APPLICATION OF BIODIVERSITY STATISTICAL INDEXES TO CONTROL THE DIVERSITY OF SOLUTIONS IN MULTIOBJECTIVE GENETIC ALGORITHMS

APLICAÇÃO DE ÍNDICES ESTATÍSTICOS DE BIODIVERSIDADE NO CONTROLE DE DIVERSIDADE DAS SOLUÇÕES EM ALGORITMOS GENÉTICOS MULTIOBJETIVO

Jean Carlo de Elias Moreira[1]
Adinovam Henriques de Macedo Pimenta[2]

ABSTRACT

In the last decades, different optimization methodologies through evolutionary algorithms have been proposed, explored and successfully applied to a wide range of problems. Each of these methodologies has distinct characteristics, typical advantages and characteristic disadvantages. However, there is a problem that is shared by almost all of them: controlling the diversity of solutions. While natural selection favors variations toward greater divergence, in artificial evolution candidate solutions are homogenized. The diversity of solutions directly affects algorithmic performance, especially in multi-objective problems. Many authors have argued that promoting diversity would be beneficial in evolutionary optimization processes and that this could help prevent premature convergence into suboptimal solutions. In this paper we analyze some statistical indexes of biodiversity widely used in ecology and their impact when inserted in evolutionary algorithms. We also suggest practical ways to measure and promote diversity in multiobjective genetic algorithms.

Key-words: Multiobjective Genetic Algorithm. Diversity. Pareto Frontier.

RESUMO

Nas últimas décadas, diferentes metodologias de otimização por meio de algoritmos evolutivos foram propostas, exploradas e aplicadas com sucesso a uma ampla gama de problemas. Cada uma destas metodologias possui características distintas, vantagens típicas e desvantagens características. No entanto, existe um problema que é compartilhado por quase todas elas: o controle da diversidade das soluções. Enquanto a seleção natural favorece variações em direção a uma maior divergência, na evolução artificial as soluções candidatas se homogeneízam. A diversidade das soluções afeta diretamente o desempenho algorítmico, principalmente em problemas multiobjetivos. Muitos autores argumentaram que a promoção da diversidade seria benéfica nos processos de otimização evolutiva e que isso poderia ajudar a evitar a convergência prematura em soluções abaixo do ideal. Neste trabalho analisamos alguns índices estatísticos de biodiversidade amplamente utilizados na ecologia e qual o seu impacto

[1] Aluno da graduação em Ciência da Computação da Faculdade de Tecnologia, Ciências e Educação – FATECE. Bolsista de Iniciação Científica da Instituição. jeancarlo@jeancarloem.com
[2] Professor Doutor em Ciência da Computação. Orientador do Projeto de Iniciação Científica. adinovam@gmail.com

quando inseridos nos algoritmos evolutivos. Sugerimos, também, formas práticas de medir e promover a diversidade em algoritmos genéticos multiobjetivo.

Palavras-chaves: Algoritmo Genético Multiobjetivo. Diversidade. Fronteira de Pareto.

Introduction

Multiobjective Optimization Problems (MOP) generally involve the simultaneous optimization of conflicting objectives. This interaction results in a set of compromised solutions, known as the ideal set of Pareto (STIGLITZ *et al.*, 2019). As none of the solutions in this set can be considered better than the others in relation to all objectives, the goal of multiobjective optimization is to find the greatest possible number of ideal Pareto solutions (or pareto-optimal, as they are also known) (LI *et al*., 2019).

According to Alexandropoulos *et al.* (2019), Multiobjective Evolutionary Algorithms (MEA) are among the most powerful techniques to meet this requirement due to their inherent parallelism and their ability to explore the similarities between the solutions. The search for widely diversified pareto-optimal solutions is not, however, an easy task (LIU *et al.*, 2016). The way in which solutions are selected can cause premature convergence of the population in restricted portions of the ideal Pareto set or even in solutions that do not belong to this set. Therefore, maintaining genetic diversity within the population is mandatory in order to find true and diverse pareto-optimal solutions (TOFFOLO *et al.*, 2003).

After the publication of Goldberg's work (1989), which suggested the use of non-dominance of solutions as a criterion to be observed during the optimization process, it generated an immense interest in MEA. The initial MEAs - *Multiobjective Genetic Algorithm* (MOGA) (FONSECA; FLEMING, 1993b), *Non-dominated Sorting Genetic Algorithm* (NSGA) (SRINIVAS; DEB, 1994a) and *Niched Pareto Netic Algormm* (HORN *et al.*, 1994) - explored this suggestion intensely.

These three algorithms showed that different ways of implementing the concept of nondominance can result in successful MEA. However, these algorithms could not guarantee the convergence to the Pareto ideal set, since an operator was missing to preserve the best solutions (elitism). Thus, the most recent MEAs focused mainly on how elitism could be introduced into a MEA. This resulted in three of the most successful MEAs in the literature: *Strength Pareto Evolutionary Algorithm* (SPEA) (ZITZLER *et al.*, 2001), *Pareto Archived Evolution Strategy* (PAES) (KNOWLES *et al.*, 1999) and *Non -dominated Sorting Genetic Algorithm II* (NSGA-II) cite deb2000fast, the latter

being the most popular. With the development of better algorithms, MEA has also been used in several practical applications (LAUMANNS *et al.*, 2001).

There is, however, a gap in the literature pointing to the absence of studies and analyzes related to the guarantee of diversified pareto-optimal solutions, as well as a way to monitor the diversification of solutions throughout the genetic process of MEA (BHOSKAR *et al.*, 2015; GEN *et al.*, 2014; PAQUETE *et al.*, 2018). In this sense, Kumar et al. (2002) and Abraham et al. (2005) suggested a series of algorithms, which guarantee convergence for pareto-optimal solutions, but do not address the following two aspects:

1. Convergent algorithms do not guarantee the maintenance of the diversity of solutions;
2. The algorithms do not specify the time complexity for their convergence.

Although it is difficult to perform the second task, even for simple objective functions (see (LAUMANNS *et al.*, 2002)) and also for mono-objective problems, the first task is just as important as the task of converging to the ideal Pareto set. Deb et al. (2002b) suggested an MEA that tries to maintain diversity while converging on pareto-optimal solutions. However, there is no evidence for its convergence properties. Knowles (2002) presented an adaptive archiving strategy. This strategy provides solutions in some "critical"regions of the Pareto set, but convergence can only be guaranteed for solutions at the ends of the Pareto set, promoting the agglomeration of solutions in extreme regions of the objective functions and, consequently, decreasing the diversity of solutions.

In this way, the main objective of this work is to identify a statistical index that measures the biodiversity of an ecosystem and to propose a way to integrate this index with the MEA, providing the MEA with a way to monitor the diversity of the solutions that are being generated during the evolutionary process of the population.

1 Evolutionary Computing

Evolutionary Computing (EC) is a branch of research that comprises a set of search and optimization techniques based on the principles of biological evolution, such as natural selection and genetic inheritance, to solve different types of problems. These techniques are being increasingly used to solve a wide variety of problems, ranging from practical applications in industry and commerce to scientific research (EIBEN *et al.*, 2007).

Through EC techniques, a population of individuals is created that reproduce and compete with each other for survival. The best individuals are more likely to survive and transfer their characteristics to new generations.

The main EC techniques are Evolutionary Strategies (ES), Evolutionary Programming (EP), Genetic Programming (GP) and Genetic Algorithms (GA) (PEDRYCZ; GOMIDE, 1998; BANZHAF *et al.*, 1998). All of these techniques use the same basic principle as the EC, but operate in different ways.

ES codes encode a population into a single vector of individuals with real values and their main idea is to combine the individuals, through crossing, to generate a descendant that will replace the worst individual in the population. The first algorithm using an evolutionary strategy was developed in 1964 at the Technical University of Berlin (KUSIAK, 2000).

EP uses prediction of the behavior of finite state machines and their optimization (PEDRYCZ; GOMIDE, 1998). In this way, each individual represents a finite state machine. The selection of individuals is based on elitism.

GP is a technique for automatic generation of computer programs proposed by Koza (1992), inspired by the GA theory proposed by Holland (1975). It is possible to create and manipulate software genetically using GP and applying concepts inherited from biology to generate computer programs automatically.

GAs are algorithms that model a solution to a specific problem in a data structure and apply operators that recombine these solutions while preserving critical information. The GA were conceived by Holland (1975), with the initial objective of studying the phenomena related to the adaptation of species and natural selection that occur in nature, as well as to develop a way of incorporating these concepts to computers (MITCHELL, 1998). GA and GP are the two main research fronts in evolutionary computing.

The next section presents the main concepts of GA, EP, GP and ES will not be studied in this proposal. More detailed information about these techniques can be found in several works (BäCK; SCHWEFEL, 1993; De Jong, 2006; EIBEN *et al.*, 2007).

2 Multiobjective Optimization

In a simple optimization problem (mono-objective) the search space is generally well defined. But when the problem has multiple objectives, contradicting each other, there is not only a single optimal solution, but a set of possible solutions.

Even though some real-world problems can be reduced to a single objective, it is often difficult to summarize all the characteristics of the problem in a single objective and solve it using mono-objective optimization techniques, such as GA (ABRAHAM; JAIN; GOLDBERG, 2005). Thus, setting multiple goals is often the best way to solve a multi-objective problem.

Multiobjective optimization seeks to find solutions to problems that have conflicting objectives, that is, if it is possible to improve the outcome of one of the objectives, the other objective of the problem will be impaired. A function f_1 is said to conflict with a function f_2 when it is not possible, at the same time, to improve the value of both.

A didactic example that illustrates this type of problem is that of a car trip in which you want to minimize travel time and fuel consumption (cost). It is known that the more a vehicle is accelerated to reduce travel time, the more fuel is used. Mathematically, the multi-objective optimization problem can be formulated as presented by (DIAS; VASCONCELOS, 2002), by the Equation 1, as fallows.

$$\left.\begin{array}{lll} maximize/minimize & f_m(\mathbf{x}), & m = 1,2,\ldots,M \\ subject\ to & g_j(\mathbf{x}) \geq 0, & j = 1,2,\ldots,J \\ & h_k(\mathbf{x}) = 0, & k = 1,2,\ldots,K \\ & \mathbf{x}_i^{(L)} \leq \mathbf{x}_i \leq \mathbf{x}_i^{(U)}, & i = 1,2,\ldots,n \end{array}\right\} \quad (1)$$

where x is the vector of n decision variables $x = (x_1, x_2, \ldots, x_n)^T$. The values $x_i^{(L)}$ and $x_i^{(U)}$, represent the minimum and maximum values respectively for the variable x_i. These limits define the *decision variable space* or *decision space D*. In addition, the x vector will also be referred to as *solution*.

The J inequalities (g_j) and the K equalities (h_k) are called constraint functions. A feasible x solution will be one that satisfies the $J + K$ constraint functions and the $2n$ limits. Otherwise, the solution will not be feasible. The set of all feasible solutions form the *doable region* or *search space S*.

Each of the M objective functions $f_1(x)$, $f_2(x)$, ..., $f_M(x)$ can be maximized or minimized. The vector objective functions f(x) forms a multi-dimensional space called objective space *Z*. So, for each x solution in the decision space, there is a f(x) in the objectives space. This is a fundamental difference in relation to the optimization of simple objectives, whose objective space is one-dimensional.

The mapping then takes place between a x (*n*-dimensional) vector and a f(x) (*M*-dimensional) vector. For example, if each element of x and f(x) is a real number, then f(x) is defined as f(x): $R^n \rightarrow R^M$.

2.1 Pareto-optimal solutions

Making decisions implies a process that consists of several factors and whose objective is to find the best solution. In some cases, it is possible to find several good solutions, none of which is quantitatively better than the other. For example, when buying a car, one can assume that one is looking for a car with the lowest price and the greatest comfort. Figure 1 illustrates several alternatives to choose from.

