



GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Aplicação de Índices Estatísticos de Biodiversidade no Controle de Diversidade das Soluções em Algoritmos Genéticos Multiobjetivo

Jean Carlo de Elias Moreira

Adinovam Henriques de Macedo Pimenta (orientador)

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 homogenizam. 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 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.

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.

Introdução

Os Problemas de Otimização Multiobjetivo (POM) geralmente envolvem a otimização simultânea de objetivos conflitantes entre si. Essa interação resulta em um conjunto de soluções comprometidas, conhecido como o conjunto ideal de Pareto (STIGLITZ et al., 2019). Como nenhuma das soluções deste conjunto pode ser considerada melhor do que as outras em relação a todos os objetivos, a meta da otimização multiobjetivo é encontrar o maior número possível de soluções ideais de Pareto (ou pareto-ótimas, como também são conhecidas) (LI et al., 2019).

Segundo Alexandropoulos et al. (2019), os Algoritmos Evolutivos Multiobjetivos (AEM) estão entre as técnicas mais poderosas para atender a esse requisito devido ao seu paralelismo inerente e à sua capacidade de explorar as semelhanças entre as soluções. A busca por soluções pareto-ótimas amplamente diversificadas não é, contudo, uma tarefa fácil (LIU et al., 2016). A maneira como as soluções são selecionadas pode causar a convergência prematura da população em porções restritas do conjunto ideal de Pareto ou até mesmo em soluções que não pertencem a este conjunto. Portanto, a manutenção da diversidade genética dentro da população é obrigatória, a fim de encontrar soluções pareto-ótimas verdadeiras e diversificadas (TOFFOLO et al., 2003).

Após a publicação do trabalho de Goldberg (1989) que sugeriu o uso da não-dominância das soluções como um critério a ser observado durante o processo de otimização, gerou um imenso interesse pelos AEM. Os AEM iniciais - *Multiobjective Genetic Algorithm* (MOGA) (FONSECA; FLEMING, 1993b), *Non-dominated Sorting Genetic Algorithm* (NSGA) (SRINIVAS; DEB, 1994a) e *Niched Pareto Genetic Algorithm* (NPGA) (HORN et al., 1994) – exploraram intensamente esta sugestão.

Estes três algoritmos mostraram que diferentes maneiras de implementar o conceito de não-dominância pode resultar em AEM bem-sucedidos. No entanto, esses algoritmos não puderam garantir a convergência para o conjunto ideal de Pareto, pois estava faltando um operador para preservação das melhores soluções (elitismo). Assim, os AEM mais recentes concentraram-se principalmente em como o elitismo poderia ser introduzido em um AEM. Isso resultou em três dos AEM mais bem sucedidos da literatura: *Strength Pareto Evolutionary Algorithm* (SPEA) (ZITZLER et al., 2001), *Pareto Archived Evolution Strategy* (PAES) (KNOWLES et al., 1999) e *Non-dominated Sorting Genetic Algorithm II* (NSGA-II) (DEB et al., 2000), sendo este último o mais popular. Com o desenvolvimento de melhores algoritmos, os AEM também foram utilizados em várias aplicações práticas (LAUMANN et al., 2001).

Existe, todavia, uma lacuna na literatura apontando para a ausência de estudos e análises relacionados à garantia de soluções pareto-ótimas diversificadas, bem como forma de monitorar a diversificação das soluções ao longo do processo genético dos AEM (BHOSKAR et al., 2015; GEN et al., 2014; PAQUETE et al., 2018). Nesse sentido, Kumar et al. (2002) e Abraham et al. (2005) sugeriram uma série de algoritmos, os quais garantem convergência para soluções pareto-ótimas, mas não abordam os dois aspectos a seguir:

- 1.Os algoritmos convergentes não garantem a manutenção da diversidade das soluções;
- 2.Os algoritmos não especificam a complexidade de tempo para sua convergência.

Embora seja difícil realizar a segunda tarefa, mesmo para funções objetivas simples (ver (LAUMANNS et al., 2002)) e também para problemas mono-objetivos, a primeira tarefa é tão importante quanto a tarefa de convergir para o conjunto ideal de Pareto. Deb et al. (2002b) sugeriram um AEM que tenta manter a diversidade enquanto converge para soluções pareto-ótimas. Entretanto, não há provas para suas propriedades de convergência. Knowles (2002)

apresentou uma estratégia de arquivamento adaptativo. Essa estratégia fornece soluções em algumas regiões "críticas" do conjunto de Pareto, mas a convergência só pode ser garantida para as soluções nas extremidades do conjunto de Pareto, promovendo a aglomeração de soluções em regiões extremas das funções-objetivo e, conseqüentemente, diminuindo a diversidade das soluções.

Desta forma, o objetivo principal deste trabalho é identificar um índice estatístico que mensura a biodiversidade de um ecossistema e propor uma maneira de integrar este índice aos AEM, proporcionando aos AEM uma maneira de monitorar a diversidade das soluções que estão sendo geradas durante o processo evolutivo da população.

1 Computação Evolutiva

A Computação Evolutiva (CE), também conhecida como Computação Evolucionária, é um ramo de pesquisa que compreende um conjunto de técnicas de busca e otimização baseadas nos princípios da evolução biológica, como a seleção natural e a herança genética, para resolver diferentes tipos de problemas. Essas técnicas estão sendo cada vez mais utilizadas para resolver uma grande variedade de problemas, que vão desde as aplicações práticas na indústria e comércio até a investigação científica (EIBEN et al., 2007).

Por meio de técnicas da CE, cria-se uma população de indivíduos que se reproduzem e competem entre si pela sobrevivência. Os melhores indivíduos possuem maiores chances de sobreviver e de transferir suas características às novas gerações.

As técnicas principais da CE são as Estratégias Evolutivas (EE), a Programação Evolutiva (PE), a Programação Genética (PG) e os Algoritmos Genéticos (AG) (PEDRYCZ; GOMIDE, 1998; BANZHAF et al., 1998). Todas estas técnicas utilizam o mesmo princípio básico da CE, mas operam de formas distintas.

As EE codificam uma população em um vetor único de indivíduos com valores reais e a sua ideia principal é a combinação dos indivíduos, mediante cruzamento, para gerar um descendente que substituirá o pior indivíduo da população. O primeiro algoritmo usando uma estratégia evolutiva foi desenvolvido em 1964 na Universidade Técnica de Berlim (KUSIAK, 2000).

Já a PE utiliza a predição do comportamento de máquinas de estado finito e sua otimização (PEDRYCZ; GOMIDE, 1998). Dessa forma, cada indivíduo representa uma máquina de estado finito. A seleção dos indivíduos é baseada no elitismo.

A PG é uma técnica de geração automática de programas de computador proposta por Koza (1992), inspirada na teoria dos AG proposta por Holland (1975). É possível criar e manipular software geneticamente usando-se PG e aplicando conceitos herdados da biologia para gerar programas de computador automaticamente.

Os AG são algoritmos que modelam uma solução para um problema específico em uma estrutura de dados e aplicam operadores que recombina essas soluções preservando informações críticas. Os AG foram concebidos por Holland (1975), com o objetivo inicial de estudar os fenômenos relacionados à adaptação das espécies e da seleção natural que ocorrem na natureza, bem como desenvolver uma maneira de incorporar estes conceitos aos computadores (MITCHELL, 1998). Os AG e a PG são as duas principais frentes de pesquisa na computação evolutiva. Na próxima seção são apresentados os conceitos principais de AG. A PE, PG e as EE não serão estudadas nesta proposta. Informações mais detalhadas sobre estas técnicas podem ser encontradas em diversos trabalhos (Bäck; SCHWEFEL, 1993; De Jong, 2006; Eiben et al., 2007).

2 Otimização Multiobjetivo

Em um problema de otimização simples (mono-objetivo) o espaço de busca geralmente é bem definido. Mas quando o problema tem múltiplos objetivos, contraditórios entre si, não existe apenas uma única solução ótima, mas sim um conjunto de possíveis soluções.

Mesmo que alguns problemas do mundo real possam ser reduzidos a um único objetivo, muitas vezes é difícil resumir todas as características do problema em um único objetivo e resolvê-lo utilizando técnicas de otimização mono-objetivo, como, por exemplo, os AG (ABRAHAM; JAIN; GOLDBERG, 2005). Assim, definir múltiplos objetivos frequentemente é a melhor forma de resolver um problema multiobjetivo.

A otimização multiobjetivo busca encontrar soluções de problemas que possuem objetivos que sejam conflitantes, ou seja, caso seja possível melhorar o resultado de um dos objetivos, o outro objetivo do problema ficará prejudicado. Diz-se que uma função f_1 é conflitante com uma função f_2 quando não é possível, ao mesmo tempo, melhorar o valor de ambas.

Um exemplo didático que ilustra bem este tipo de problema é o de uma viagem de automóvel em que se deseja minimizar o tempo de viagem e o consumo de combustível (custo). Sabe-se que quanto mais se acelera um veículo para diminuir o tempo de viagem, mais se gasta combustível. Matematicamente, o problema de otimização multiobjetivo pode ser formulado como apresentado por (DIAS; VASCONCELOS, 2002), pela Equação 1, como segue.

$$\begin{aligned}
 & \text{maximizar/minimizar } f_m(\mathbf{x}), & m = 1, 2, \dots, & \boxed{} \\
 & \text{Restrita a } g_j(\mathbf{x}) \geq 0, & j = 1, 2, \dots, J & \boxed{} \\
 & & & \boxed{} \\
 & h_k(\mathbf{x}) = 0, & k = 1, 2, \dots, K & \boxed{} \\
 & \mathbf{x}^{(L)} \leq \mathbf{x} \leq \mathbf{x}^{(U)} & & \boxed{} \boxed{} \\
 & i = 1, 2, \dots, n & &
 \end{aligned} \tag{1}$$

em que $\mathbf{x}$ é o vetor de n variáveis de decisão $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$. Os valores $x^{(L)}$ e $x^{(U)}$ são os limites inferior e superior das variáveis de decisão, respectivamente.

x_i , representam o mínimo e máximo valores respectivamente para a variável x_i . Estes limites definem o *espaço das variáveis de decisão* ou *espaço de decisão* D . Além disso, o vetor $\mathbf{x}$ será referido também como *solução*.

As J desigualdades (g_j) e as K igualdades (h_k) são chamadas de funções de restrição. Uma solução $\mathbf{x}$ factível será aquela que satisfaça as $J + K$ funções de restrição e os $2n$ limites. Caso contrário a solução será não factível. O conjunto de todas as soluções factíveis formam a *região factível* ou *espaço de busca* S .

Cada uma das M funções objetivo $f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_M(\mathbf{x})$ pode ser maximizada ou minimizada. O vetor funções objetivo $\mathbf{f}(\mathbf{x})$ forma um espaço multi-dimensional chamado espaço de objetivos Z . Então, para cada solução $\mathbf{x}$ no espaço de decisão, existe um $\mathbf{f}(\mathbf{x})$ no espaço de objetivos. Esta é uma diferença fundamental em relação à otimização de objetivos simples, cujo espaço de objetivos é unidimensional.

O mapeamento acontece então entre um vetor $\mathbf{x}$ (n -dimensional) e um vetor $\mathbf{f}(\mathbf{x})$ (M -dimensional). Por exemplo, se cada elemento de $\mathbf{x}$ e $\mathbf{f}(\mathbf{x})$ é um número real, então $\mathbf{f}(\mathbf{x})$ é definida como $\mathbf{f}(\mathbf{x}): \mathbb{R}^n \rightarrow \mathbb{R}^M$.

2.1 Soluções Pareto-ótimas

Tomar decisões implica em um processo que consiste de vários fatores e cujo objetivo é encontrar a melhor solução. Em alguns casos, é possível encontrar várias soluções boas, sendo que nenhuma é quantitativamente melhor do que a outra. Por exemplo, ao se comprar um carro, pode-se supor-se que se está procurando um carro com o menor preço e o maior conforto. A Figura 1 ilustra várias alternativas de escolha.

