Learning and Nonlinear Models - Journal of the Brazilian Society on Computational Intelligence (SBIC), Vol. 19, Iss. 2, pp. 51-68, 2021

Uma Introdução a Metodologias Fuzzy para Otimização Multiobjetivo em Ambiente Incerto

Ricardo Coêlho
Departamento de Estatística e Matemática Aplicada
Centro de Ciências, Universidade Federal do Ceará
rcoelhos@dema.ufc.br

Resumo – Este artigo traz uma introdução aos conceitos de otimização multiobjetivo em ambientes incertos com o auxílio do paradigma de Programação Matemática Fuzzy. O objetivo é apresentar metodologias baseadas em Programação Matemática Fuzzy para resolver problemas de Programação Multiobjetivo, que modelam problemas de otimização com várias funções objetivo e, em muitos casos, conflitantes. O artigo também apresenta um amplo levantamento da bibliografia clássica dessa área, como por exemplo os Algoritmos Evolutivos para Otimização Multiobjetivo (AEOM), que apresentam um crescente desenvolvimento de algoritmos e aplicações em problemas práticos desde o início deste século. Ademais, uma das etapas da modelagem de problemas práticos está no tratamento da incerteza presente em dados reais e aqui apresentamos algumas propostas que usam a Teoria de Conjuntos *Fuzzy* para esse tratamento.

Palavras-chave – Programação Multiobjetivo, Programação Matemática *Fuzzy*, Teoria dos Conjuntos *Fuzzy*, Algoritmos Evolutivos, Computação Flexível.

Abstract – This paper provides an introduction about multiobjective optimization concepts in uncertain environment with the aid of the Fuzzy Mathematical Programming paradigm. The goal is to present methodologies based on Fuzzy Mathematical Programming to solve Multiobjective Programming problems, which model optimization problems with several objective functions and, in many cases, conflicting ones. The article also presents an extensive survey of the classic bibliography in this area, such as the Evolutionary Algorithms for Multi-Objective Optimization (MOEAs), which presents a growing development of algorithms and applications in practical problems since the beginning of this century. Besides, one of the steps in modelling practical problems is to deal with the uncertainty in the real-world data and here we show some proposals that use the Fuzzy Sets Theory for this treatment.

Keywords – Multiobjective Programming, Fuzzy Mathematical Programming, Fuzzy Sets Theory, Evolutionary Algorithms, Soft Computing.

1. INTRODUÇÃO

As decisões tomadas diariamente são responsáveis pela curva de aprendizagem na construção das experiências e dos conhecimentos no decorrer de nossa vida, de acordo com os conceitos que nos foram transmitidos. As características da curva de aprendizado é um tema que norteia alguns grupos de pesquisadores na tentativa de entendê-las e reproduzi-las. De acordo com algumas pesquisas, a curva de aprendizado pode ser modelada matematicamente, isto é, pode ser convertido em um conjunto de regras formuladas de maneira matemática e, portanto, observar se um computador pode também tomar decisões conflitantes e de difícil solução em questão de pouco tempo.

Algo em comum em um número considerável de problemas práticos a serem resolvidos é a presença de dados imprecisos ou vagos, que tornam alguns desses problemas intratáveis. Há algumas técnicas para tratar esse tipo de informação, que estão diretamente conectadas ao método de resolução a ser escolhido. Muito já foi discutido sobre esse assunto, ao ponto de fazerem parte do conjunto de temas que pertencem ao mundo da ficção científica, do uso de robôs que façam os serviços mais pesados e domésticos, dos equipamentos que funcionam com o nosso pensamento entre muitas outras que facilitariam a nossa vida.

O primeiro passo na tentativa de entender esse tipo de problema foi discutido na metade do século passado quando foi criada a linha de pesquisa chamada Inteligência Artificial (IA). Contudo, algumas subáreas de IA não obtiveram um avanço esperado e esses problemas ocorreram porque a complexidade de seu tratamento é muito elevada e também porque os seres humanos têm uma capacidade de formar uma ampla variedade de tarefas físicas e mentais sem qualquer medida ou cálculo. Procurando compreender o fraco desenvolvimento de algumas áreas da IA, alguns pesquisadores começaram a discutir os motivos, e atribuem esse lento desenvolvimento ao fato de que alguns objetos são tratados com valores rígidos, enquanto que eles surgem de percepções que são diferentes dependendo do ponto de vista que é analisado. Em [84] é possível identificar alguns problemas na definição de IA original e aponta uma nova direção em IA.