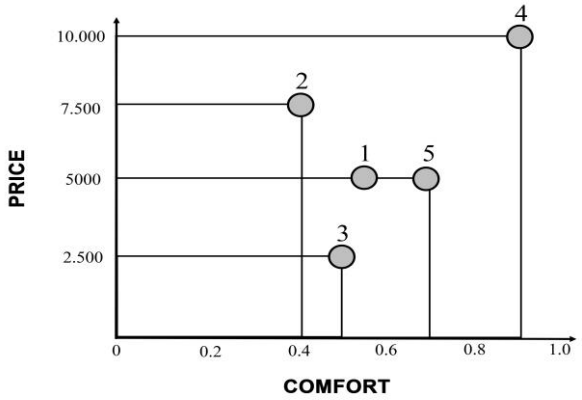


Figura 1 – Car purchase example, considering cost and comfort.

The goal is to minimize cost and maximize comfort. In this case, there are five possible purchase options. Intuitively, solution 1 is discarded, as solution 5 provides more comfort for the same price. Solution 2 is also discarded, as solutions 3 and 5 provide more comfort than solution 2 and at a lower price. There are then three solutions (3, 4 and 5) that are good alternatives for purchase. In quantitative terms, none is better than the other, as one is more comfortable, but less expensive, and vice versa. There is then a balance between the objectives. The greater the comfort, the higher the price of the car and the cheaper, the lower the comfort of the car.

It is said that a solution *dominates* another if its values are better in all objectives, or are better in one objective and tie in others. For example, solution 1 dominates solution 2, but is dominated by solutions 4 and 5. No solution dominates solutions 3, 4, and 5. Solutions 3, 4, and 5 can be said to be equally good. So there is a set of optimal solutions.

This set is called *set of non-dominated solutions*. The other solutions (1 and 2) form the *set of dominated solutions*.

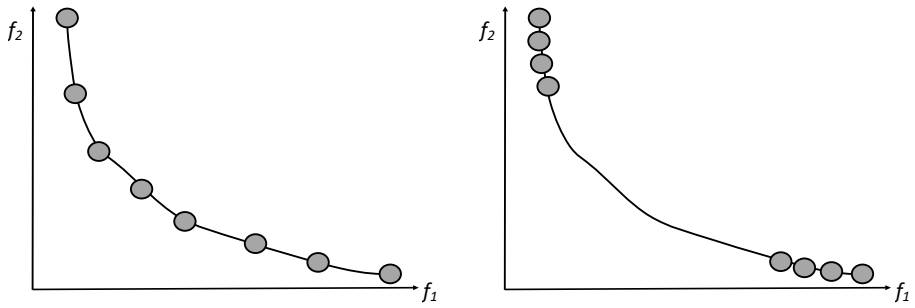
Considering that the non-dominated points are in a continuous space, a curve can be drawn connecting these points. All points contained in the curve form the *front of Pareto* or *frontier of Pareto*.

2.1 Goals in Multiobjective Optimization Problems

In a multiobjective optimization problem, all Pareto-optimal solutions, that is, the solutions found at the Pareto frontier, are equally important. Deb et al. (2002a) highlights two important goals in multiobjective optimization:

- Find a set of solutions that are as close as possible to the Pareto frontier;
- Find a set of non-dominated solutions with the greatest possible diversity.

The first goal is common for any optimization process. Solutions very far from the Pareto Frontier are not desirable, as this indicates that these solutions are far from the best solutions. However, finding the greatest diversity within non-dominated solutions is a specific goal for multiobjective optimization. Figure 2a presents a Pareto frontier with a good distribution of solutions when compared to that shown in Figure 2b.



(a) Balanced distribution (b) Unbalanced distribution

Figura 2 – Distribution of solutions along the Pareto frontier

In Figure 2b the solutions are distributed only in some regions, with each region privileging one of the objectives, which characterizes these solutions as *unbalanced*. It is necessary to ensure the greatest possible coverage of the frontier, making it possible to obtain a set of solutions with balanced objectives.

2.3 Differences between Single-Objective and Multi-Objective Optimization

Deb *et al.* (2002a) identifies three important differences between multi-objective optimization and single-objective optimization:

- In single-objective optimization problems, the goal is to find an optimal global solution (maximum or minimum). In multiobjective problems, there can be more than one global optimum. Finding the set of solutions on the Pareto frontier is as important as preserving the diversity in this set. An efficient algorithm for multiobjective optimization must consider both aspects;
- In multiobjective optimization problems, two spaces are considered (that of variables and that of objectives) instead of one. Single-objective optimization problems work only in the variable space, as they seek only one solution in the objective space. Maintaining diversity in both spaces increases the complexity of the problem, since the proximity between two solutions in the variable space does not imply proximity in the objective space;
- Traditional methods of multiobjective optimization are based on a simple function that weighs each objective. They can also treat each objective separately, using the other objectives as restrictions. Therefore, a multi-objective optimization problem can be converted into a simple optimization problem.

2.4 Pareto Dominance Operator

Multiobjective problems, unlike single-objective problems, have more than one solution. These solutions are known as *non-dominated solutions* or *efficient solutions*.

If there are M objective functions f_m, $m = 1, ..., M$ the operator $\lhd$ between two solutions, $x \lhd y$, means that the solution x is better than the y solution on at least one particular goal. Conversely, $x \rhd y$ denotes that the x solution is worse than the y solution for some purpose.

One $x^{(1)}$ solution is said to dominate another $x^{(2)}$ solution (represented mathematically as $x^{(1)} \prec x^{(2)}$ if the following conditions are met:

- The solution $x^{(1)}$ is no worse than $x^{(2)}$ in all objectives, that is, $f_m\left(x^{(1)}\right) \not\rhd f_m\left(x^{(2)}\right)$ for all $m = 1,2, ...,M$;

- The solution $x^{(1)}$ is strictly better than $x^{(2)}$ at least in one objective, that $f_m(x^{(1)}) \lhd f_m(x^{(2)})$ at least in one $m = 1,2, ...,M$.

If both conditions are met, it can be said that $x^{(1)}$ dominates $x^{(2)}$:

- $x^{(1)}$ dominates $x^{(2)}$;
- $x^{(1)}$ is not dominated by $x^{(2)}$;
- $x^{(1)}$ is not inferior than $x^{(2)}$.

If the solution $x^{(1)}$ is not dominated by any solution $x^{(i)}$ in all the feasible space, it is said that $x^{(1)}$ is an efficient non-dominated solution or Pareto-optimal solution.

In Figure 1, solution 5 dominates solution 1 ($5 \prec 1$), and solution 3 dominates solution 2 ($3 \prec 2$). Therefore, the concept of dominance allows to compare solutions with multiple objectives.

Thus, using these definitions, when a finite set of solutions is found, it becomes possible to make comparisons of the solutions two by two, dividing them into a group called *P* dominated solutions and *P'* non-dominated solutions. *P'* solutions are not dominated by any other *P* solution. If the *P'* non-dominated set covers the entire doable search space, it is called the global Pareto-optimal set. Figure 3 illustrates the spaces of the decision variables and objectives. The global Pareto-optimal frontier is also shown in this figure.

The Pareto-optimal frontier illustrated in Figure 3 is formed by values of objective functions $f(x) = (f_1(x), \dots, f_m(x))$ corresponding to each solution in the search space. Therefore, for each of the solutions found in the variable space, these solutions are represented in the objectives space, evaluating each one of them in each of the existing objectives.

3 Genetic Algorithms

GAs can be defined as iterative methods that contain a population with a fixed number of individuals, each of which is represented by a finite linear sequence of symbols, called chromosomes, that encode a possible solution to a given problem, or, depending on coding, part of the solution. These possible solutions are encoded in a space, called a search space, which can contain all possible solutions to the problem in question. An aptitude value is associated with each chromosome and this value represents how suitable the solution is the solution it encodes. In this way, individuals can be changed

until an individual is obtained who codifies a satisfactory solution to the problem addressed.

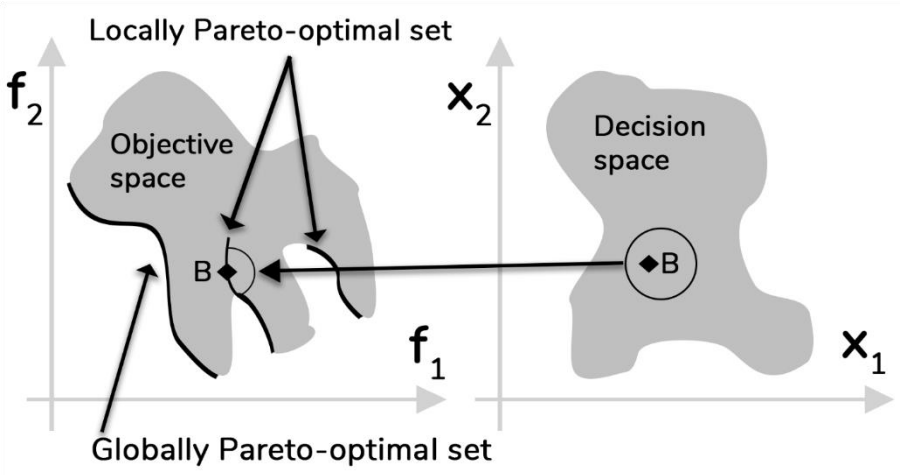


Figura 3 – Local and global Pareto-optimal solutions.

GAs are frequently used in search problems, having found wide application in several scientific areas, among which can be mentioned those related to solution optimization problems, machine learning, development of mathematical strategies and formulas, analysis of economic models, engineering problems, diverse applications in biology such as bacterial simulation, immune systems, ecosystems and discovery of topologies and properties of organic molecules (GOLDBERG, 1989; HAUPT; HAUPT, 2004; CHAMBERS, 2001; MICHALEWICZ, 1996; MITCHELL, 1998).

The functioning of an GA occurs from the initialization process, which creates a *population* of individuals representing initial solutions that are then submitted to the evolution process, which goes through the following steps: evaluation, selection, crossing and mutation. Each of these steps will be covered below.

3.1 Avaliation

It is at this stage that the first step for selection is taken. Each individual receives an evaluation according to their degree of aptitude, that is, it measures how good the individual is in solving the problem in question. Taking into account that this task is performed for each individual in each generation, it can be said that its computational cost is relatively high.

3.3 Selection

In the selection stage, individuals are selected for the next stage, the crossing. The degree of aptitude of each individual is used to make a draw in which the most apt individuals are more likely to be chosen.

The following are some selection methods.

Roulette: In this method, individuals from a population are selected for the next generation using roulette. Each individual in the population is represented in the roulette by a slice proportional to their fitness index. Thus, individuals with greater aptitude occupy larger slices of the roulette, while individuals with less aptitude occupy smaller slices. For the selection of individuals, the roulette wheel is rotated *N* times, where *N* is the initial population number. Each time the wheel stops spinning, the chromosome selected by the marker will be copied to the next generation. Chromosomes with more roulette space are more likely to be selected.

In Figure 4 a roulette is shown that reflects the fitness values of the 5 individuals (S1, S2, S3, S4 and S5) of a population, with individual S2 (with the largest slice) being selected by the marker.

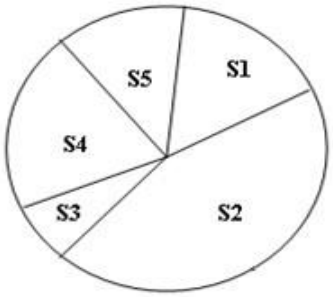


Figura 4 – Roulette selection method.

Tournament: When this method is used, a *n* number of individuals in the population is chosen at random and with the same probability. The chromosome with the highest fitness among these *n* chromosomes is selected for the intermediate population. This process will be repeated until the intermediate population is filled.