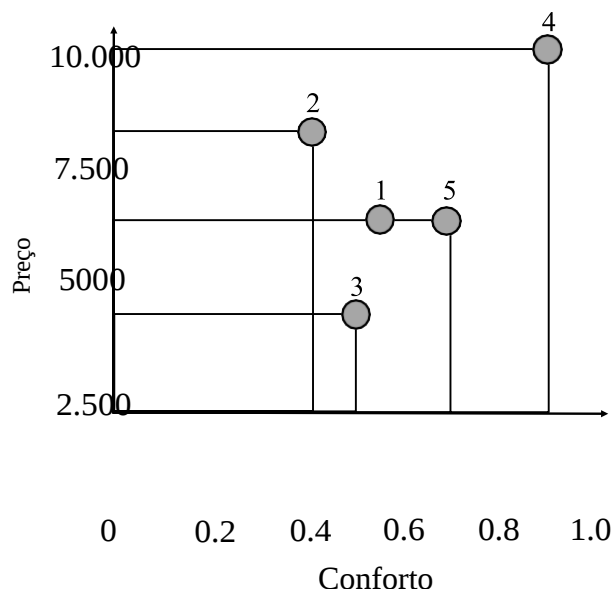


Figura 1 – Exemplo de compra de carro, considerando o custo e o conforto.

O objetivo é minimizar o custo e maximizar o conforto. Neste caso tem-se cinco possíveis opções de compra. Intuitivamente, descarta-se a solução 1, já que a solução 5 fornece mais conforto por igual preço. A solução 2 também é descartada, já que as soluções 3 e 5 fornecem mais conforto que a solução 2 e por um preço menor. Tem-se então três soluções (3, 4 e 5) que são boas alternativas de compra. Em termos quantitativos, nenhuma é melhor do que a outra, pois uma é mais confortável, porém menos barata, e vice-versa. Existe então um balanceamento entre os objetivos. Quanto maior o conforto, maior é o preço do carro e quanto mais barato, menor é o conforto do carro.

Diz-se que uma solução *domina* uma outra se os seus valores são melhores em todos os objetivos, ou é melhor em um dos objetivos e empata nos outros. Por exemplo, a solução 1 domina a solução 2, mas é dominada pelas soluções 4 e 5. Nenhuma solução domina as soluções 3, 4, e 5. Pode-se dizer que as soluções 3, 4, e 5 são igualmente boas. Portanto, existe um conjunto de soluções ótimas. A este conjunto dá-se o nome de *conjunto de soluções não dominadas*. As outras soluções (1 e 2) formam o *conjunto de soluções dominadas*.

Considerando que os pontos não dominados estão em um espaço contínuo, pode-se desenhar uma curva ligando esses pontos. Todos os pontos contidos na curva formam a *frete de Pareto* ou *fronteira de Pareto*.

2.2 Metas em Problemas de Otimização Multiobjetivo

Em um problema de otimização multiobjetivo, todas as soluções Pareto-ótimas, ou seja, as soluções encontradas na fronteira de Pareto, são igualmente importantes. Deb et al. (2002a)

destaca duas importantes metas em otimização multiobjetivo:

- Encontrar um conjunto de soluções que estejam o mais próximo possível da fronteira de Pareto;
- Encontrar um conjunto de soluções não dominadas com a maior diversidade possível.

A primeira meta é comum para qualquer processo de otimização. Soluções muito distantes da Fronteira de Pareto não são desejáveis, pois isto indica que estas soluções estão distantes das melhores soluções. Porém, encontrar a maior diversidade dentro das soluções não dominadas é uma meta específica para a otimização multiobjetivo. A Figura 2a apresenta uma fronteira de Pareto com uma boa distribuição de soluções quando comparada com a da apresentada na Figura 2b.

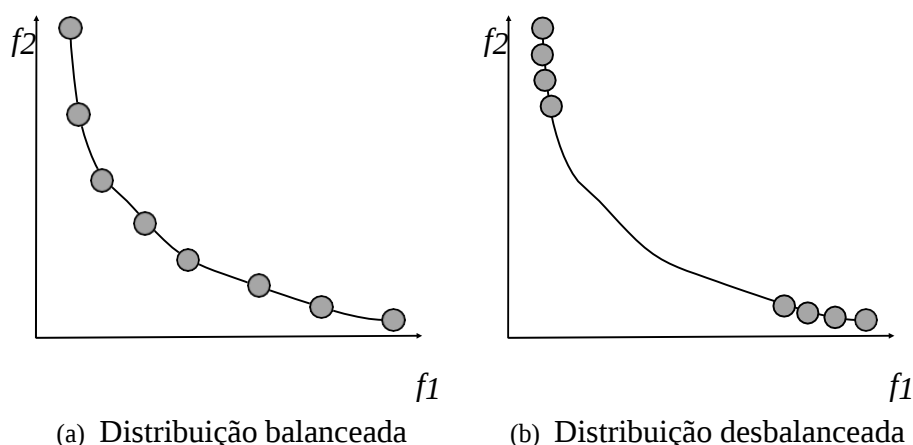


Figura 2 – Distribuição das soluções ao longo da fronteira de Pareto

Na Figura 2b as soluções estão distribuídas apenas em algumas regiões, sendo que cada região privilegia um dos objetivos, o que caracteriza essas soluções como *desbalanceadas*. É necessário assegurar a maior cobertura possível da fronteira, fazendo com que se obtenha um conjunto de soluções com os objetivos balanceados.

2.3 Diferenças entre Otimização Mono-objetivo e Multiobjetivo

Deb et al. (2002a) identifica três diferenças importantes entre a otimização multiobjetivo e a otimização de mono-objetivo:

- Em problemas de otimização mono-objetivo, a meta é achar uma solução ótima global (máximo ou mínimo). Em problemas multiobjetivo pode existir mais de um ótimo global. Achar o conjunto de soluções da fronteira de Pareto é tão importante quanto preservar a diversidade neste conjunto. Um algoritmo eficiente para otimização multiobjetivo deve considerar ambos os aspectos;
- Em problemas de otimização multiobjetivo são considerados dois espaços (o de variáveis e o de objetivos) ao invés de um. Problemas de otimização mono-objetivo trabalham unicamente no espaço de variáveis, já que procuram apenas uma solução no espaço de objetivos. Manter a diversidade em ambos espaços aumenta a complexidade do problema, dado que a proximidade entre duas soluções no espaço de variáveis não implica proximidade no espaço de objetivos;

- Os métodos tradicionais de otimização multiobjetivo estão baseados em uma função simples que pondera cada objetivo. Podem também tratar cada objetivo separadamente, utilizando

os demais objetivos como restrições. Portanto, um problema de otimização multiobjetivo pode ser convertido em um problema de otimização simples.

2.4 Operador de Dominância de Pareto

Os problemas multiobjetivos, diferentemente dos problemas mono-objetivos, possuem mais de uma solução. Estas soluções são conhecidas como *soluções não-dominadas* ou *soluções eficientes*.

Se existem M funções objetivo $f_m, m = 1, \dots, M$ o operador / entre duas soluções, x/y , significa que a solução x é melhor do que a solução y em pelo menos um objetivo em particular. Reciprocamente $x \cdot y$ denota que a solução x é pior do que a solução y para algum objetivo.

Diz-se que uma solução $\mathbf{x}^{(1)}$ domina uma outra solução $\mathbf{x}^{(2)}$ (representado matematicamente como $\mathbf{x}^{(1)} \prec \mathbf{x}^{(2)}$) se as seguintes condições são satisfeitas:

- A solução $\mathbf{x}^{(1)}$ não é pior que $\mathbf{x}^{(2)}$ em todos os objetivos, ou seja, $f_m(\mathbf{x}^{(1)}) \not\geq f_m(\mathbf{x}^{(2)})$ para todo $m = 1, 2, \dots, M$;
- A solução $\mathbf{x}^{(1)}$ é estritamente melhor que $\mathbf{x}^{(2)}$ pelo menos em um objetivo, ou seja, $f_m(\mathbf{x}^{(1)}) < f_m(\mathbf{x}^{(2)})$ pelo menos em um $m = 1, 2, \dots, M$.

Se ambas as condições são satisfeitas, pode-se dizer que $\mathbf{x}^{(1)}$ domina $\mathbf{x}^{(2)}$:

- $\mathbf{x}^{(1)}$ domina $\mathbf{x}^{(2)}$;
- $\mathbf{x}^{(1)}$ é não dominada por $\mathbf{x}^{(2)}$;
- $\mathbf{x}^{(1)}$ é não inferior que $\mathbf{x}^{(2)}$.

Se a solução $\mathbf{x}^{(1)}$ não é dominada por nenhuma solução $\mathbf{x}^{(i)}$ qualquer, em todo o espaço factível, diz-se que $\mathbf{x}^{(1)}$ é uma solução eficiente, não-dominada ou solução Pareto-ótima.

Na Figura 1, a solução 5 domina a solução 1 ($5 \prec 1$), e a solução 3 domina a solução 2 ($3 \prec 2$). Portanto, o conceito de dominância permite comparar soluções com múltiplos objetivos.

Assim, utilizando estas definições, quando um conjunto de soluções finito é encontrado, torna-se possível realizar comparações das soluções duas a duas, dividindo-as em um grupo chamado de soluções dominadas P e de soluções não-dominadas P' . As soluções de P' são não-dominadas por qualquer outra solução presente em P . Se o conjunto não-dominado P' abrange a totalidade do espaço de busca factível, ele é chamado de conjunto Pareto-ótimo global. A Figura 3 ilustra os espaços das variáveis de decisão e dos objetivos. É também mostrado nesta figura a fronteira Pareto-ótima global.

A fronteira Pareto-ótima ilustrada na Figura 3 é formada por valores das funções objetivo $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_m(\mathbf{x}))$ correspondentes a cada solução no espaço de busca. Logo, para cada uma das soluções encontradas no espaço de variáveis, estas soluções são representadas no espaço dos objetivos, avaliando cada uma delas em cada um dos objetivos existentes.

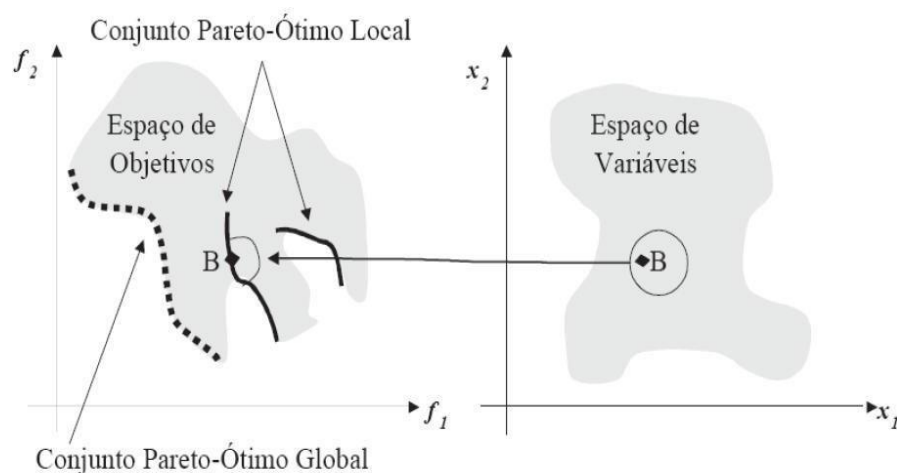


Figura 3 – Soluções Pareto-ótimo locais e globais.

3 Algoritmos Genéticos

Os AG podem ser definidos como métodos iterativos que contêm uma população com um número fixo de indivíduos, em que cada um deles é representado por uma sequência linear finita de símbolos, denominada cromossomo, que codificam uma possível solução para um dado problema, ou, dependendo da codificação, parte da solução. Essas possíveis soluções são codificadas em um espaço, chamado de espaço de busca, que pode conter todas as soluções possíveis para o problema em questão. A cada cromossomo é associado um valor de aptidão e este valor representa quão adequada ao problema é a solução que ele codifica. Dessa forma, os indivíduos podem ser alterados até que se obtenha um indivíduo que codifique uma solução satisfatória para o problema abordado.

