67–82 (1997). https://doi.org/10.1109/4235.585893
59. Y. Hong, Q. Ren, J. Zeng, Adaptive population size for univariate marginal distribution algorithm. In: 2005 IEEE Congress on Evolutionary Computation, **2**, 1396–14022 (2005). https://doi.org/10.1109/CEC.2005.1554853
60. T. Hu, Y.P. Chen, W. Banzhaf, Wimax network planning using adaptive-population-size genetic algorithm. EvoCOMNET'10, 31–40. Springer, Berlin, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12242-2_4
61. T.J. Choi, C.W. Ahn, An adaptive population resizing scheme for differential evolution in numerical optimization. J. Comput. Theor. Nanosci. (2015). https://doi.org/10.1166/jctn.2015.3895
62. S. Silva, P.J.N. Silva, E. Costa, Resource-limited genetic programming: replacing tree depth limits, in *Adaptive and Natural Computing Algorithms*. ed. by B. Ribeiro, R.F. Albrecht, A. Dobnikar, D.W. Pearson, N.C. Steele (Springer, Vienna, 2005), pp.243–246
63. S. Silva, E. Costa, Resource-limited genetic programming: the dynamic approach. In: Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation. GECCO '05, 1673–1680. Association for Computing Machinery, New York, NY, USA (2005). https://doi.org/10.1145/1068009.1068290
64. R. Tanabe, A.S. Fukunaga, Improving the search performance of shade using linear population size reduction. In: 2014 IEEE Congress on Evolutionary Computation (CEC), 1658–1665 (2014). https://doi.org/10.1109/CEC.2014.6900380
65. A. Piotrowski, L-shade optimization algorithms with population-wide inertia. Information Sciences **468** (2018) https://doi.org/10.1016/j.ins.2018.08.030
66. R. Tanabe, A. Fukunaga, Success-history based parameter adaptation for differential evolution. In: 2013 IEEE Congress on Evolutionary Computation, 71–78 (2013). https://doi.org/10.1109/CEC.2013.6557555
67. D. Chen, C. Zhao, Particle swarm optimization with adaptive population size and its application. Appl. Soft Comput. **9**, 39–48 (2009). https://doi.org/10.1016/j.asoc.2008.03.001

68. R. Polakova, J. Tvrdik, P. Bujok, Differential evolution with adaptive mechanism of population size according to current population diversity. Swarm Evolut. Comput. (2019). https://doi.org/10.1016/j.swevo.2019.03.014

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Davide Farinati[1] · Leonardo Vanneschi[1]

✉ Davide Farinati
dfarinati@novaims.unl.pt

Leonardo Vanneschi
lvanneschi@novaims.unl.pt

1 NOVA Information Management School (NOVA IMS), Universidade Nova de Lisboa, Campus de Campolide, 1070-312 Lisboa, Portugal