You can also add a technique called *elitism* to these methods. In this technique, all individuals are ordered according to their aptitude value and individuals who fall below the average value are discarded. This technique results in a population that will have individuals with an aptitude value above the average of the previous population. This technique also ensures that the best individual obtained in the current generation will

be present in the next generation that is being generated, preserving their genetic characteristics.

3.4 Crossover

Selected individuals, they pass with a pre-established probability for the crossing process. This probability is called the crossing rate. In this process, two individuals are selected for the crossing, generating two new individuals that will compose the next generation, thus replacing the two selected individuals.

Among the crossing methods, we can cite (GOLDBERG, 1989):

One Point Crossover In this method, a cutoff point p is chosen whichever $1 < p < w$, w being the length of the chromosome. The first child chromosome will receive the same genes as parent 1 between the 1 and p index. The following genes will be populated with the genes from parent 2 between the $p + 1$ and w index. The second child chromosome will be filled in the opposite way, that is, it will receive the same genes as parent 2 included between the 1 and p index. The following genes will be filled with the parent's genes between the $p + 1$ and w index. Figure 5 illustrates an example of this type of crossover.

Two Points Crossover At the crossing of 2 points, 2 points, p_1 and p_2, are randomly chosen from the parent chromosomes such that $p_1 < p_2$. The first child chromosome will have the same genes as the parent 1 chromosome. However, the region comprising p_1 and p_2 will have the parent 2 genes. The second child chromosome will have the same genes as the parent 2 chromosome. the region comprising p_1 and p_2 will have the genes of parent 1. Figure 6 illustrates an example of this type of crossover.

Figura 5 – One Point Crossover.

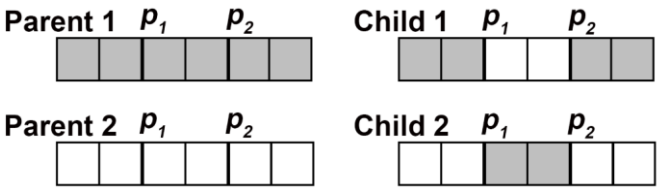


Figura 6 – Two Points Crossover.

3.6 Arithmetic Crossing

Arithmetic crossover is defined as a linear combination of two vectors. If x_1 and x_2 are crossed, the offspring will be $x'_1 = a \cdot x'_1 + (1 - a) \cdot x'_2$ e $x'_2 = a \cdot x'_2 + (1 - a) \cdot x'_1$, where a is a random number in the range [0,1].

3.7 Mutation

Mutation is a process that guarantees the exploration of several alternatives and is applied randomly to individuals of the current generation with a predefined probability rate called *mutation rate*. The mutation is nothing more than some kind of change that must be made on a gene. This change depends on how the gene was designed. In a chromosome with binary encoding, for example, the mutation changes a gene with a value of 1 to a value of 0 and vice versa. In a chromosome with real or full encoding, the gene is changed by changing the current value of the gene for some other value among the possible valid values for the gene in question.

Although randomness is an intrinsic characteristic of GA, they do a targeted search. The new individuals generated have characteristics of the previous population, and these characteristics are used as a history that directs the search for a better result than the result generated by the previous population.

In Algorithm 0.1 the basic steps of an GA are presented in which the stopping criterion is given by the parameter *maxGenerations*.

4 Multiobjective Genetic Algorithms

To solve multiobjective optimization problems, GA can be used, transforming the multiobjective optimization problem into a mono-objective optimization problem, using some preference criteria to find a single solution. This solution must have acceptable values in all objectives according to the established preference criteria, such as, for example, prioritizing one or more objectives giving weights to them. If more than one solution is required, it is necessary to run the algorithm repeatedly by changing the preference criteria.

On the other hand, it is also possible to obtain multiple solutions in an GA execution to solve multiobjective problems, trying to find the solutions that are on the Pareto frontier considering the dominance of these solutions.

Algoritmo 0.1: Genetic Algorithm

```
1  Begin
2      t := 1;
3      initializes(P_t); // P_t is population of the generation t
4      While t <= maxGeracoes Do
5          evaluate(P_t);
6          t := t + 1 ;
7          select(t, P_{t-1});
8          applyCrossover(P_t);
9          applyMutation(P_t);
10     End
11 End
```

GAs that consider the dominance of solutions using a set of objective functions are called Multiobjective Genetic Algorithms (MOGA).

Goldberg (1989) states in his book that the use of GAs for multiobjective problem solving started when Schaffer (1985) implemented the first version of an MOGA called VEGA (*Vector Evaluated Genetic Algorithm*). In the proposed algorithm, considering a population of *N* individuals and *K* objectives, this population is divided into *K* sub-populations with *N/K* individuals in each one. The GAs selection operator is applied separately for each of the subpopulations, that is, for the *k* subpopulation, only the *k*-th objective is considered for the purposes of the selection, and, subsequently, these subpopulations and the other genetic crossover and mutation operators apply.

The work of Coello (2006) provides an overview of the history of multiobjective optimization. This work divides the existing algorithms until then. The first of them has algorithms that feature the greater emphasis on simplicity. These algorithms include VEGA, *Nondominated Sorting Genetic Algorithm* (NSGA), *Niched-Pareto Genetic Algorithm* (NPGA) and *Multi-Objective Genetic Algorithm* (MOGA). The second generation of algorithms places greater emphasis on efficiency. Among the algorithms classified in this generation are: *Strength Pareto Evolutionary Algorithm* (SPEA), *Strength Pareto Evolutionary Algorithm* II (SPEA2), *Pareto Archived Evolution Strategy* (PAES) and *Nondominated Sorting Genetic Algorithm* II (NSGA-II). Table 1 shows a summary of the main AGMO.

Among the various methods cited to find non-dominated solutions, in this work the focus will be given to the NSGA-II algorithm. The interest in this algorithm in

particular is due to the fact that it is one of the most popular and one that has better results reported in the literature when compared to other approaches (ZITZLER; LAUMANNS; THIELE, 2001; ZONG-YI *et al*., 2008; ANTONELLI; DUCANGE; MARCELLONI, 2014; MARTIN *et al*., 2014). The operation of the NSGA-II is detailed below.

Tabela 1 – Different types of MOGA

Acronyms	Algorithm	Reference
VEGA	*Vector Evaluated Genetic Algorithm*	(SCHAFFER, 1985)
WBGA	*Weight Based Genetic Algorithm*	(HAJELA; LIN, 1992)
MOGA	*Multiple Objective Genetic Algorithm*	(FONSECA; FLEMING, 1993a)
NSGA	*Non-Dominated Sorting Genetic Algorithm*	(SRINIVAS; DEB, 1994b)
NPGA	*Niched-Pareto Genetic lgorithm*	(HORN *et al*., 1994)
PPES	*Predator-Prey Evolution Strategy*	(LAUMANNS; RUDOLPH; SCHWEFEL, 1998)
REMOEA	*Rudoph's Elitist Multi-Objective Evolutionay Algorithm*	(RUDOLPH, 2001)
NSGA-II	*Elitist Non-Dominated Sorting Genetic Algorithm*	(DEB *et al*., 2002a)
SPGA, SPEA2	*Strenght Pareto Evolutionary Algorithm* 1 e 2	(ZITZLER; THIELE, 1998; ZITZLER; LAUMANNS; THIELE, 2001)
TGA	*Thermodynamical Genetic Algorithm*	(KITA *et al*., 1996)
PAES	*Pareto-Archived Evolutionary Strategy*	(KNOWLES; CORNE, 1999)
MOMGA-I, MOMGA-II	*Multi-Objective Messy Genetic Algorithm*	(VAN VELDHUIZEN, 1999)
PESA-I, PESA-II	*Pareto Envelope-Base Selection Algorithm*	(CORNE; NOWLES; OATES, 2000; CORNE *et al*., 2001)

4.1 Nondominated Sorting Genetic Algorithm II (NSGA-II)

Proposed by (DEB *et al*., 2002a), this optimization method was based on its predecessor NSGA, which, in turn, was implemented based on the idea of (GOLDBERG, 1989). The central idea of the NSGA is to classify individuals on non-dominated borders and apply a method to diversify the solutions as much as possible.

The difference between this implementation and that of a mono-objective GA is only in the way in which the selection operator is employed. Both the crossover operators and the mutation operators are identical to those used in the mono-objective version of the GA.

NSGA-II presented solutions to problems found in NSGA, such as the high complexity of the proposed procedure for ordering non-dominance and the absence of elitism. To solve the aforementioned problems, NSGA-II defines a new procedure for ordering solutions based on the non-dominance criterion and creates a new concept, called *crowding distance*, which makes it responsible for maintaining the diversity of the population. It also defines a method called *crowded comparison*, which aims to compare the generated solutions.

NSGA-II works with a P parent population to generate a Q daughter population. In the first generation, the initial population P_0 is ordered by non-dominance. Each of the solutions present in the population P_0 receives an aptitude value according to its level of non-dominance (1 for the best level, 2 for the next and so on). After this process, the selection, crossing and mutation operators are applied, thus obtaining a daughter population Q_0. Both existing populations are N in size.

The NSGA-II algorithm pseudocode is presented in Algorithm 0.2 with all the details of the iterative process.

Algoritmo 0.2: NSGA-II

```
1  begin
2      P: Parent Population;
3      Q: Offspring Population;
4      T: Fixed size for P and Q;
5      F_i: Solution set in the frontier j;
6      n: current generation;
7      N: Max number of generation;
8      Generate the initial population P_0 randomly;
9      Q_0 = {};
10     Assign n = 0;
11     Carry out the selection, crossover and mutation in P_n to generate the offspring
         population Q_n;
12     Assign R_n = P_n ∪ Q_n;
13     Perform sorting by non-dominance in R_n, generating F_i, i = {1, ..., v} in R_n;
14     Create P_{n+1} = {};
15     while |P_{n+1}| + |F_i| ≤ N do
16         Copy the solutions F_i in P_{n+1};
17         i = i + 1
18     end
19     Calculate the crowd distance for each solution in F_i;
20     Sort F_i decreasing according to the crowd distance of each solution;
21     Copy the first N − |P_{n+1}| ordered solutions of F_i to P_{n+1};
22     if n ≥ N then
23         Stop;
24     else
25         Assign n = n + 1 and go back to step 10;
26     end
27 end
```

In NSGA-II it is necessary to calculate the crowd distance for each solution on a F_i frontier. The crowd distance *d* of a solution *j*, denoted by d_j, represents an estimate of the perimeter formed by the cuboid whose vertices are its closest neighbors. Figure 7 shows the crowd distance for a *i* solution.