Os AG são frequentemente usados em problemas de busca, tendo encontrado ampla aplicação em diversas áreas científicas, entre as quais podem ser citadas aquelas relativas a problemas de otimização de soluções, aprendizado de máquina, desenvolvimento de estratégias e fórmulas matemáticas, análise de modelos econômicos, problemas de engenharia, diversas aplicações na biologia tais como simulação de bactérias, sistemas imunológicos, ecossistemas e descoberta de topologias e propriedades de moléculas orgânicas (GOLDBERG, 1989; HAUPT; HAUPT, 2004; CHAMBERS, 2001; MICHALEWICZ, 1996; MITCHELL, 1998).

O funcionamento de um AG dá-se a partir do processo de inicialização, que cria uma *população* de indivíduos representando soluções iniciais que são, então, submetidas ao processo de evolução, que passa pelas seguintes etapas: avaliação, seleção, cruzamento e mutação. Cada uma destas etapas serão abordadas a seguir.

3.1 Avaliação

É nesta fase que é dado o primeiro passo para a seleção. Cada indivíduo recebe uma avaliação de acordo com o seu grau de aptidão, ou seja, mede-se quão bom é o indivíduo para solucionar o problema em questão. Levando-se em conta que esta tarefa é executada para cada indivíduo a cada geração, pode-se dizer que o seu custo computacional é relativamente alto.

3.2 Seleção

Na etapa de seleção, os indivíduos são selecionados para a próxima etapa, o cruzamento. Utiliza-se o grau de aptidão de cada indivíduo para fazer um sorteio no qual os indivíduos mais aptos possuem maior probabilidade de serem escolhidos.

A seguir são listados alguns métodos de seleção.

Roleta: Neste método os indivíduos de uma população são selecionados para a próxima geração utilizando uma roleta. Cada indivíduo da população é representado na roleta por uma fatia proporcional ao seu índice de aptidão. Assim, indivíduos com maior aptidão ocupam fatias maiores da roleta, enquanto indivíduos com menor aptidão ocupam fatias menores. Para a seleção dos indivíduos, a roleta é girada N vezes, onde N é número da população inicial. A cada vez que a roleta parar de girar, o cromossomo selecionado pelo marcador será copiado para a próxima geração. Cromossomos com maior espaço na roleta terão maior chance de serem selecionados.

Na Figura 4 é apresentada uma roleta que reflete os valores de aptidão dos 5 indivíduos (S1, S2, S3, S4 e S5) de uma população, sendo que o indivíduo S2 (com a maior fatia) foi selecionado pelo marcador.

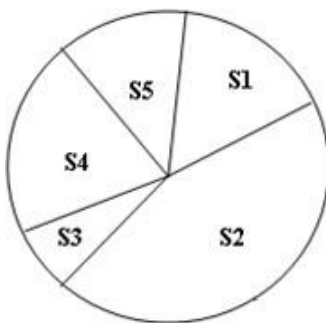


Figura 4 – Método de seleção por roleta.

Torneio: Quando este método é utilizado, um número n de indivíduos da população é escolhido de modo aleatório e com mesma probabilidade. O cromossomo com maior aptidão dentre estes n cromossomos é selecionado para a população intermediária. Este processo se repetirá até que a população intermediária seja preenchida.

Pode-se, ainda, adicionar a esses métodos uma técnica chamada *elitismo*. Nesta técnica todos os indivíduos são ordenados de acordo com o seu valor de aptidão e os indivíduos que ficarem abaixo do valor médio são descartados. Esta técnica resulta em uma população que terá indivíduos com valor de aptidão acima da média da população anterior. Esta técnica também garante que o melhor indivíduo obtido na geração atual estará presente na próxima geração que está sendo gerada, preservando suas características genéticas.

3.3 Cruzamento

Selecionados os indivíduos, eles passam com uma probabilidade pré-estabelecida pelo processo de cruzamento. Essa probabilidade é chamada de taxa de cruzamento. Neste processo são selecionados dois indivíduos para o cruzamento, gerando dois novos indivíduos que irão compor a próxima geração substituindo, assim, os dois indivíduos selecionados.

Dentre os métodos de cruzamento pode-se destacar (GOLDBERG, 1989):

Cruzamento de 1 ponto Neste método, é escolhido um ponto de corte p qualquer onde $1 < p < w$, sendo w o comprimento do cromossomo. O primeiro cromossomo filho receberá os mesmos genes do pai 1 compreendidos entre o índice 1 e p . Os genes seguintes serão preenchidos com os genes do pai 2 compreendidos entre o índice $p + 1$ e w . O segundo cromossomo filho terá o preenchimento pela forma contrária, ou seja, receberá os mesmos genes do pai 2 compreendidos entre o índice 1 e p . Os genes seguintes serão preenchidos com os genes do pai 1 compreendidos entre o índice $p + 1$ e w . A Figura 5 ilustra um exemplo deste tipo de cruzamento.

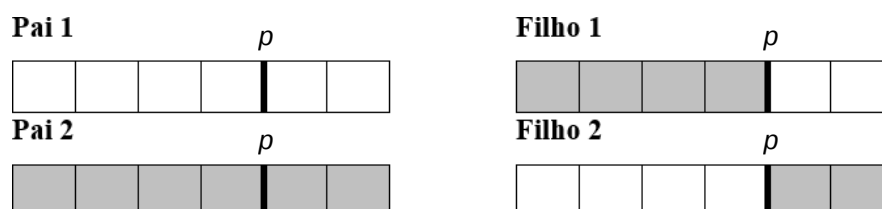


Figura 5 – Cruzamento de 1 ponto.

Cruzamento de 2 pontos No cruzamento de 2 pontos são escolhidos aleatoriamente 2 pontos, p_1 e p_2 , dos cromossomos pais tal que $p_1 < p_2$. O primeiro cromossomo filho terá os mesmos genes do cromossomo do pai 1. Entretanto, a região que compreende p_1 e p_2 terá os genes do pai 2. O segundo cromossomo filho terá os mesmos genes do cromossomo do pai 2. Entretanto, a região que compreende p_1 e p_2 terá os genes do pai 1. A Figura 6 ilustra um exemplo deste tipo de cruzamento.

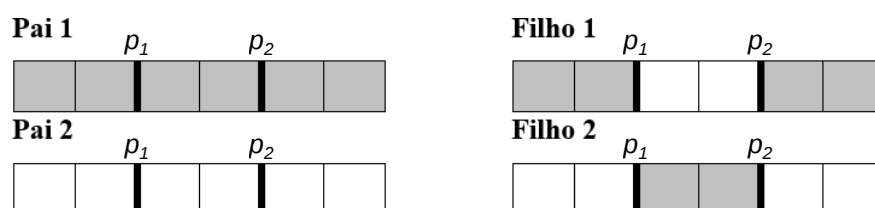


Figura 6 – Cruzamento de 2 pontos.

3.4 Cruzamento Aritmético

O cruzamento aritmético é definido como uma combinação linear de dois vetores. Se $\mathbf{x}_1$ e $\mathbf{x}_2$ são cruzados, os descendentes serão $\mathbf{x}_1^l = a \mathbf{x}_1^l + (1 - a) \mathbf{x}_2^l$ e $\mathbf{x}_2^l = a \mathbf{x}_2^l + (1 - a) \mathbf{x}_1^l$, onde a é um número aleatório no intervalo $[0, 1]$.

3.5 Mutação

A mutação é um processo que garante a exploração de diversas alternativas e é aplicada aleatoriamente sobre os indivíduos da geração atual com uma taxa de probabilidade pré-definida chamada *taxa de mutação*. A mutação nada mais é do que algum tipo de mudança que deve ser realizada sobre um gene. Essa mudança depende de como o gene foi projetado. Em um cromossomo com codificação binária, por exemplo, a mutação altera um gene com valor 1 para valor 0 e vice-versa. Em um cromossomo com codificação real ou inteira, o gene é alterado trocando o valor atual do gene por algum outro valor dentre os possíveis valores válidos para o gene em questão.

Apesar de a aleatoriedade ser uma característica intrínseca aos AG, eles fazem uma busca direcionada. Os novos indivíduos gerados possuem características da população anterior, e essas características são usadas como histórico que direciona a busca para um resultado melhor que o resultado gerado pela população anterior.

No Algoritmo 0.1 são apresentados os passos básicos de um AG em que o critério de parada é dado pelo parâmetro *maxGeracoes*.

Algoritmo 0.1: AlgoritmoGenetico

```
1 Início
2    $t := 1$ ;
3   inicializa( $P_t$ ); //  $P_t$  é a população da geração  $t$ 
4   Enquanto  $t \leq \text{maxGeracoes}$  Faça
5     avalia( $P_t$ );
6      $t := t + 1$ ;
7     seleciona( $t, P_{t-1}$ );
8     aplicaCruzamento( $P_t$ );
9     aplicaMutacao( $P_t$ );
10  Fim-Enquanto
11 Fim
```

4 Algoritmos Genéticos Multiobjetivo

Para resolver problemas de otimização multiobjetivo, pode-se empregar os AG, transformando o problema de otimização multiobjetivo em um problema de otimização mono-objetivo, utilizando algum critério de preferência para encontrar uma única solução. Essa solução deve possuir valores aceitáveis em todos os objetivos segundo o critério de preferências estabelecido, como, por exemplo, priorizar um ou mais objetivos dando pesos a eles. Se mais de uma solução for requerida, é necessário executar repetidamente o algoritmo modificando o critério de preferências.

Por outro lado, pode-se também obter múltiplas soluções em uma execução do AG para resolver problemas multiobjetivo, tentando encontrar as soluções que estejam na fronteira de Pareto considerando a dominância destas soluções.

Os AG que consideram a dominância das soluções utilizando um conjunto de funções objetivo são denominados Algoritmos Genéticos Multiobjetivo (AGMO).

Goldberg (1989) afirma em seu livro que o uso de AGs para a solução de problemas multiobjetivo teve início quando Schaffer (1985) implementou a primeira versão de um AGMO denominado VEGA (*Vector Evaluated Genetic Algorithm*). No algoritmo proposto, considerando uma população de N indivíduos e K objetivos, divide-se esta população em K subpopulações com N/K indivíduos em cada uma delas. O operador de seleção dos AGs é aplicado separadamente para cada uma das subpopulações, isto é, para a subpopulação k considera-se apenas o k -ésimo objetivo para fins da seleção, e, posteriormente, unem-se estas subpopulações e aplica-se os outros operadores genéticos de cruzamento e mutação.

O trabalho de Coello (2006) apresenta uma visão geral da história da otimização multiobjetivo. Este trabalho divide os algoritmos existentes até então. A primeira delas possui algoritmos que possuem como característica a ênfase maior na simplicidade. Entre estes algoritmos destacam-se o VEGA, o *Nondominated Sorting Genetic Algorithm* (NSGA), o *Niched-Pareto*

Genetic Algorithm (NPGA) e o Multi-Objective Genetic Algorithm (MOGA). A segunda geração

dos algoritmos dá maior ênfase à eficiência. Entre os algoritmos classificados nesta geração estão: *Strength Pareto Evolutionary Algorithm* (SPEA), *Strength Pareto Evolutionary Algorithm II* (SPEA2), *Pareto Archived Evolution Strategy* (PAES) e o *Nondominated Sorting Genetic Algorithm II* (NSGA-II). A Tabela 1 mostra um resumo dos principais AGMO.