The crowd distance calculation can be seen in detail in the Algorithm 0.3. where:

f_k: objective function value *k*

I^m: list ordered descending by f_k of the solutions in F_i

$d_{I_t^m}$: crowd distance of the *t*-th solution in I^m

$f_k^{(I_{t+1}^m)} and f_k^{(I_{t-1}^m)}$: f_k neighbors of the *t*-th solution in I^m

f_k^{max} and f_k^{min}: global maximum and minimum for the f_k

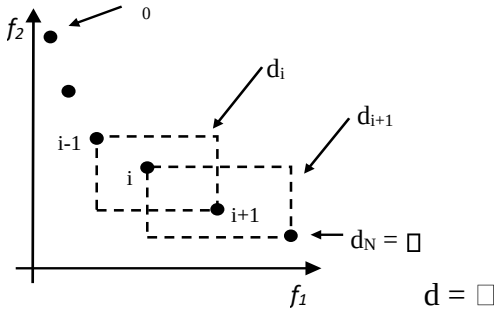


Figura 7 – NSGA-II crowd distance calculation in a two-dimensional space (DEB et al., 2002a)

Algoritmo 0.3: Crowd Distance

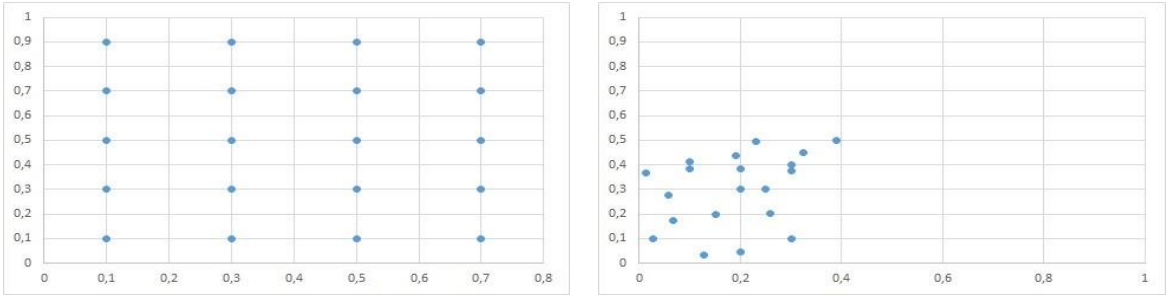
1 F_i: set of solutions at border i v_i: number of solutions at border i m: number of objective functions
2 For each j solution in F_i, assign $d_j = 0$ For each objective function $k = 1, 2,\ ldots, m$
3 Generate the list of solutions I^m equal to the set of solutions of F_i ordered descending by f_k
4 Assign $d_{I_1^m} = d_{I_{v_i}^m} = \infty$
5 For solutions $t = 2, \ldots, v_i - 1$
6 $d_{I_t^m} = d_{I_t^m} + \frac{f_k^{(I_{t+1}^m)} - f_k^{(I_{t-1}^m)}}{f_k^{\max} - f_k^{\min}}$

5 Diversity

In this section we will describe how diversity is treated in the context of MEAs and how it is treated in the context of ecosystems.

5.1 Diversity in Multiobjective Evolutionary Algorithms

In evolutionary algorithms, the term diversity indicates differences of individuals. Figure 8a is an example of a population with a high level of diversity, while Figure 8b exemplifies a population with a low level of diversity.



(a) Population with high diversity (b) Population with low diversity

Figura 8 – Population with high and low diversity.

Diversity is an important characteristic of a population, because without it there is a risk that the individual with the greatest aptitude will dominate the entire population before the search space is properly explored. This control of diversity in MEA is commonly done, using the Hamming distance pairwise as follows:

$$D(P) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} d_h\left(i_j, i_k\right) \tag{2}$$

where $dh(i_j,i_k)$ is the Hamming distance between two individuals and n is the number of individuals in the population. The shorter the distance, the less diversity.

Population diversity can also be computed using the inertia method. When calculating the inertia of a binary *string*, each bit is considered an independent dimension. In this case, the coordinates of the centroid $(c_1,c_2,c_3,...,c_L)$ of this *string* of bits P of length L are computed as:

$$c_i = \frac{\sum_{j=1}^{j=P} x_{ij}}{P} \tag{3}$$

and the inertia on the centroid is like:

$$I = \sum_{i=1}^{i=L} \sum_{j=1}^{j=P} \left(x_{ij} - c_i\right)^2 \tag{4}$$

where s_{ij} is the bit in position i of the j-th *string* and c_i is the i-th centroid coordinate.

The great disadvantage of these methods is the computational effort required to calculate the measurements in pairs for each generation of MEA. This work sought inspiration in the way how diversity is calculated in ecosystems (a subject that will be discussed in the next subsection).

5.2 Diversity in ecosystems

Ecosystem diversity is calculated based on two indices: species richness and equitability.

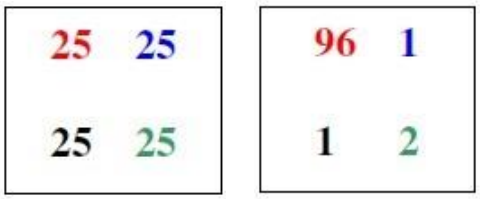
The *species richness* is simply the total number of S species in a sample unit. Consequently, the species richness is very dependent on the sample size, because the

larger the sample, the greater the number of species that can be sampled. Thus, the species richness says little about the organization of the community, increasing according to the area, even without changing the habitat.

The *equitability* expresses the way in which the number of individuals is distributed among the different species, that is, it indicates whether the different species have similar or diverging abundance (number of individuals).

Therefore, diversity is a function of the number of species and the fairness of the importance values of the same. For example, we can see in Figure 9 that both Sample A and Sample B have the same species richness (S = 4). However, Sample A has high equitability and low dominance, while Sample B has high dominance and low equitability. Thus, it can be concluded that:

1. Equitability is the inverse of dominance;
2. Sample A is more diverse than Sample B.



(a) Sample A (b) Sample B

Figura 9 – Species richness.

Table 2 lists the most used indexes in the literature for the calculation of ecosystem diversity. For these equations we have that p_i is the proportion of the species in relation to the total number of specimens found in the surveys carried out; n_i is the number of individuals of each species; N is the total number of individuals in the sample; and N_{max} is the number of individuals of the most abundant species.

Tabela 2 – Indexes of diversity in ecosystems

Index	Equation	Reference
Shannon	$I_{sh} = -\sum_{i=1}^{S} p_i \log p_i$	(CURY et al., 2005)
Simpson	$I_{si} = 1 - \frac{\sum n_i(n_i-1)}{N(N-1)}$	(KEYLOCK, 2005)
Menhinick	$I_{me} = \frac{S}{\sqrt{N}}$	(BANDEIRA et al., 2013)
Macintosh	$I_{ma} = \frac{N-\left(\sqrt{\sum n_i^2}\right)}{N-\sqrt{N}}$	(JONES et al., 2016)
BergerPaker	$I_{bp} = \frac{N_{max}}{N}$	(ALRAWASHDEH[1]; ALRAWASHDEH, 2016)

The Menhinick index (I_{me}) is a simple diversity index that considers only the number of species and the square root of the total number of individuals. The Simpson

index (I_{si}) has an advantage over the Menhinick index, as it not only considers the number of species and the total number of individuals, but also the proportion of the total occurrence of each species. The McIntosh index (I_{ma}) is a simple and more complex index than the Menhinick, as it considers the total number of individuals and the distribution between species. The Berger-Parker index (I_{bp}) is simple when compared to the Simpson index, but efficient. He considers the largest proportion of the species with the largest number of individuals. The most used index to measure the diversity of a community is the Shannon index (I_{sh}), as it incorporates both wealth and equitability. Equitability is most commonly expressed by the Pielou Index:

$$J' = \frac{H'\,(\text{observed})}{H'\,(\text{maximum})} \tag{5}$$

H'maximum being the maximum possible diversity that can be observed if all species show equal abundance whereas *H'*(*observed*) is real diversity. The value of *H'maximum* can be calculated as:

$$H'(\text{maximum}) = \log S \tag{6}$$

As can be seen, these indices are relatively simple and at the same time widely used by statisticians in ecosystems. The simplicity of these indexes is due to the fact that they do not need to compare the position of a certain individual in relation to the other, which for the computing context brings great efficiency when compared to the distance calculation methods used in the MEA.

The biggest barrier that prevents the use of these indexes as a way to measure the diversity of solutions in MEA is to be able to identify categorize the solutions as belonging to a certain species, since the search space is not divided into micro or macro regions. In this way, it is necessary to find a way to create regions in the search space in such a way that they can be interpreted with species in these statistical indexes. In the next section, we propose a way to dissolve this barrier.

6 Proposed Method

So that the statistical indexes that measure the biodiversity of ecosystems can be used within the context of the MEA, the concept of *artificial speciation* was created, in which pseudopartitions are created in the search space. Each pseudopartition represents

the region (or *habitat*) where a particular species is concentrated. It is understood, therefore, that each solution located in the pseudopartition *i* belongs to the species *i*. In this way, pseudopartitions play the same role that species play in calculating the indices in Table 2.

Figure 10 exemplifies how artificial speciation can be applied to the search space. In this figure, each black square represents a solution in the search space and each diagonal (red lines) represents the area occupied by a given species. Therefore, solutions that are on the same diagonal belong to the same species. The partitioning size (number of diagonal lines) can be adjusted according to the problem being addressed.

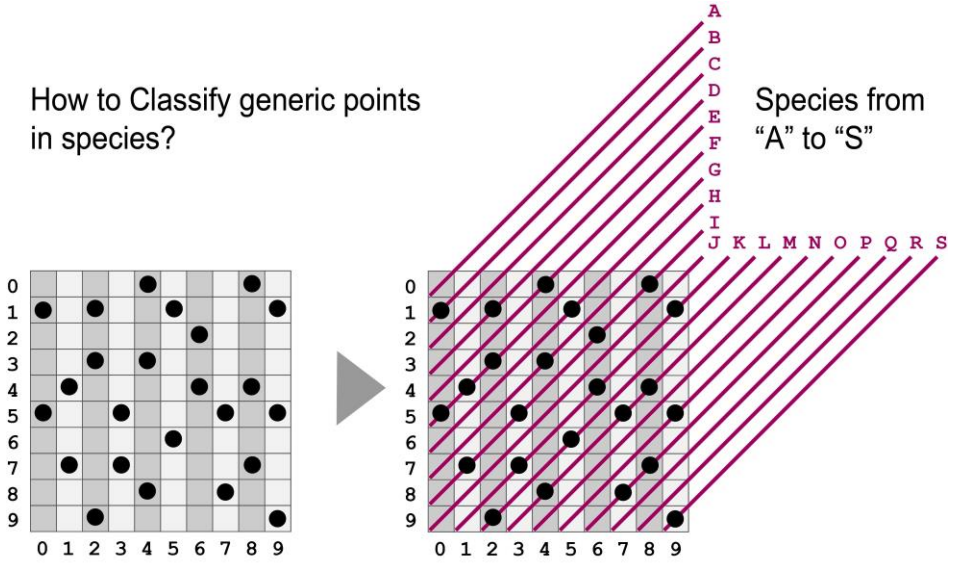


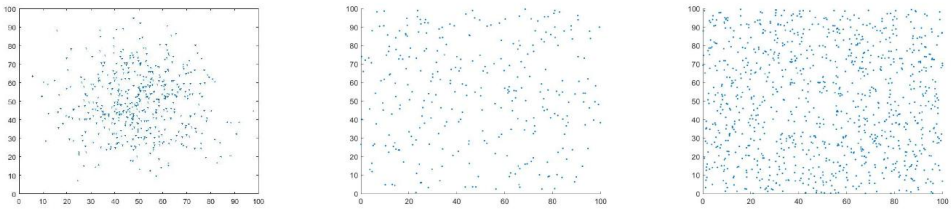
Figura 10 – Artificial Speciation Example.

Since artificial speciation makes each solution belong to a specific species, the control of the population's diversity can be done based on the statistical indexes of biodiversity. In this way, these indexes can be inserted in any class of Evolutionary Algorithm.

7 Experiments and Results

In order to understand how the use of statistical indexes behave with data in a search space in optimization problems, artificial data sets were created that are distributed in groups that may be more dense or less dense in relation to the distribution of points.

Of these data sets, 11 were created with random distribution, varying only the concentration of the points. Figure 11 exemplifies 3 of these sets.



(a) 50 points (b) 500 points (c) 1000 points

Figura 11 – Random data distribution.

Figure 11a has a tendency to concentrate at the center. This is because in real cases, MEA can converge to great locations. The purpose of this data set is to simulate the concentration in these great locations. Figure 11b simulates a situation in which the solutions would be dispersed. Figure 11c simulates the situation in which solutions are dispersed, but there are a greater number of solutions.

In these experiments 176 datasets were created that simulate 2 species that extend mainly latitudinally, 176 datasets that simulate 3 species that extend mainly latitudinally, 176 datasets that simulate 4 species that extend in elliptical shape over 4 quadrant of the search space and 176 data sets that simulate 5 species that extend in an elliptical format across 5 quadrant of the search space. Figures 12, 13 and 14 exemplify some of these data sets.