Tabela 1 – Diferentes tipos de AGMO

Sigla	Nome do Algoritmo	Referência
VEGA	<i>Vector Evaluated Genetic Algorithm</i>	(SCHAFFER, 1985)
WBGA	<i>Weight Based Genetic Algorithm</i>	(HAJELA; LIN, 1992)
MOGA	<i>Multiple Objective Genetic Algorithm</i>	(FONSECA; FLEMING, 1993a)
NSGA	<i>Non-Dominated Sorting Genetic Algorithm</i>	(SRINIVAS; DEB, 1994b)
NPGA	<i>Niched-Pareto Genetic Algorithm</i>	(HORN et al., 1994)
PPES	<i>Predator-Prey Evolution Strategy</i>	(LAUMANNNS; RUDOLPH ; SCHWEFEL, 1998)
REMOEA	<i>Rudolph's Elitist Multi-Objective Evolutionary Algorithm</i>	(RUDOLPH, 2001)
NSGA-II	<i>Elitist Non-Dominated Sorting Genetic Algorithm</i>	(DEB et al., 2002a)
SPGA, SPEA2	<i>Strength Pareto Evolutionary Algorithm 1 e 2</i>	(ZITZLER; THIELE, 1998; ZITZLER; LAUMANNNS; THIELE, 2001)
TGA	<i>Thermodynamical Genetic Algorithm</i>	(KITA et al., 1996)
PAES	<i>Pareto-Archived Evolutionary Strategy</i>	(KNOWLES; CORNE, 1999)
MOMGA-I, MOMGA-II	<i>Multi-Objective Messy Genetic Algorithm</i>	(Van Veldhuizen, 1999)
PESA-I, PESA-II	<i>Pareto Envelope-Base Selection Algorithm</i>	(CORNE; KNOWLES; OATES, 2000; CORNE et al., 2001)

Dentre os diversos métodos citados para se encontrar soluções não-dominadas, neste trabalho o foco será dado ao algoritmo NSGA-II. O interesse neste algoritmo em especial se dá pelo fato de ele ser um dos mais populares e um dos que possuem melhores resultados reportados na literatura quando comparado a outras abordagens (ZITZLER; LAUMANNNS; THIELE, 2001; ZONG-YI et al., 2008; ANTONELLI; DUCANGE; MARCELLONI, 2014; MARTIN et al., 2014).

O funcionamento do NSGA-II é detalhado a seguir.

4.1 Nondominated Sorting Genetic Algorithm II (NSGA-II)

Proposto por (DEB et al., 2002a), este método de otimização foi baseado em seu predecessor NSGA, que, por sua vez, foi implementado a partir de uma ideia de (GOLDBERG, 1989). A ideia central do NSGA é classificar os indivíduos em fronteiras não-dominadas e aplicar um método para diversificar o máximo possível as soluções.

A diferença desta implementação em relação à de um AG mono-objetivo está apenas no modo com que o operador de seleção é empregado. Tanto os operadores de cruzamento quanto os operadores de mutação são idênticos aos utilizados na versão mono-objetivo do AG.

O NSGA-II apresentou soluções para problemas encontrados no NSGA, tais como a alta complexidade do procedimento proposto para a ordenação de não-dominância e a inexistência de elitismo. Para a solução dos problemas citados, o NSGA-II define um novo procedimento para a ordenação das soluções com base no critério de não-dominância e cria um novo conceito, chamado de *crowding distance* (distância de multidão), que se torna responsável por manter a diversidade da população. Também define um método chamado *crowded comparison* (comparação de multidão), que tem como objetivo comparar as soluções geradas.

O NSGA-II trabalha com uma população pai P para gerar uma população filha Q . Na primeira geração, a população inicial P_0 é ordenada por não-dominância. Cada uma das soluções presentes na população P_0 recebe um valor de aptidão conforme o seu nível de não-dominância (1 para o melhor nível, 2 para o seguinte e assim por diante). Após este processo aplicam-se os operadores de seleção, cruzamento e mutação, obtendo-se assim uma população filha Q_0 . Ambas as populações existentes possuem tamanho N .

O pseudocódigo do algoritmo NSGA-II é apresentado no Algoritmo 0.2 com todos os detalhes do processo iterativo.

Algoritmo 0.2: NSGA-II

```
1  Início
2     $P$  : População pai;
3     $Q$ : População filha;
4     $T$  : Tamanho fixo para  $P$  e  $Q$ ;
5     $F_i$ : Conjunto de soluções na fronteira  $i$ ;
6     $n$ : Número da geração atual;
7     $N$  : Número máximo de gerações;
8    Gerar a população inicial  $P_0$  aleatoriamente;
9     $Q_0 = \{\}$  ;
10   Atribuir  $n = 0$ ;
11   Realizar a seleção, o cruzamento e a mutação em  $P_n$  para gerar a população filha  $Q_n$ ;
12   Fazer  $R_n = P_n \cup Q_n$ ;
13   Realizar a ordenação por não dominância em  $R_n$ , gerando  $F_i, i = \{1, \dots, \psi\}$  em  $R_n$ ;
14   Criar  $P_{n+1} = \{\}$  ;
15   Enquanto  $|P_{n+1} \cup F_i| \leq N$  Faça
16     Copiar as soluções de  $F_i$  em  $P_{n+1}$ ;
17      $i = i + 1$ 
18   Fim-Enquanto
19   Calcular a distância de multidão para cada solução em  $F_i$ ;
```

```

20  Ordenar  $F_i$  decrescentemente de acordo com a distância de multidão de cada solução;
21  Copiar as primeiras  $N - |P_{n+1}|$  soluções ordenadas de  $F_i$  para  $P_{n+1}$ ;
22  Se  $n \geq N$  Então
23      Pare;
24  Senão
25      Atribuir  $n = n + 1$  e voltar ao passo 10;
26  Fim-Se
27  Fim

```

No NSGA-II é necessário calcular a distância de multidão para cada solução em uma fronteira F_i . A distância de multidão d de uma solução j , denotada por d_j , representa uma estimativa do perímetro formado pelo cubóide cujos vértices são os seus vizinhos mais próximos. A Figura 7 mostra a distância de multidão para uma solução i .

O cálculo da distância de multidão pode ser visto em detalhes no Algoritmo 0.3.

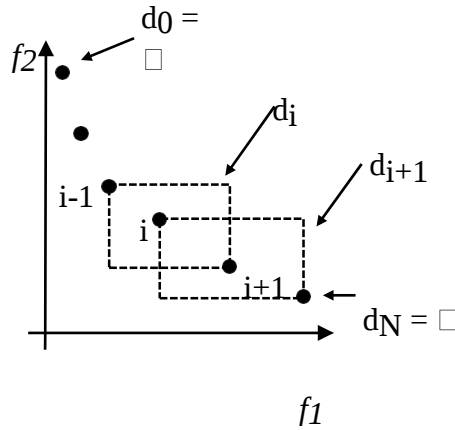


Figura 7 – Cálculo da distância de multidão do NSGA-II em um espaço bidimensional (DEB et al., 2002a)

Algoritmo 0.3: Distância de Multidão

- 1 F_i : conjunto de soluções na fronteira i v_i : número de soluções na fronteira i m : número de funções objetivo
 - 2 Para cada solução j em F_i atribui-se $d_j = 0$ Para cada função objetivo $k = 1, 2, \dots, m$
 - 3 Gerar a lista de soluções I^m igual ao conjunto soluções de F_i ordenadas decrescentemente por f_k
 - 4 Fazer $d_1^m = d_{v_i}^m = \infty$
 - 5 Para as soluções $t = 2, \dots, v_i - 1$

$$d_t^m = d_t^m + \frac{f_k^{t+1} - f_k^{t-1}}{\max_k \min_k (f_k^{t+1} - f_k^{t-1})}$$
-

$$d_t = d_t + \frac{f_k^{t+1} - f_k^{t-1}}{\max_k \min_k (f_k^{t+1} - f_k^{t-1})}$$

onde:

f_k : valor da função objetivo k

I^m : lista ordenada decrescentemente por f_k das soluções em F_i

d_t^m : distância de multidão da t -ésima solução em I^m

(I^m) e (I^m)

f_k^{t+1} e f_k^{t-1} : f_k dos vizinhos da t -ésima solução em I^m

$f_k^{\max}$ e $f_k^{\min}$: máximo e mínimo global para o f_k

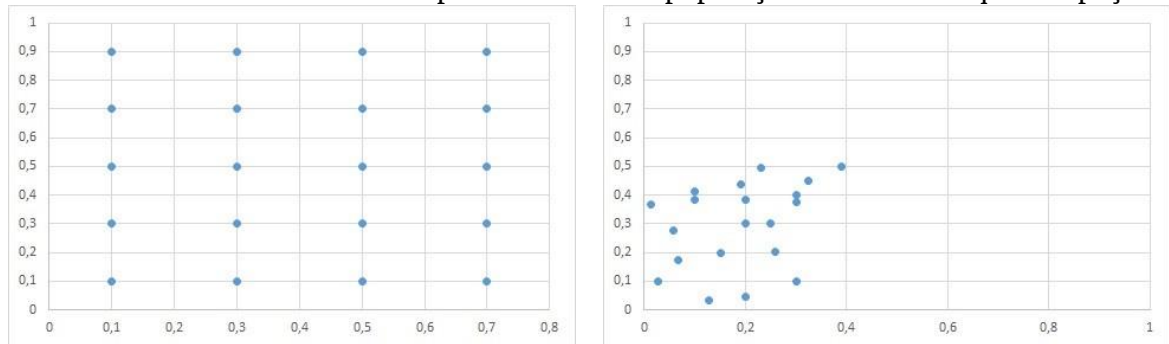
5 Diversidade

Nesta seção descreveremos como a diversidade é tratada no contexto dos AEM e como ela é tratada no contexto dos ecossistemas.

5.1 Diversidade em Algoritmos Evolutivos Multiobjetivo

Nos algoritmos evolutivos, o termo diversidade indica divergências de indivíduos. A Figura 8a é um exemplo de uma população com alto nível de diversidade, enquanto a Figura 8b exemplifica uma população com baixo nível de diversidade.

A diversidade é uma característica importante de uma população, pois sem ela existe o risco de o indivíduo com a maior aptidão dominar a população inteira antes que o espaço de



(a) População com alta diversidade (b) População com baixa diversidade

Figura 8 – População com alta e baixa diversidade.

busca seja explorado adequadamente. Este controle da diversidade em AEM é comumente feito, utilizando a distância de Hamming par-a-par da seguinte forma:

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

onde $dh(i_j, i_k)$ é a distância de Hamming entre dois indivíduos e n é o número de indivíduos na população. Quanto menor a distância, menor é a diversidade.

A diversidade populacional também pode ser computada usando o método de inércia. Ao calcular a inércia de uma *string* binária, cada bit é considerado uma dimensão independente. Neste caso, as coordenadas do centróide $(c_1, c_2, c_3, \dots, c_L)$ dessa *string* de bits P de comprimento L são computadas como:

$$c_i = \frac{\sum_{j=1}^P x_{ij}}{P} \quad (3)$$

e a inércia sobre o centróide é como:

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

onde s_{ij} é o bit na posição i da j -ésima *string* e c_i é a i -ésima coordenada do centróide.

A grande desvantagem destes métodos é o esforço computacional necessário para calcular as medidas par-a-par a cada geração dos AEM. Buscou-se então neste trabalho a inspiração na forma como a diversidade é calculada nos ecossistemas (assunto que será discutido na próxima subseção).

5.2 Diversidade em ecossistemas

A diversidade em ecossistemas é calculada com base em dois índices: riqueza de espécies e equitabilidade.

A *riqueza de espécies* é simplesmente o número total de espécies S em uma unidade amostral. Consequentemente, a riqueza de espécies é muito dependente do tamanho amostral,

pois quanto maior for a amostra, maior será o número de espécies que poderão ser amostradas. Assim, a riqueza de espécies diz pouco a respeito da organização da comunidade, aumentando em função da área, mesmo sem modificação do habitat.

A *equitabilidade* expressa a maneira pela qual o número de indivíduos está distribuído entre as diferentes espécies, isto é, indica se as diferentes espécies possuem abundância (número de indivíduos) semelhantes ou divergentes.