The pattern of the data created follows the pattern shown in these figures. The difference between one set and another is the number of points per group, which can be 50, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 2000, 3000, 4000, 5000 or 6000 points by group. In all, 715 artificial data sets were created.

After creating the data, artificial speciation was created on each of these 715 data sets. In order to understand what is the best way to partition data into species, we created 4 types of partitioning that we label as: **strip** partitioning, **checkered** partitioning, **ray** partitioning and **grill** partitioning.

Figure 15 exemplifies these 4 types of partitioning. In this figure, data set 579 with 8 partitions (species) was chosen arbitrarily.

We can see in Figure 15 that regardless of how the search space for this data set is partitioned, it is possible to identify that there are partitions with greater or lesser concentration of solutions. This is desirable, since the indexes will be able to reflect the diversity of the population based on this phenomenon.

In the experiments, we used the four types of partitioning for each of the 715 data sets. For the *checkered* and *grill* partitioning we tested with 2, 4, 5, 8 and 10 partitions. For the *ray* and *strip* partitioning we tested with 2, 3, 4, 5, 6, 7, 8, 9, 10 and 11 partitions. Thus, a total of 21,450 simulations were carried out.

The average of the index results for each partitioning can be seen in Figure 16. In this figure, we can see that the Shannon index was the one that best reacted to changes in diversity in the population, standing out from the other indexes. The greater the number of partitions, the smaller the area of these partitions. Being smaller, the probability of having some partitions with many solutions and others even empty is high. Because of this, the Shannon index results in higher values for scenarios where the number of partitions is higher.

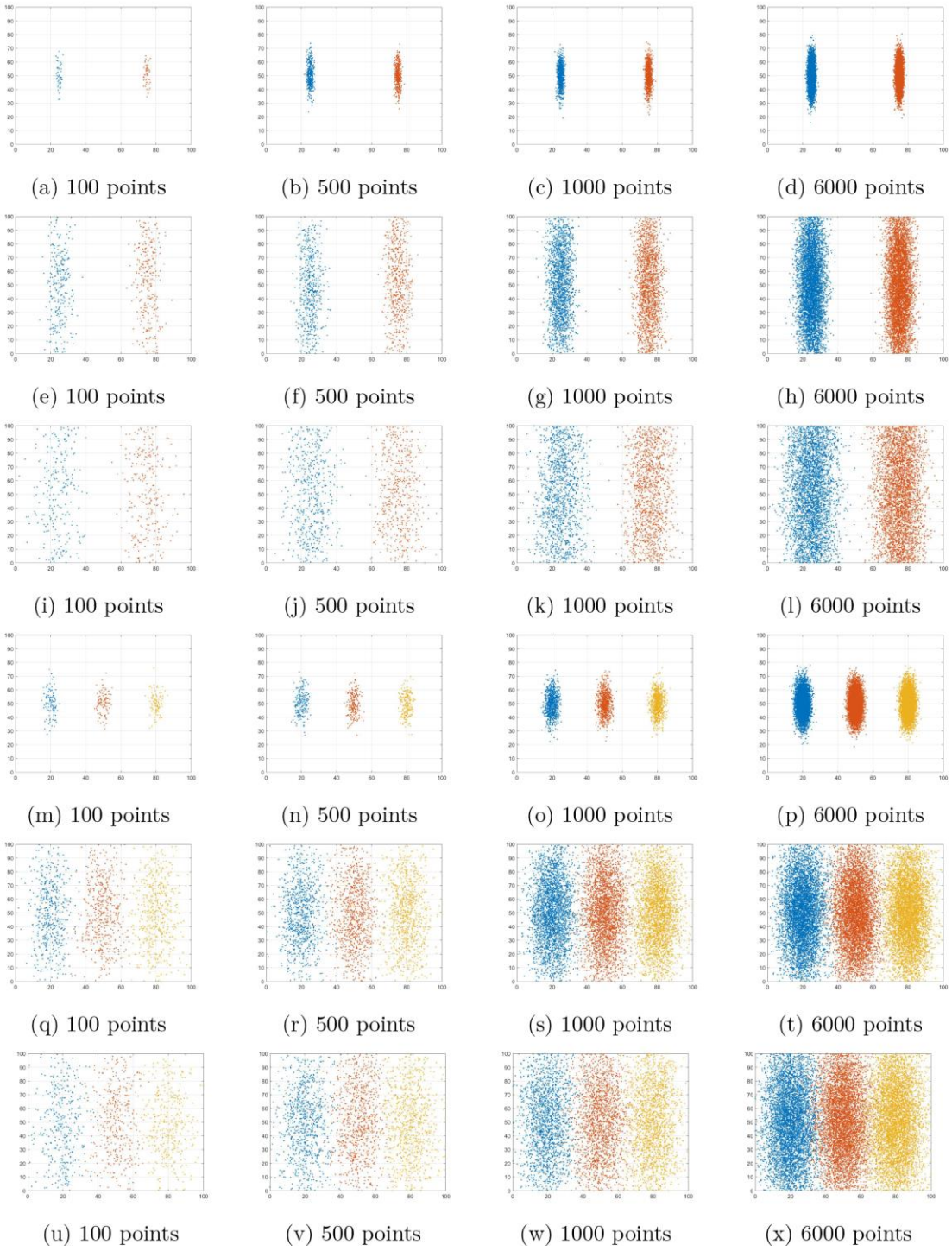


Figura 12 – Concentrated distribution over 3 latitudinal bands.

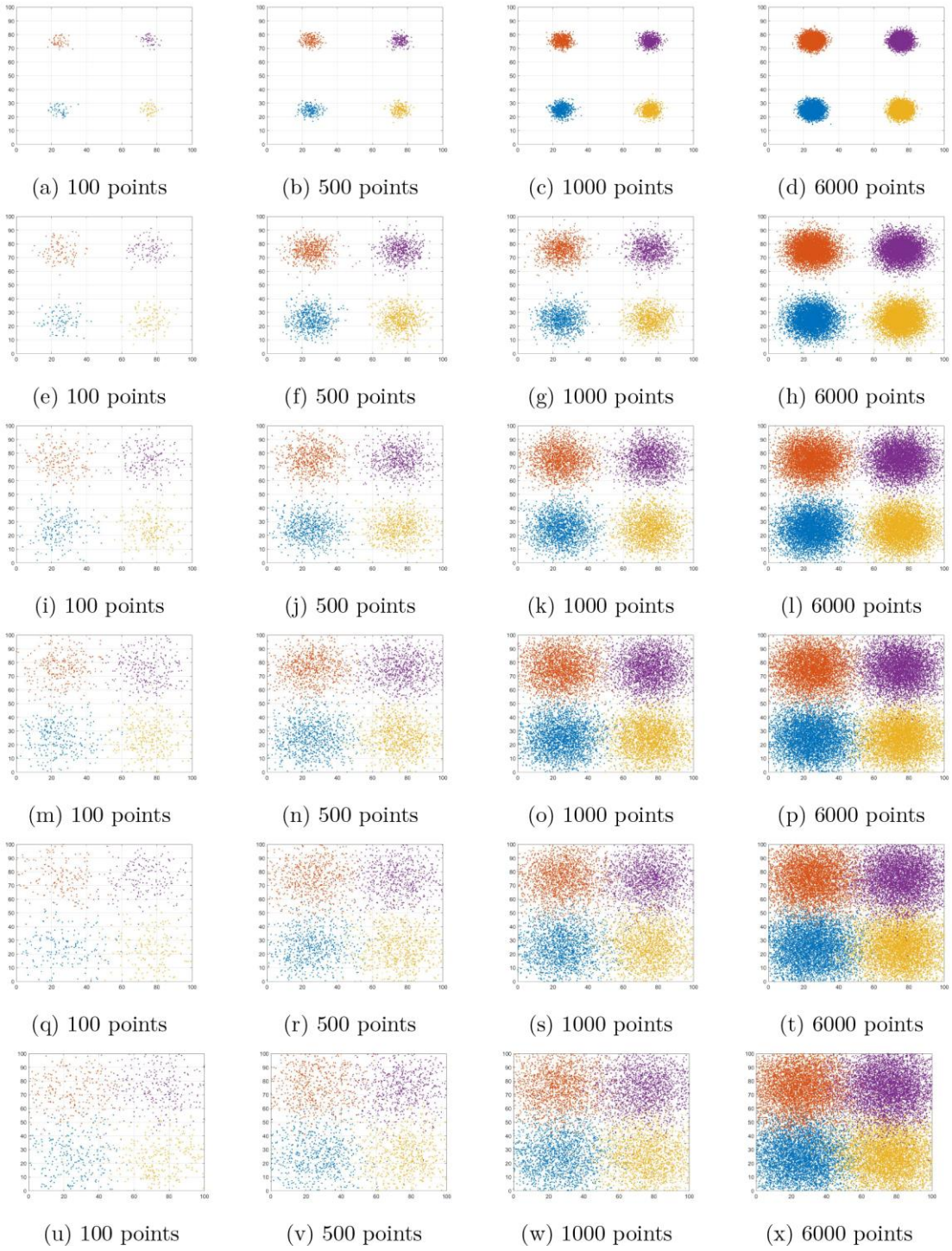


Figura 13 – Concentrated distribution over 4 elliptical areas.

Since the main focus of the MEA is on obtaining diversified solutions on the Pareto frontier, we also simulate the behavior of these statistical indices in artificial data that are distributed as if they were on a Pareto frontier whose objective is to minimize two functions. The same configurations of the previous simulations were used, but with other data sets. In these new data sets *n* points (solutions) were randomly distributed

simulating a Pareto frontier. Each simulation also changes the number of points that are distributed along the Pareto frontier. It starts with 5 points and then, with each new simulation, 10 more points are added until the limit of 305 points is reached. Figure 17 illustrates the distribution of points along the Pareto frontier. In this Figure 35 points and 8 partitions were used.

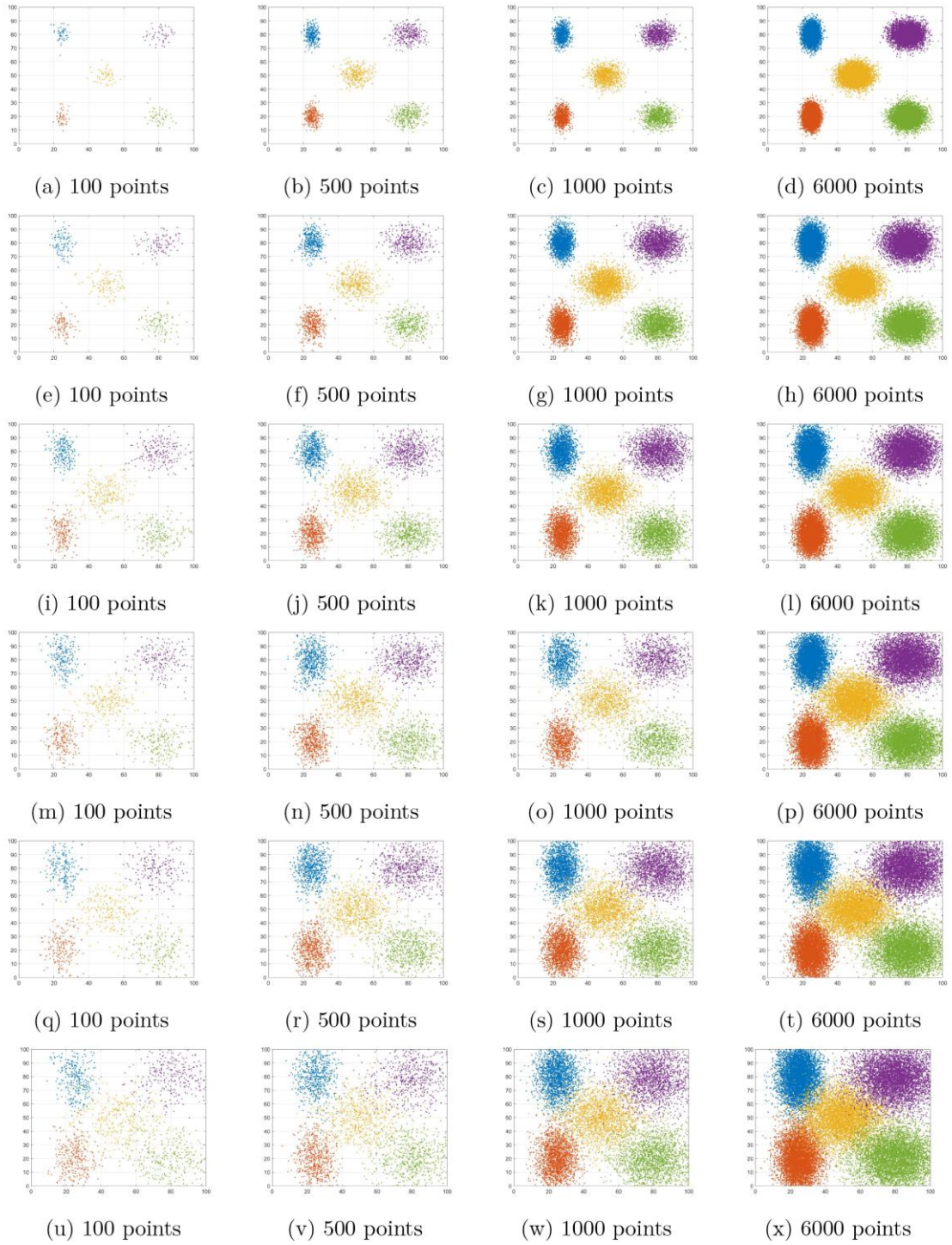
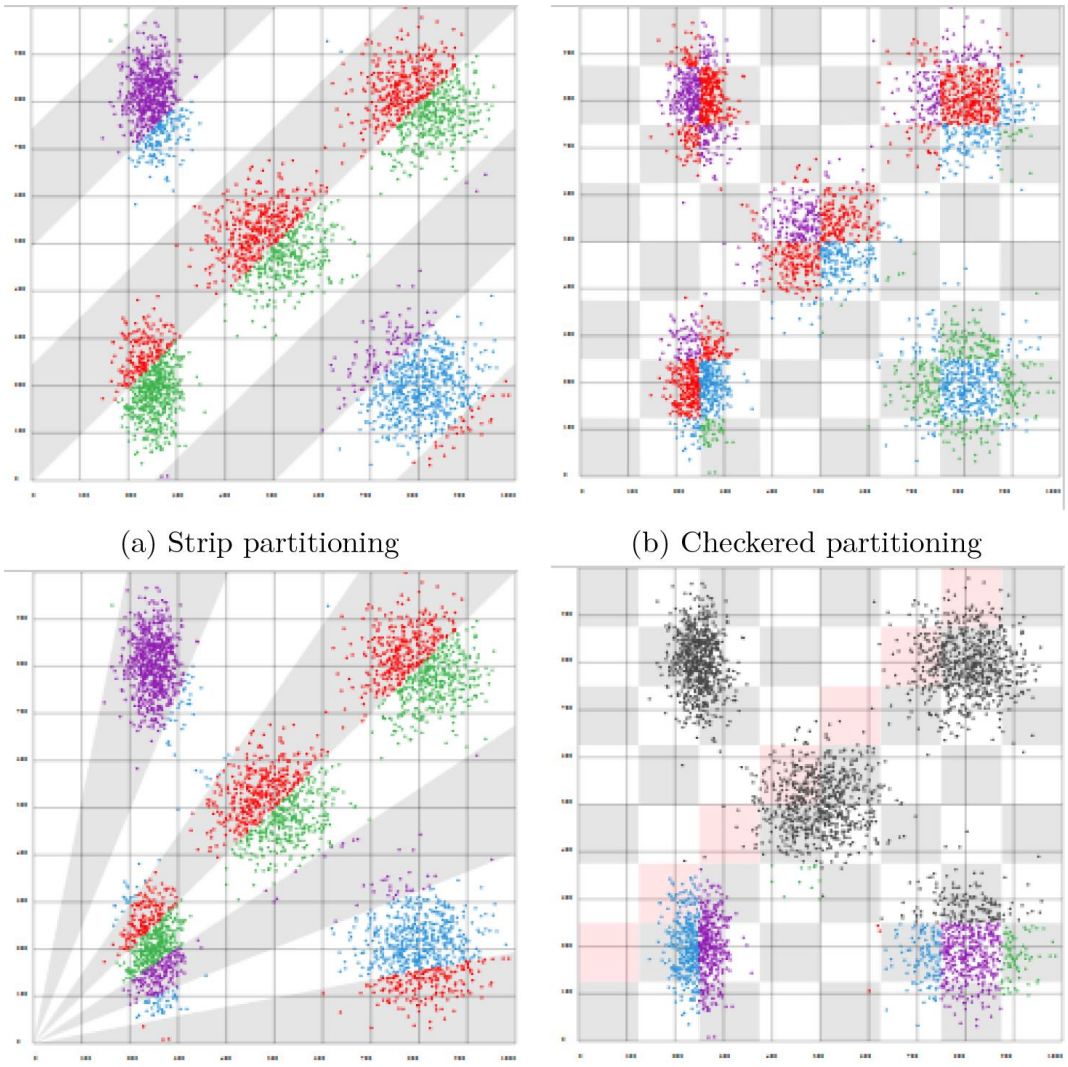


Figura 14 – Concentrated distribution over 5 elliptical areas.



(a) Strip partitioning (b) Checkered partitioning

(c) Ray partitioning (d) Grill partitioning

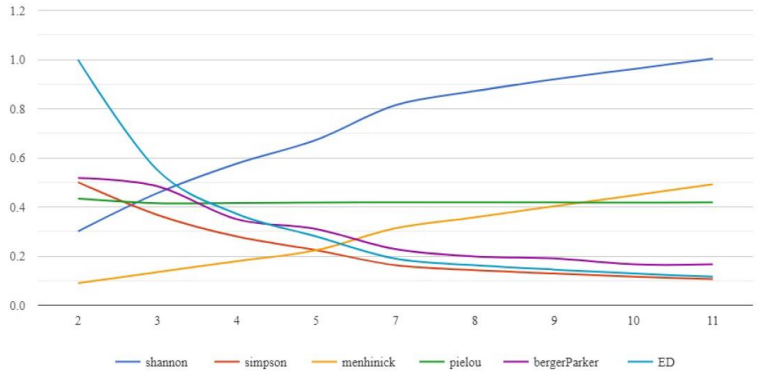
Figura 15 – Used partitioning (artificial data set 579).

The average of the indices after these simulations can be seen in Figure 18. We can see in this figure that the Shannon indexes, with the exception of chess partitioning, also better reflect the diversity at the Pareto frontier when compared to the other indexes. Therefore, we suggest adopting the Shannon index as an index to monitor the diversity of the population and the Pareto frontier in MEA.

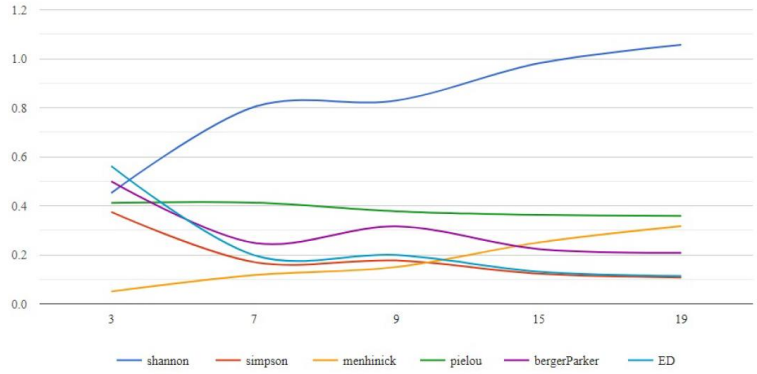
Among the 4 types of partitioning, we suggest the use of ray partitioning, as it is the one that best simulates the behavior of population convergence in the context of MEA. In this case, the rays must converge in the same direction as the population converges. This will depend on the purpose of the functions (maximization or minimization).

As a suggestion, we propose a modification of the NSGA-II (0.2) to insert the Shannon index as a control of population diversity. We chose NSGA-II because it is the

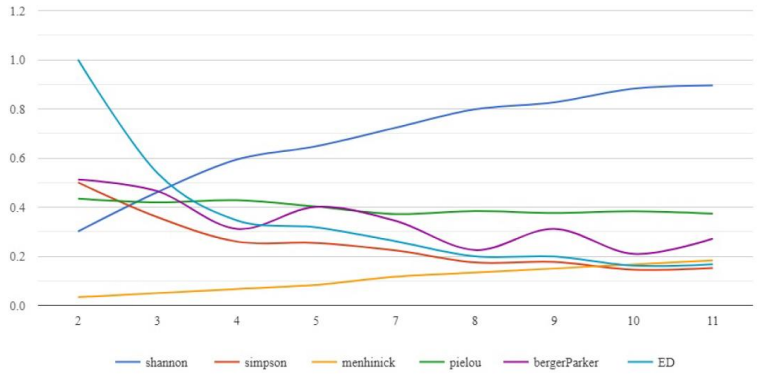
most popular MEA in the literature. Figure 19 shows how the adapted NSGA-II works and Figure 20 is the algorithm that shows how to use the shannon index to control diversity.



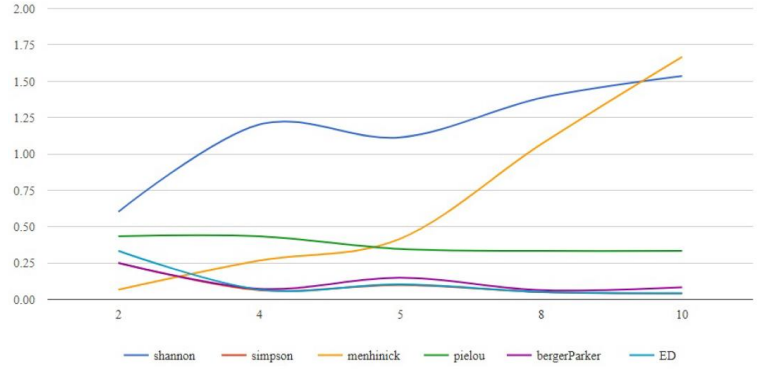
(a) Strip partitioning



(b) Checkered partitioning



(c) Ray partitioning



(d) Grill partitioning

Figura 16 – Average indexes for each partitioning.

Conclusion

In this work, we propose a more efficient way to control the diversity of the population of MEA integrating the statistical indexes applied in ecology for similar purposes.