Portanto, a diversidade é uma função do número de espécies e da equitabilidade dos valores de importância da mesma. Por exemplo, podemos observar na Figura 9 que tanto a Amostra A quanto a Amostra B possuem a mesma riqueza de espécies ($S = 4$). Entretanto, a Amostra A possui alta equitabilidade e baixa dominância, enquanto a amostra B, possui alta dominância e baixa equitabilidade. Assim, pode-se concluir que:

1. A equitabilidade é o inverso da dominância;
2. A Amostra A é mais diversa que a Amostra B.

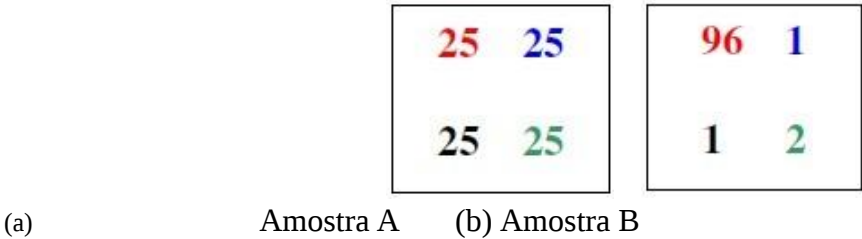


Figura 9 – Riqueza de espécies.

A Tabela 2 elenca os índices mais utilizados na literatura para o cálculo da diversidade em ecossistemas. Para estas equações temos que p_i é a proporção da espécie em relação ao número total de espécimes encontrados nos levantamentos realizados; n_i é o número de indivíduos de cada espécie; N é o número total de indivíduos da amostra; e N_{max} é o número de indivíduos da espécie mais abundante.

Tabela 2 – Índices de diversidade em ecossistemas

Índice	Equação	Referência
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	$I_{me} = \frac{\sqrt{S}}{\sqrt{N}}$	(BANDEIRA et al., 2013)
Macintosh	$I = \frac{\sqrt{S}}{N}$	(JONES et al., 2016)
BergerPaker	$I = \frac{\sqrt{S}}{N}$	(ALRAWASHDEH ¹ ; ALRAWASHDEH, 2016)

$$I_{bp} = \frac{\sum_{i=1}^m \frac{n_i^2}{N}}{N}$$

O índice de Menhinick (I_{me}) é um índice simples de diversidade que considera somente o número de espécies e a raiz quadrada do número total de indivíduos. O índice de Simpson (I_{si}) possui uma vantagem em relação aos índices de Menhinick, pois não somente considera o número de espécies e o total de indivíduos, mas também a proporção do total de ocorrência de cada espécie. O índice de McIntosh (I_{ma}) é um índice simples e mais complexo que o Menhinick, pois

considera número total de indivíduos e a distribuição entre as espécies. O índice de Berger-Parker (I_{bp}) é simples quando comparado com o índice de Simpson, porém eficiente. Ele considera a maior proporção da espécie com maior número de indivíduos. O índice mais usado para medir a diversidade de uma comunidade é o índice de Shannon (I_{sh}), pois incorpora tanto a riqueza quanto a equitabilidade. A equitabilidade é mais comumente expressada pelo Índice de Pielou:

$$J^J = \frac{H^J}{H^J(\text{máximo})} \quad (5)$$

(observado)

sendo que $H^J(\text{máximo})$ é a diversidade máxima possível que pode ser observada se todas as espécies apresentarem igual abundância ao passo que $H^J(\text{observado})$ é diversidade real. O valor de $H^J(\text{máximo})$ pode ser calculado como:

$$H^J(\text{máximo}) = \log S \quad (6)$$

Como pode-se observar, estes índices são relativamente simples e ao mesmo tempo amplamente utilizados pelos estatísticos em ecossistemas. A simplicidade destes índices se dá pelo fato de não necessitarem comparar a posição de um determinado indivíduo com relação ao outro, o que para o contexto da computação traz uma grande eficiência quando comparado aos métodos de cálculo de distância usados nos AEM.

A maior barreira que impede o uso destes índices como forma de medir de diversidade das soluções em AEM é conseguir identificar categorizar as soluções como sendo pertencentes a uma determinada espécie, pois o espaço de busca não é dividido em micro ou macro regiões. Desta forma, faz-se necessário encontrar uma maneira de criar regiões no espaço de busca de tal forma que elas possam ser interpretadas com espécies nestes índices estatísticos. Na próxima seção, propomos uma maneira de dissolver esta barreira.

6 Método Proposto

Para que os índices estatísticos que medem a biodiversidade de ecossistemas possam ser utilizados dentro do contexto dos AEM, foi criado o conceito de *especiação artificial*, em que pseudopartições são criadas no espaço de busca. Cada pseudopartição representa a região (ou *habitat*) onde se concentra uma determinada espécie. Entende-se, portanto, que cada solução localizada na pseudopartição i pertence à espécie i . Desta forma, as pseudopartições exercem o mesmo papel que as espécies exercem no cálculo dos índices da Tabela 2.

A Figura 10 exemplifica como a especiação artificial pode ser aplicada ao espaço de busca. Nesta figura, cada quadrado preto representa uma solução na espaço de busca e cada diagonal (linhas vermelhas) representa a área ocupada por uma determinada espécie. Logo, as soluções que estão na mesma diagonal pertencem à mesma espécie. O tamanho do particionamento (número de linhas diagonais) pode ser ajustado conforme o problema que está sendo abordado.

Uma vez que a especiação artificial faz com que cada solução pertença a uma determinada espécie, o controle da diversidade da população pode ser feito a partir dos índices estatístico de biodiversidade. Desta forma, estes índices podem ser inseridos nos em qualquer classe de Algoritmo Evolutivo.

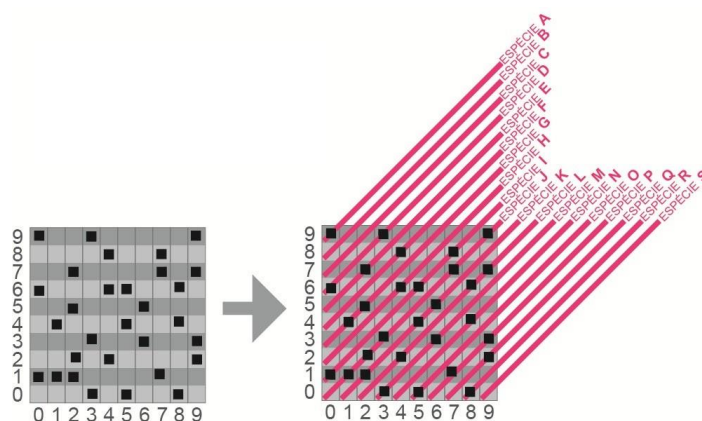
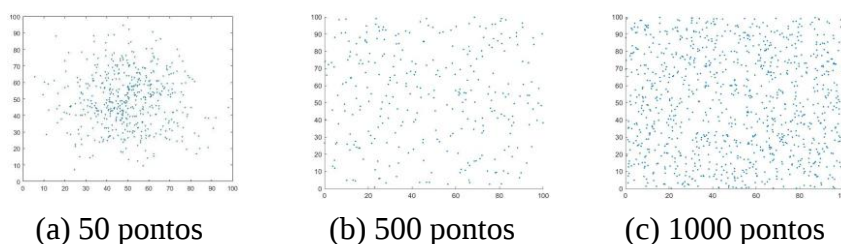


Figura 10 – Exemplo de Especiação Artificial.

7 Experimentos e Resultados

A fim de entender como o uso dos índices estatísticos se comportam com os dados em um espaço de busca em problemas de otimização, foram criados conjuntos de dados artificiais que se distribuem em grupos que podem ser mais densos ou menos densos em relação à distribuição dos pontos.

Destes conjuntos de dados, 11 foram criados com distribuição aleatória, variando somente a concentração dos pontos. A Figura 11 exemplifica 3 destes conjuntos.



(a) 50 pontos

(b) 500 pontos

(c) 1000 pontos

Figura 11 – Distribuição aleatória dos dados.

A Figura 11a tem uma tendência de concentração ao centro. Isso porque em casos reais, os AEM podem convergir para ótimos locais. O objetivo deste conjunto de dados é simular a concentração nestes ótimos locais. A Figura 11b simula situação em que as soluções estariam dispersas. A Figura 11c simula a situação em que as soluções são dispersas, mas há um número maior de soluções.

Foram criados 176 conjuntos de dados que simulam 2 espécies que se estendem principalmente latitudinalmente, 176 conjuntos de dados que simulam 3 espécies que se estendem principalmente latitudinalmente, 176 conjuntos de dados que simulam 4 espécies que se estendem em formato elíptico ao longo de 4 quadrante do espaço de busca e 176 conjuntos de dados que simulam 5 espécies que se estendem em formato elíptico ao longo de 5 quadrante do espaço de busca. As Figuras 12, 13 e 14 exemplificam alguns destes conjuntos de dados.

O padrão dos dados criados segue o padrão mostrado nestas figuras. A diferença entre um conjunto e outro é a quantidade de pontos por grupo, que pode ser 50, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 2000, 3000, 4000, 5000 ou 6000 pontos por grupo. Ao todo foram criados 715 conjuntos de dados artificiais.

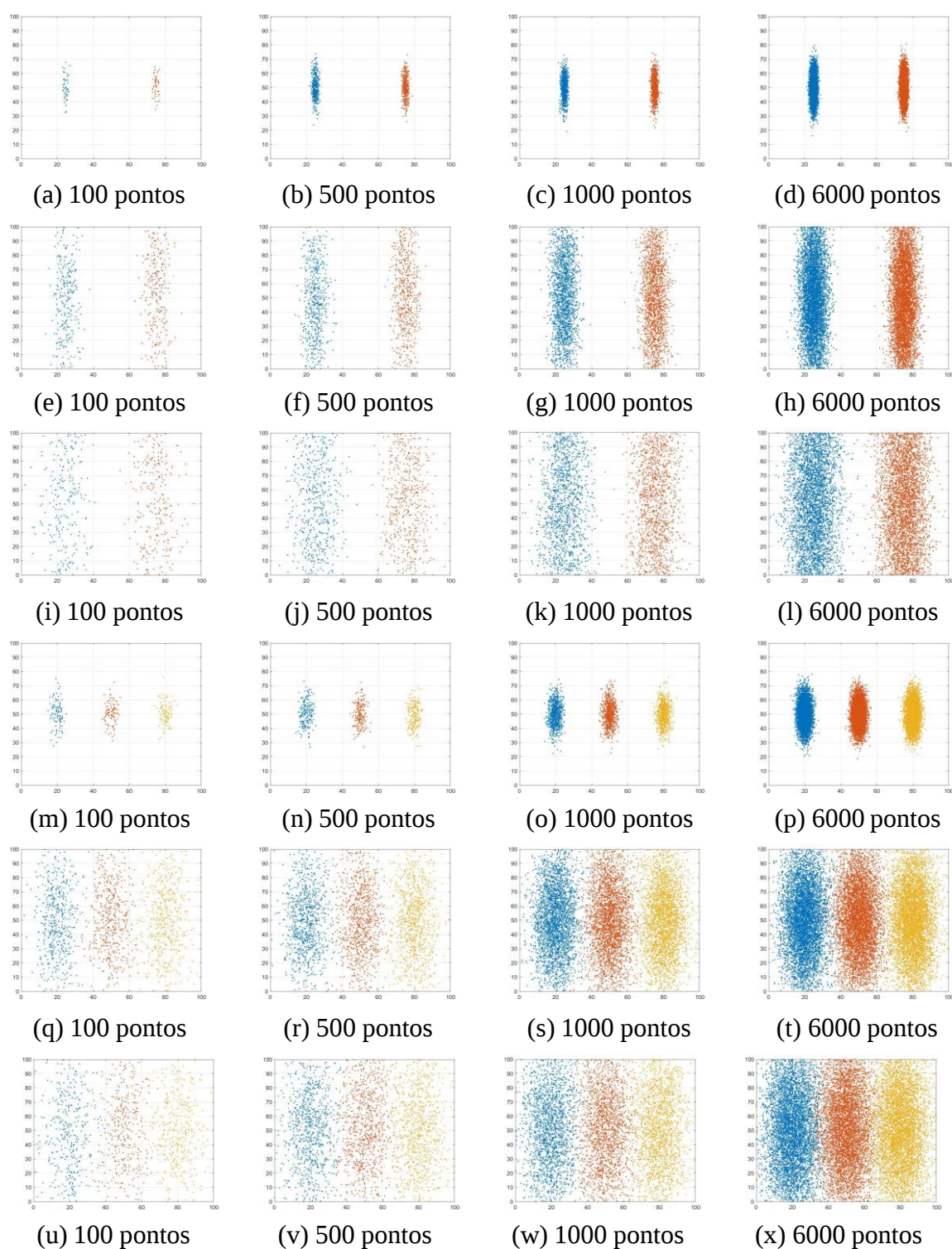


Figura 12 – Distribuição concentrada por 3 faixas latitudinais.