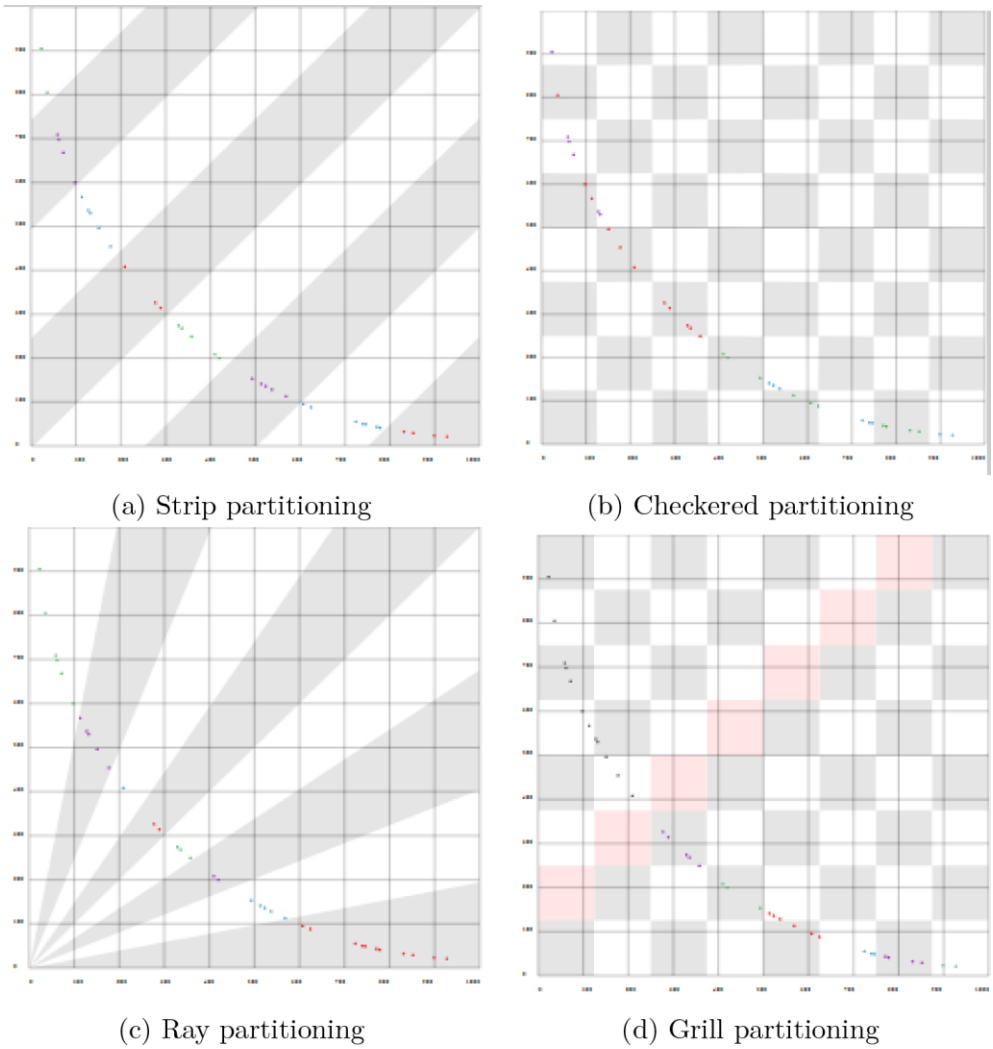


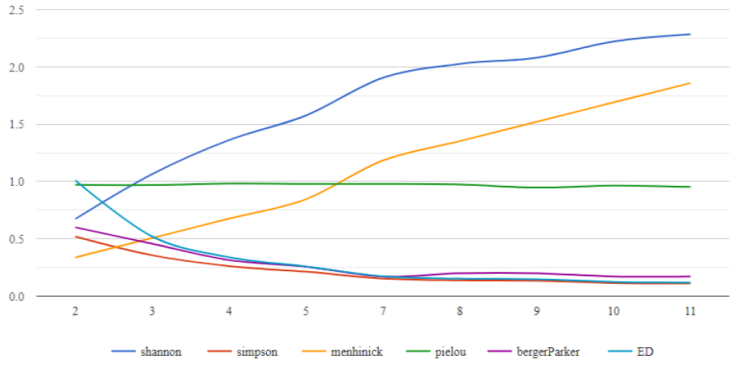
Figura 17 – Used partitioning (artificial data set 579).

The biggest problem encountered in using these indexes was the lack of a speciation technique that could separate the solutions by species. In this work, 4 ways of doing this were proposed by partitioning the search space.

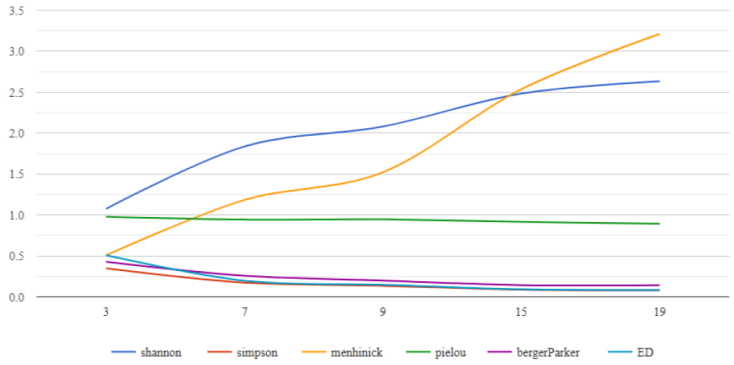
The results showed that ray partitioning is the best way to insert these indices in MEA. The index that was most suitable for the purpose proposed in this work was the Shannon index. This index clearly reflects the diversity of solutions.

We also propose an amendment to the NSGA-II as an example of how to insert statistical indexes of diversity into an MEA. As future work we want to apply the

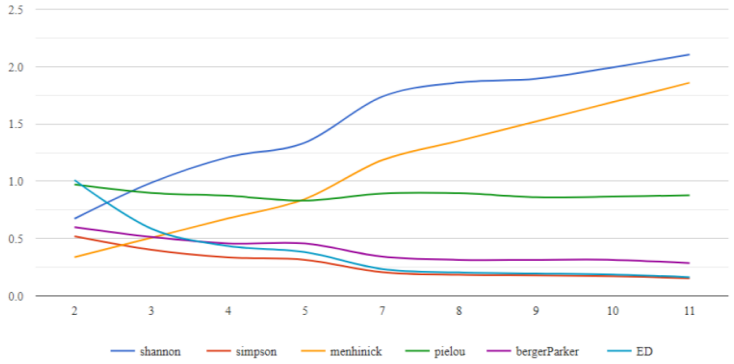
modified NSGA-II to real problems and compare the diversity of its solutions with algorithms from the literature.



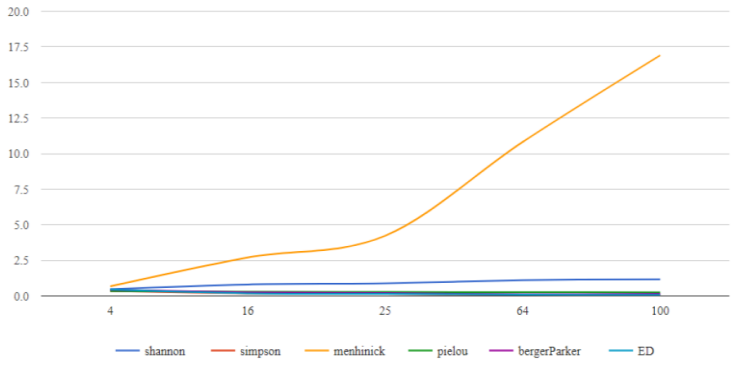
(a) Strip partitioning



(b) Checkered partitioning



(c) Ray partitioning



(d) Grill partitioning

Figura 18 – Average indexes for each partitioning considering only the Pareto frontier.

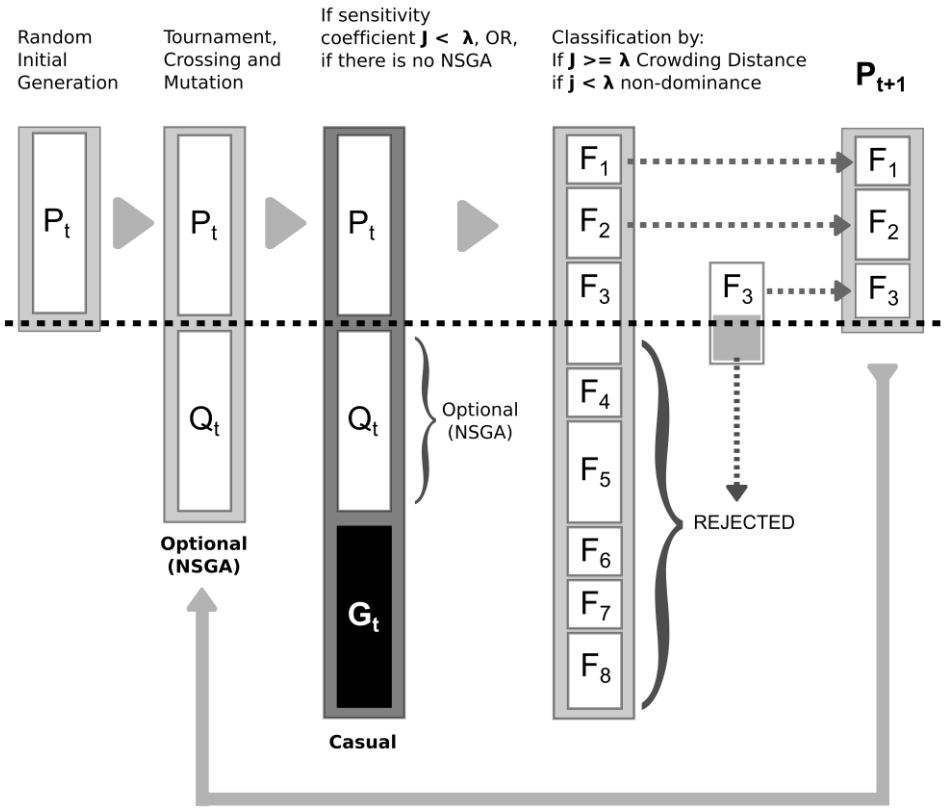


Figura 19 – Operation of the modified NSGA-II.

Algorithm 1: Adaptação do NSGA-II

Input: **int** tG, **float** λ $\subset \{0..1\}$, **bool** nsga2, **bool** nsgan, **int** tR$\subset \{5, 8, 11\}$

Output: $P_{(tG-1)}$

```
1  P_0 = Q_0 = CE_0 = {}
2  P_0 = gerarPopulacaoAleatoria()                          // Geração Inicial
3  mR = tPop / tR
   /* Loop para cada geração                                                  */
4  for t ∈ {0..tG} do
       /* Se for com NSGA-II, executa os passos puros do mesmo               */
5      if nsga2 then
6          Q_t = nsga2GerarFilhos(P_t)        // Q_t nos moldes do NSGA-II
7          R_t = P_t ∪ Q_t
8      else
9          R_t = P_t
       /* Se NSGAN habilitado, calcula o Coeficiente de Sensibilidade (J)    */
10     if nsgan then
11         J = shannon(P_t) / log(tR)  // J = shannon(P_t) 1/log(tR)
12     else
13         J = 0
       /* Executa o algoritmo fast-non-dominated-sort para identificar os
          Frontes como NSGA-II                                               */
14     if ((nsgan) AND (J≤λ)) OR (not nsga2) then
           /* => se NSGAN habilitado e sensibilizado (J ≤λ)                  */
           /* Executa SEM crowding-distance (apenas dominância comum)        */
15         F = fast-non-dominated-sort-Modificado(R_t)
           /* Verifica se NSGA-II está desabilitado e por isso, NSGAN precisa
              preencher tudo sozinho                                         */
16         puro = ((not nsga2) AND (length(P_t+1) ≤ length(P_0)))
           /* Se for o caso, NSGAN adiciona novos cromossomos                */
17         nsgan(P_t+1, tR, mR, puro ? tPop : 0)
18     else
           /* => se for NSGA-II puro                                         */
           /* Executa COM crowding-distance (original NSGA-II)               */
19         F = fast-non-dominated-sort(R_t)
       /* Passos iguais ao NSGA-II: adicionar à nova população P_t+1, os
          Frontes que cabem                                                  */
20     P_t+1 = {}                                  // nova população vazia
21     i = 0                                        // Contador de frontes
22     while (length(P_t+1) + length(P_i)) ≤tPop do
           /* Calcula o crowding-distance                                    */
23         if (nsgan) AND (J≤λ) then
24             crowding-distance-assigment-Modificado(F_i)
25         else
26             crowding-distance-assigment(F_i)
27         P_t+1 = P_t+1 ∪F_i                  // adiona o fronte à nova população
28         i += 1                                       // incrementa o fronte
29     ordenarDesc(F_i) // Ordena o fronte que não cabe, decrescentemente
       /* Deste fronte, seleciona os melhores para inclusão (corte)          */
30     P_t+1 = P_t+1 ∪F_i[0:tTop - length(P_t+1)]
31 return P_tG-1
32
```

Figura 20 – Operation of the modified NSGA-II.

Referências

ABRAHAM et al. Evolutionary multiobjective optimization. *In:* **Evolutionary Multiobjective Optimization**. [S.l.]: Springer, 2005. p. 1–6.

ABRAHAM, A.; JAIN, L.; GOLDBERG, R. **Evolutionary Multiobjective Optimization**. London: Springer-Verlag, 2005. 1–9 p. (Advanced Information and Knowledge Processing).

ALEXANDROPOULOS *et al.* Multi-objective evolutionary optimization algorithms for machine learning: A recent survey. *In:* **Approximation and Optimization**. [S.l.]: Springer, 2019. p. 35–55.

ALRAWASHDEH[1], I. M.; ALRAWASHDEH, N. Q. Species diversity and genetic diversity of paronychia argentea lam. at jerash and wadishueib ecosystems in jordan. **Species diversity**, v. 6, n. 4, 2016.

ANTONELLI, M.; DUCANGE, P.; MARCELLONI, F. A fast and efficient multi-objective evolutionary learning scheme for fuzzy rule-based classifiers. **Information Sciences**, v. 283, p. 36–54, 2014.

BäCK, T.; SCHWEFEL, H.-P. An Overview of Evolutionary Algorithms for Parameter Optimization. **Evolutionary Computation**, v. 1, n. 1, p. 1–23, mar. 1993.

BANDEIRA, B. *et al*. Mathematical convergences of biodiversity indices. **Ecological Indicators**, Elsevier, v. 29, p. 522–528, 2013.

BANZHAF, W. *et al*. **Genetic Programming: An Introduction On the Automatic Evolution of Computer Programs and Its Applications.** [S.l.]: Morgan Kaufmann Publishers Inc., 1998. 470 p.

BHOSKAR et al. Genetic algorithm and its applications to mechanical engineering: A review. **Materials Today: Proceedings**, Elsevier, v. 2, n. 4-5, p. 2624–2630, 2015.

CHAMBERS, L. **Genetic algorithms.** [S.l.]: Chapman and Hall/CRC, Boca Raton, 2001.

COELLO, C. Evolutionary multi-objective optimization: a historical view of the field. **IEEE Computational Intelligence Magazine**, v. 1, n. 1, p. 28–36, fev. 2006.

CORNE, D. *et al.* PESA-II: Region-based Selection in Evolutionary Multiobjective Optimization. In: **Proceedings of the Genetic and Evolutionary Computation Conference** *(GECCO2001)*. [S.l.: s.n.], 2001. p. 283–290.

CORNE, D.; KNOWLES, J.; OATES, M. The Pareto Envelope-Based Selection Algorithm for Multiobjective Optimization. In: SCHOENAUER, M. et al. (Ed.). **Parallel Problem Solving from Nature-PPSN VI.** Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, (Lecture Notes in Computer Science, v. 1917). p. 839–848.

CURY, P. M. *et al.* Viability theory for an ecosystem approach to fisheries. *ICES journal of marine science*, Oxford University Press, v. 62, n. 3, p. 577–584, 2005.

De Jong, K. A. **Evolutionary computation: a unified approach.** [S.l.]: MIT press, 2006.

DEB, K. *et al.* A fast and elitist multiobjective genetic algorithm: NSGA-II. **IEEE Transactions on Evolutionary Computation**, v. 6, n. 2, p. 182–197, abr. 2002.

DEB, K. *et al.* Scalable multi-objective optimization test problems. In: ***Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600).*** [S.l.]: IEEE, 2002. v. 1, p. 825–830.

DIAS, A.; VASCONCELOS, J. de. Multiobjective genetic algorithms applied to solve optimization problems. **IEEE Transactions on Magnetics**, v. 38, n. 2, p. 1133–1136, mar. 2002.

EIBEN, A. E. et al. Parameter control in evolutionary algorithms. In: **Parameter setting in evolutionary algorithms.** [S.l.]: Springer, 2007. p. 19–46.

FONSECA, C.; FLEMING, P. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. **Proceedings of the Conference on Genetic Algorithms,** p. 416–423, 1993. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.48.9077>.

FONSECA, C. M.; FLEMING, P. J. Multiobjective genetic algorithms. In: IET. **IEE colloquium on genetic algorithms for control systems engineering.** [S.l.], 1993. p. 6–1.

GEN et al. Multiobjective evolutionary algorithm for manufacturing scheduling problems: state-of-the-art survey. **Journal of Intelligent Manufacturing,** Springer, v. 25, n. 5, p. 849–866, 2014.

GOLDBERG, D. E. **Genetic Algorithms in Search, Optimization, and Machine Learning**. [S.l.]: Addison-Wesley, 1989. Addison-We. 412 p.

HAJELA, P.; LIN, C. Y. Genetic search strategies in multicriterion optimal design. **Structural Optimization,** v. 4, n. 2, p. 99–107, jun. 1992. Citado na página 12.

HAUPT, R. L.; HAUPT, S. E. **Practical genetic algorithms.** [S.l.]: John Wiley & Sons, 2004.

HOLLAND, J. H. **Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence.** [S.l.]: University of Michigan Press, 1975. 183 p.

HORN, J. *et al*. A niched pareto genetic algorithm for multiobjective optimization. In: CITESEER. **Proceedings of the first IEEE conference on evolutionary computation, IEEE world congress on computational intelligence.** [S.l.], 1994. v. 1, p. 82–87.

JONES, E. V. *et al.* A comparative analysis of three marine governance systems for implementing the convention on biological diversity (cbd). **Marine Policy**, Elsevier, v. 66, p. 30–38, 2016.

KEYLOCK, C. Simpson diversity and the shannon–wiener index as special cases of a generalized entropy. **Oikos**, Wiley Online Library, v. 109, n. 1, p. 203–207, 2005.

KITA, H. *et al*. Multi-objective Optimization by Means of the Thermodynamical Genetic Algorithm. **Lecture Notes in Computer Science**, p. 504–512, 1996.

KNOWLES *et al.* The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation. In: **Congress on Evolutionary Computation (CEC99)**. [S.l.: s.n.], 1999. v. 1, p. 98–105.

KNOWLES, J.; CORNE, D. The Pareto archived evolution strategy: a new baseline algorithm for Pareto multiobjective optimisation. In: **Proceedings of the 1999 Congress on Evolutionary Computation-CEC99.** [S.l.]: IEEE, 1999. v. 1, p. 98–105.

KNOWLES, J. D. **Local-search and hybrid evolutionary algorithms for Pareto optimization.** Tese (Doutorado) — University of Reading Reading, 2002.

KOZA, J. Genetic Programming: on the Programming of Computers by Means of Natural Selection. **Notes**, p. 1–187, 1992.

KUMAR *et al*. Improved sampling of the pareto-front in multiobjective genetic optimizations by steady-state evolution: a pareto converging genetic algorithm. **Evolutionary computation**, MIT Press, v. 10, n. 3, p. 283–314, 2002.

KUSIAK, A. Evolutionary computation and data mining. In: GOPALAKRISHNAN, B.; GUNASEKARAN, A. (Ed.). **Proc. SPIE 4192, Intelligent Systems in Design and Manufacturing III, 1.** [S.l.: s.n.], 2000. v. 4192, p. 1–10.

LAUMANNS *et al.* On the effects of archiving, elitism, and density based selection in evolutionary multi-objective optimization. In: SPRINGER. **International Conference on Evolutionary Multi-Criterion Optimization**. [S.l.], 2001. p. 181–196.

LAUMANNS *et al.* Running time analysis of multi-objective evolutionary algorithms on a simple discrete optimization problem. In: SPRINGER. **International Conference on Parallel Problem Solving from Nature.** [S.l.], 2002. p. 44–53.

LAUMANNS, M.; RUDOLPH, G.; SCHWEFEL, H.-P. A spatial predator-prey approach to multi-objective optimization: A preliminary study. In: EIBEN, A. E. *et al*. (Ed.). **Parallel Problem Solving from Nature-PPSN V**. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, (Lecture Notes in Computer Science, v. 1498). p. 241–249.

LI *et al.* Variable-length pareto optimization via decomposition-based evolutionary multiobjective algorithm. **IEEE Transactions on Evolutionary Computation**, IEEE, 2019.

LIU *et al.* Investigating the effect of imbalance between convergence and diversity in evolutionary multiobjective algorithms. **IEEE Transactions on Evolutionary Computation**, IEEE, v. 21, n. 3, p. 408–425, 2016.

MARTIN, D. *et al.* QAR-CIP-NSGA-II: A new multi-objective evolutionary algorithm to mine quantitative association rules. **Information Sciences**, v. 258, n. 0, p. 1–28, 2014. ISSN 0020-0255. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0020025513006427>.

MICHALEWICZ, Z. **Genetic Algorithms + Data Structures = Evolution Programs.** [S.l.]: Springer Science & Business Media, 1996. v. 95. 387 p.

MITCHELL, M. **An introduction to genetic algorithms.** [S.l.]: MIT Press, 1998. v. 32. 209 p.

PAQUETE *et al.* Stochastic local search algorithms for multiobjective combinatorial optimization: A. **Handbook of Approximation Algorithms and Metaheuristics: Methologies and Traditional Applications**, CRC Press, v. 1, 2018.

PEDRYCZ, W.; GOMIDE, F. **An Introduction to Fuzzy Sets:** Analysis and Design. [S.l.]: MIT Press, 1998. (A Bradford book).
RUDOLPH, G. Evolutionary Search under Partially Ordered Fitness Sets. *In:* **Proceedings of the International Symposium on Information Science Innovations in Engineering of Natural and Artificial Intelligent Systems.** [S.l.: s.n.], 2001. p. 818–822.

SCHAFFER, J. Multiple objective optimization with vector evaluated genetic algorithms. In: **Proceedings of the 1st international Conference on Genetic Algorithms.** [S.l.: s.n.], 1985. p. 93–100.

SRINIVAS, N.; DEB, K. Muiltiobjective optimization using nondominated sorting in genetic algorithms. **Evolutionary computation**, MIT Press, v. 2, n. 3, p. 221–248, 1994.

SRINIVAS, N.; DEB, K. Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. **Evolutionary Computation**, v. 2, n. 3, p. 221–248, set. 1994.

STIGLITZ *et al.* **Characterization, Existence, and Pareto Optimality in Markets with Asymmetric Information and Endogenous and Asymmetric Disclosures:** Basic Analytics of Revisiting Rothschild-Stiglitz. [S.l.], 2019.

TOFFOLO *et al.* Genetic diversity as an objective in multi-objective evolutionary algorithms. **Evolutionary computation**, MIT Press, v. 11, n. 2, p. 151–167, 2003.

VAN VELDHUIZEN, D. A. **Multiobjective Evolutionary Algorithms:** Classifications, Analyses, and New Innovations. 272 p. Tese (Doctoral) — Air Force Institute of Technology, 1999.

ZITZLER *et al.* Spea2: Improving the strength pareto evolutionary algorithm. **TIK-report**, Eidgenössische Technische Hochschule Zürich (ETH), Institut für Technische ..., v. 103, 2001.

ZITZLER, E.; LAUMANNS, M.; THIELE, L. **SPEA2:** Improving the Strength Pareto Evolutionary Algorithm. [S.l.], 2001. 21 p.

ZITZLER, E.; THIELE, L. **An Evolutionary Algorithm for Multiob jective Optimization:** The Strength Pareto Approach. [S.l.], 1998. 43 p.

ZONG-YI, X. *et al.* Multi-objective fuzzy modeling using NSGA-II. In: **IEEE International Conference on Cybernetics and Intelligent Systems, CIS 2008.** [S.l.: s.n.], 2008. ISBN 9781424416745.