Após a criação dos dados, foi criada a especiação artificial sobre cada um destes 715 conjuntos de dados. A fim de entender qual a melhor forma de particionar os dados em espécies, criamos 4 tipos de particionamento que rotulamos como: particionamento por **faixa**, particionamento **xadrez**, particionamento **quadrado** e particionamento por **raio**.

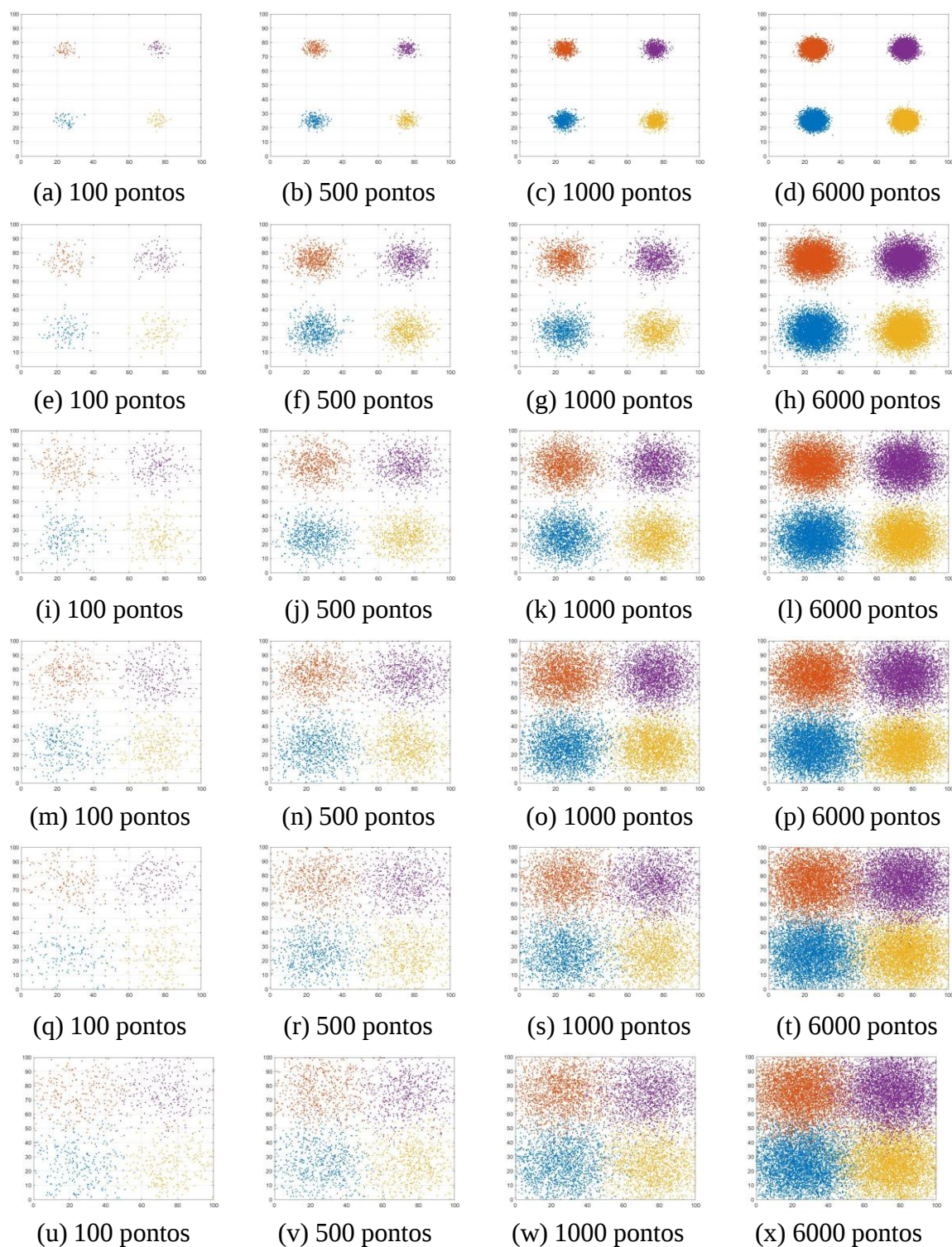


Figura 13 – Distribuição concentrada por 4 áreas elípticas.

A Figura 15 exemplifica estes 4 tipos de particionamento. Nesta figura, foi escolhido arbitrariamente o conjunto de dados 579 com 8 particionamentos (espécies).

Podemos observar na Figura 15 que independentemente de como se particiona o espaço

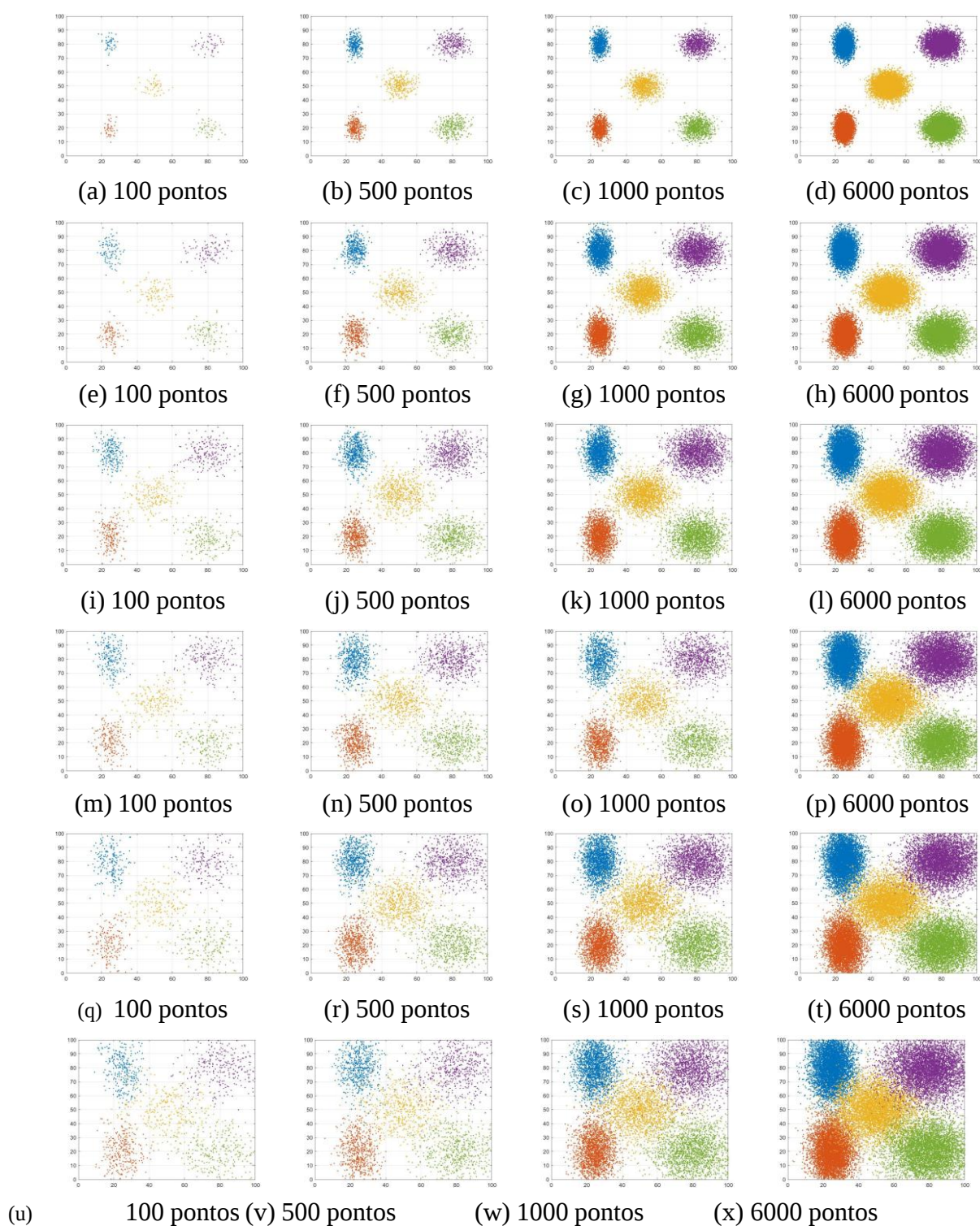


Figura 14 – Distribuição concentrada por 5 áreas elípticas.

de busca para este conjunto de dados é possível identificar que há partições com maior ou com menor concentração de soluções. Isso é desejável, pois os índices conseguirão refletir a diversidade da população a partir deste fenômeno.

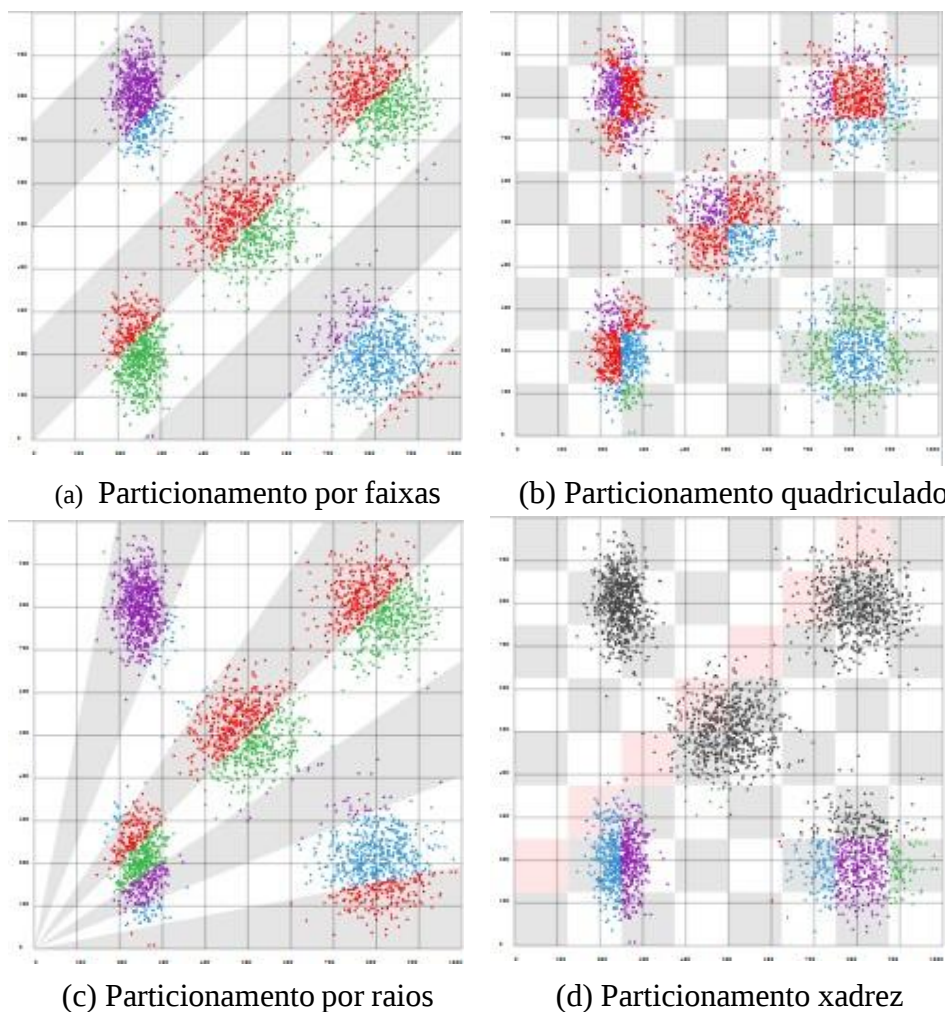
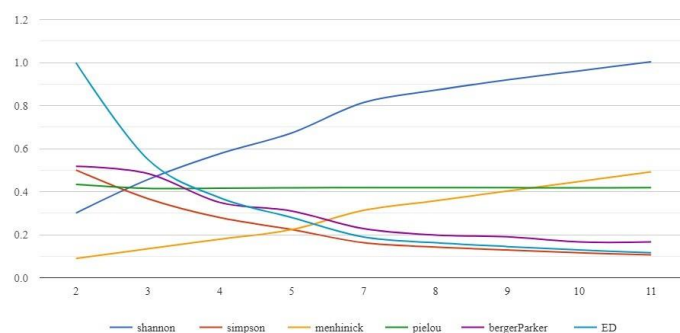


Figura 15 – Particionamentos usados (conjunto de dados artificiais 579).

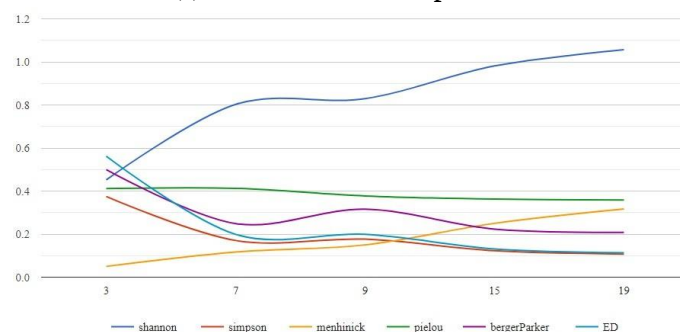
Nos experimentos, usamos os quatro tipos de particionamento para cada um dos 715 conjuntos de dados. Para os particionamentos *quadriculado* e *xadrez* testamos com 2, 4, 5, 8 e 10 partições. Para os particionamentos *raio* e *faixas* testamos com 2, 3, 4, 5, 6, 7, 8, 9, 10 e 11 partições. Desta forma, ao todo foram realizados 21.450 simulações.

A média dos resultados dos índices para cada particionamento pode ser visto na Figura 16. Nesta figura podemos ver que o índice de Shannon foi o que melhor reagiu às alterações da diversidade na população destacando-se dos demais índices. Quanto maior for o número de partições, menor será a área destas partições. Sendo menores, a probabilidade de ter algumas partições com muitas soluções e outras até mesmo vazias é alta. Por conta disso, o índice de Shannon resulta em valores mais elevados para os cenários onde o número de partições é maior.

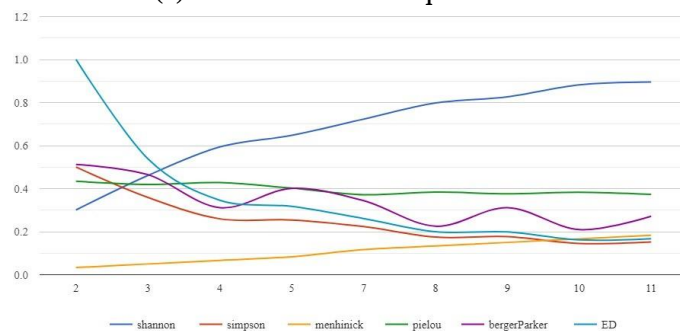
Uma vez que o foco maior dos AEM está em obter soluções diversificadas na fronteira de Pareto, simulamos também o comportamento destes índices estatísticos em dados artificiais que se distribuem como se estivessem sobre uma fronteira de Pareto cujo objetivo é minimizar duas funções. Foram utilizadas as mesmas configurações das simulações anteriores, mas com outros conjuntos de dados. Nestes novos conjuntos de dados n pontos (soluções) foram distribuídos aleatoriamente simulando uma fronteira de Pareto. Cada simulação altera também o número de pontos que são distribuídos ao longo da fronteira de Pareto. Inicia-se com 5 pontos e depois,



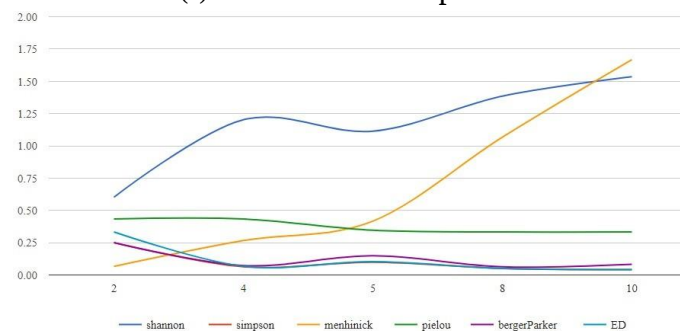
(a) Particionamento por faixas



(b) Particionamento quadriculado



(c) Particionamento por raios



(d) Particionamento xadrez

Figura 16 – Média dos índices para cada particionamento.

a cada nova simulação, são acrescentados mais 10 pontos até que o limite de 305 pontos seja alcançado. A Figura 17 ilustra a distribuição dos pontos ao longo da fronteira de Pareto. Nesta Figura foram usados 35 pontos e 8 partições.

A média dos índices após estas simulações podem ser vistas na Figura 18. Podemos

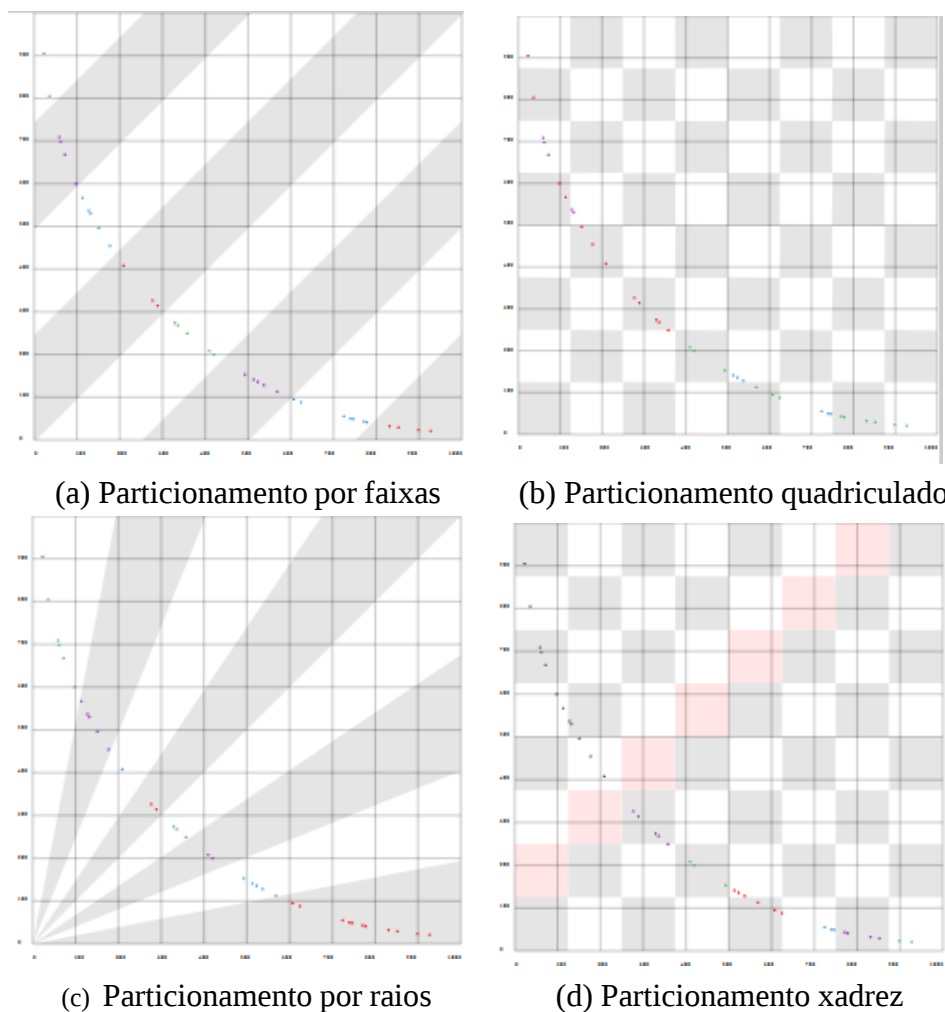


Figura 17 – Particionamentos usados (conjunto de dados artificiais 579).

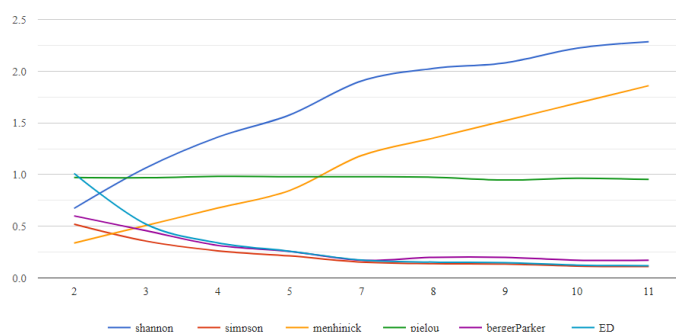
observar nesta figura que o índices de Shannon, com exceção para o particionamento xadrez, também reflete melhor a diversidade na fronteira de Pareto quando comparado aos outros índices. Desta forma, sugerimos adotar o índice Shannon como um índice para monitorar a diversidade da população e da fronteira de Pareto em AEM.

Dentre os 4 tipos de particionamento, sugerimos o uso do particionamento por raios, pois é o que melhor simula o comportamento da convergência da população no contexto dos AEM. Neste caso, os raios devem convergir para a mesma direção que a população converge. Isso vai depender do objetivo das funções (maximização ou minimização).

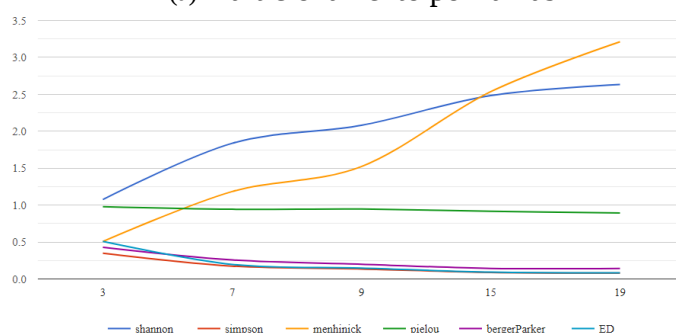
Como sugestão, propomos uma modificação no NSGA-II (0.2) para inserir o índice de Shannon como controle de diversidade da população. Escolhemos o NSGA-II por ser o AEM mais popular da literatura. A Figura 19 mostra o funcionamento do NSGA-II adaptado e a Figura 20 é o algoritmo que mostra como usar o índice de shannon no controle da diversidade.

Considerações finais

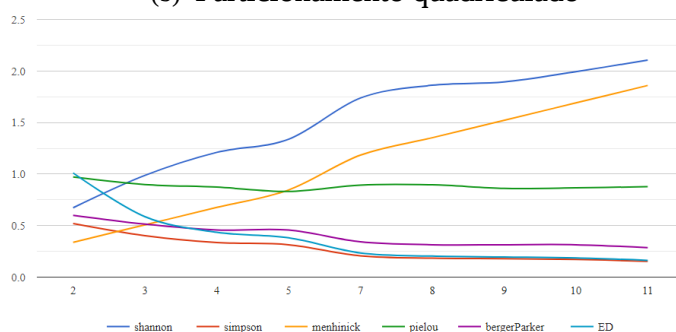
Neste trabalho propomos uma maneira mais eficiente de controlar a diversidade da população dos AEM integrando os índices estatísticos aplicados na ecologia para fins semelhantes.



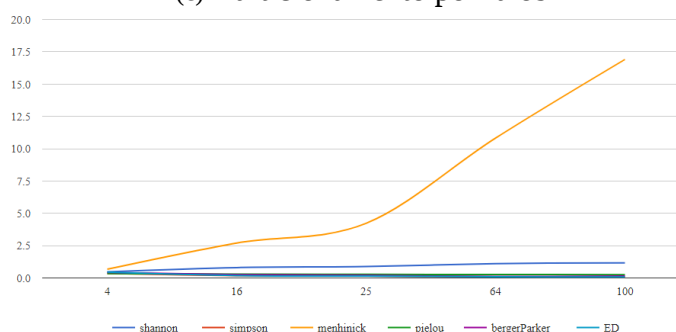
(a) Particionamento por faixas



(b) Particionamento quadriculado



(c) Particionamento por raios



(d) Particionamento xadrez

Figura 18 – Média dos índices para cada particionamento considerando somente a fronteira de Pareto.

O maior problema encontrado para usar estes índices foi a falta de uma técnica de especiação que pudesse separar as soluções por espécie. Foram propostos neste trabalho 4 maneiras de fazer isso particionando o espaço de busca.

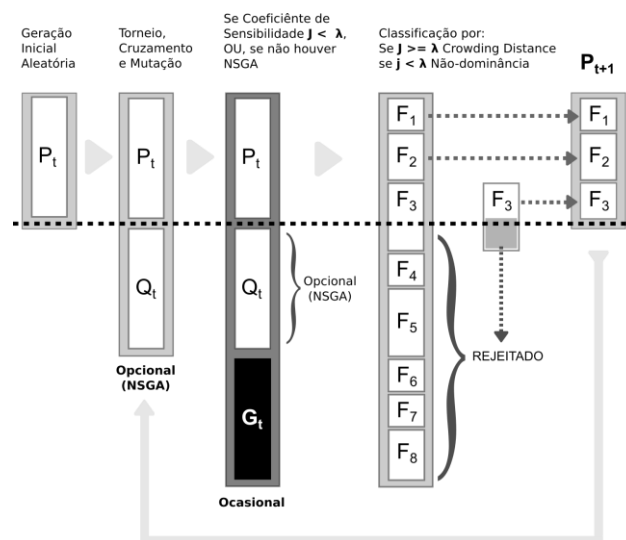


Figura 19 – Funcionamento do NSGA-II modificado.

Os resultados demonstraram que o particionamento por raio é a melhor maneira de inserir estes índices no AEM. O índice que mais se mostrou adequado ao propósito proposto neste trabalho foi o índice de Shannon. Este índice reflete de maneira evidente a diversidade das soluções.

Propomos, ainda, uma alteração no NSGA-II como um exemplo de como inserir índices estatísticos de diversidade em um AEM. Como trabalho futuro queremos aplicar o NSGA-II modificado em problemas reais e comparar a diversidade de suas soluções com algoritmos da literatura.

Referências

- ABRAHAM et al. Evolutionary multiobjective optimization. In: *Evolutionary Multiobjective Optimization*. [S.l.]: Springer, 2005. p. 1–6. Citado na página 2.
- ABRAHAM, A.; JAIN, L.; GOLDBERG, R. *Evolutionary Multiobjective Optimization*. London: Springer-Verlag, 2005. 1–9 p. (Advanced Information and Knowledge Processing). Citado na página 4.
- 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. Citado na página 2.
- ALRAWASHDEH¹, 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. Citado na página 16.
- 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. ISSN 00200255. Citado na página 12.

- BÄCK, T.; SCHWEFEL, H.-P. An Overview of Evolutionary Algorithms for Parameter Optimization. *Evolutionary Computation*, v. 1, n. 1, p. 1–23, mar. 1993. Citado 2 vezes nas páginas 3 e 4.
- BANDEIRA, B. et al. Mathematical convergences of biodiversity indices. *Ecological Indicators*, Elsevier, v. 29, p. 522–528, 2013. Citado na página 16.
- 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. Citado na página 3.
- 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. Citado na página 2.
- CHAMBERS, L. *Genetic algorithms*. [S.l.]: Chapman and Hall/CRC, Boca Raton, 2001. Citado na página 8.
- 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. Citado na página 11.
- 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. Citado na página 12.
- 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. Citado na página 12.
- 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. Citado na página 16.
- De Jong, K. A. *Evolutionary computation: a unified approach*. [S.l.]: MIT press, 2006. Citado 2 vezes nas páginas 3 e 4.
- DEB et al. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. In: SPRINGER. *International conference on parallel problem solving from nature*. [S.l.], 2000. p. 849–858. Citado na página 2.
- 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. Citado 4 vezes nas páginas 5, 6, 12 e 14.
- 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. ISBN 0-7803-7282-4. ISSN 10706631. Citado na página 2.
- 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. Citado na página 4.
- EIBEN, A. E. et al. Parameter control in evolutionary algorithms. In: *Parameter setting in evolutionary algorithms*. [S.l.]: Springer, 2007. p. 19–46. Citado 2 vezes nas páginas 3 e 4.

FONSECA, C.; FLEMING, P. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. *Proceedings of the Conference on Genetic Algorithms*, p. 416–423, 1993. ISSN 14639076. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.48.9077>>. Citado na página 12.

FONSECA, C. M.; FLEMING, P. J. Multiobjective genetic algorithms. In: IET. *IEEE colloquium on genetic algorithms for control systems engineering*. [S.l.], 1993. p. 6–1. Citado na página 2.

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. Citado na página 2.

GOLDBERG, D. E. *Genetic Algorithms in Search, Optimization, and Machine Learning*. [S.l.]: Addison-Wesley, 1989. Addison-We. 412 p. Citado 4 vezes nas páginas 8, 9, 11 e 12.

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. Citado na página 8.

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. Citado na página 3.

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. Citado 2 vezes nas páginas 2 e 12.

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. Citado na página 16.

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. Citado na página 16.

KITA, H. et al. Multi-objective Optimization by Means of the Thermodynamical Genetic Algorithm. *Lecture Notes in Computer Science*, p. 504–512, 1996. Citado na página 12.

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. Citado na página 2.

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. Citado na página 12.

KNOWLES, J. D. *Local-search and hybrid evolutionary algorithms for Pareto optimization*. Tese (Doutorado) — University of Reading Reading, 2002. Citado na página 2.

KOZA, J. Genetic Programming: on the Programming of Computers by Means of Natural Selection. *Notes*, p. 1–187, 1992. Citado na página 3.

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. Citado na página 2.

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. Citado na página 3.

LAUMANNNS 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. Citado na página 2.

LAUMANNNS 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. Citado na página 2.

LAUMANNNS, 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. Citado na página 12.

LI et al. Variable-length pareto optimization via decomposition-based evolutionary multiobjective algorithm. *IEEE Transactions on Evolutionary Computation*, IEEE, 2019. Citado na página 2.

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. Citado na página 2.

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>>. Citado na página 12.

MICHALEWICZ, Z. *Genetic Algorithms + Data Structures = Evolution Programs*. [S.l.]: Springer Science & Business Media, 1996. v. 95. 387 p. Citado na página 8.

MITCHELL, M. *An introduction to genetic algorithms*. [S.l.]: MIT Press, 1998. v. 32. 209 p. Citado 2 vezes nas páginas 3 e 8.

PAQUETE et al. Stochastic local search algorithms for multiobjective combinatorial optimization: A. *Handbook of Approximation Algorithms and Metaheuristics: Methodologies and Traditional Applications*, CRC Press, v. 1, 2018. Citado na página 2.

PEDRYCZ, W.; GOMIDE, F. *An Introduction to Fuzzy Sets: Analysis and Design*. [S.l.]: MIT Press, 1998. (A Bradford book). Citado na página 3.

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. Citado na página 12.

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. Citado 2 vezes nas páginas 11 e 12.

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

SRINIVAS, N.; DEB, K. Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, v. 2, n. 3, p. 221–248, set. 1994. Citado na página 12.

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. Citado na página 2.

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. Citado na página 2.

Van Veldhuizen, D. A. *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*. 272 p. Tese (Doctoral) — Air Force Institute of Technology, 1999. Citado na página 12.

ZITZLER et al. Spea2: Improving the strength pareto evolutionary algorithm. *TIK-report*, Eidgenössische Technische Hochschule Zürich (ETH), Institut für Technische ..., v. 103, 2001. Citado na página 2.

ZITZLER, E.; LAUMANN, M.; THIELE, L. *SPEA2: Improving the Strength Pareto Evolutionary Algorithm*. [S.l.], 2001. 21 p. Citado na página 12.

ZITZLER, E.; THIELE, L. *An Evolutionary Algorithm for Multiobjective Optimization: The Strength Pareto Approach*. [S.l.], 1998. 43 p. Citado na página 12.

ZONG-YI, X. et al. Multi-objective fuzzy modeling using NSGA-II. In: *2008 IEEE International Conference on Cybernetics and Intelligent Systems, CIS 2008*. [S.l.: s.n.], 2008. ISBN 9781424416745. Citado na página 12.

Algorithm 1: Adaptação do NSGA-II

```
Input: int  $tG$ , float  $\lambda$ 
         $\subset \{0..1\}$ , bool  $nsga2$ , bool  $nsgan$ , int  $tR \subset \{5, 8, 11\}$ 
Output:  $P_{(tG-1)}$ 
1  $P_0 = Q_0 = CE_0 = \{\}$ 
2  $P_0 = \text{gerarPopulacaoAleatoria}()$  // Geração Inicial
3  $mR = tPop / tR$ 
   /* Loop para cada geração */
4 for  $t \in \{0..tG\}$  do
   /* Se for com NSGA-II, executa os passos puros do mesmo */
5   if  $nsga2$  then
6      $Q_t = \text{nsga2GerarFilhos}(P_t)$  //  $Q_t$  nos moldes do NSGA-II
7      $R_t = P_t \cup Q_t$ 
8   else
9      $R_t = P_t$ 
   /* Se NSGAN habilitado, calcula o Coeficiente de Sensibilidade (J) */
10  if  $nsgan$  then
11     $J = \text{shannon}(P_t) / \log(tR)$  //  $J = \text{shannon}(P_t) \frac{1}{\log(tR)}$ 
12  else
13     $J = 0$ 
   /* Executa o algoritmo fast-non-dominated-sort para identificar os
      Frontes como NSGA-II */
14  if  $((nsgan) \text{ AND } (J \leq \lambda)) \text{ OR } (not\ nsga2)$  then
   /* => se NSGAN habilitado e sensibilizado ( $J \leq \lambda$ ) */
   /* Executa SEM crowding-distance (apenas dominância comum) */
15     $F = \text{fast-non-dominated-sort-Modificado}(R_t)$ 
   /* Verifica se NSGA-II está desabilitado e por isso, NSGAN precisa
      preencher tudo sozinho */
16     $puro = ((not\ nsga2) \text{ AND } (\text{length}(P_{t+1}) \leq \text{length}(P_0)))$ 
   /* 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     $F = \text{fast-non-dominated-sort}(R_t)$ 
   /* 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  $(\text{length}(P_{t+1}) + \text{length}(F_i)) \leq tPop$  do
   /* Calcula o crowding-distance */
23      if  $(nsgan) \text{ AND } (J \leq \lambda)$  then
24        crowding-distance-assignment-Modificado( $F_i$ )
25      else
26        crowding-distance-assignment( $F_i$ )
27       $P_{t+1} = P_{t+1} \cup F_i$  // 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     $P_{t+1} = P_{t+1} \cup F_i[0:tPop - \text{length}(P_{t+1})]$ 
31 return  $P_{tG-1}$ 
32
```

Figura 20 – Funcionamento do NSGA-II modificado.