No final do século passado, uma coleção de metodologias baseadas em uma abordagem mais flexível foi criada. Essas metodologias podem ser usadas separadamente ou mescladas de distintas maneiras, no intuito que elas cooperem entre si na obtenção de uma solução satisfatória. Esse grupo heterogêneo de metodologias foi primeiramente definido em [82, 83] e foi chamado de Computação Flexível, que agrega uma coleção de metodologias que ajudam a tratar tolerância por imprecisões e incertezas para lograr tratabilidade, robustez e baixo custo de solução. Suas principais componentes são a lógica nebulosa, raciocínio probabilístico, neurocomputação e metaheurísticas. A computação flexível está sendo usada em uma gama vasta de aplicações e a cada dia tem aumentado a sua importância. O principal modelo de regra que a computação flexível usa é o pensamento humano. A Figura 1 descreve os elementos que fazem parte da computação flexível e como ela é organizada.

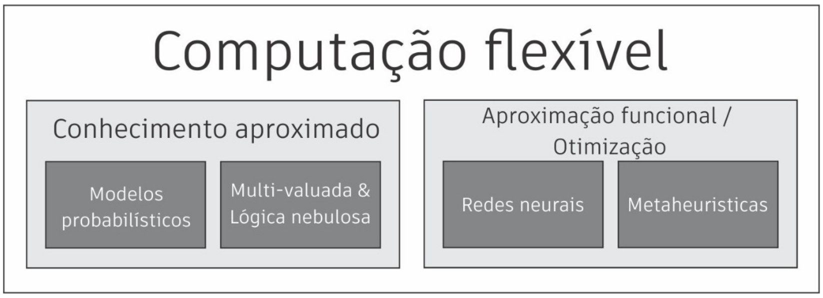

Figura 1: Estrutura dos elementos da Computação Flexível.

Dentre as metodologias pertencentes ao grupo de Raciocínio Aproximado, pode-se destacar a Lógica *Fuzzy* (ou Lógica Nebulosa), que é responsável por tratar dados imprecisos e/ou vagos presentes nos problemas práticos. As técnicas de lógica nebulosa têm sido utilizadas em diferentes campos de conhecimento e em diversas aplicações: aproximação de funções, previsão de séries temporais, filtragem de sinais, identificação e controle de processos, otimização e pesquisa operacional, dentre outros. Um número significativo de implementações práticas vem consolidando a lógica nebulosa, por meio de sistemas *fuzzy*, não só em instalações industriais, como em muitos produtos manufaturados.

A Pesquisa Operacional (PO) é um ramo da ciência que tem por finalidade desenvolver técnicas para otimizar o desempenho de sistemas e auxiliar na tomada de decisão. Suas aplicações encontram-se nas áreas industriais, de negócios, militares, governamentais, entre outras. O estudo da PO pode ser dividido em algumas subáreas, sendo a Programação Matemática (PM) uma delas. Ela tem como meta solucionar problemas que envolvem minimização (ou maximização) de uma ou várias funções objetivo, nos quais podem ser do tipo irrestrito ou restrito. Um problema de programação matemática com vários objetivos pertence a um conjunto que é resolvido por otimização multiobjetivo e esses objetivos são muitas vezes conflitantes e frequentes na vida real. Ao contrário de um problema mono-objetivo, os problemas multiobjetivos não possuem uma solução ótima única. Uma solução adequada deve obter um desempenho adequado para todos os objetivos. Em geral, os métodos designados para esses problemas são bastante específicos, de forma que as estruturas desses algoritmos dificilmente podem ser generalizadas para resolver outros problemas.

Este trabalho está dividido da seguinte forma: na Seção 2 é apresentada uma breve descrição dos conceitos básicos, formulação matemática e alguns métodos para resolver problemas de otimização multiobjetivo; na Seção 3 é apresentada uma visão geral sobre uma seleção de algoritmos evolutivos desenvolvidos para resolver problemas de otimização multiobjetivo; na Seção 4 são descritos os conceitos de lógica nebulosa, assim como os de otimização multiobjetivo com incertezas; e na seção 5 são apresentadas as conclusões desse trabalho.

2. UMA BREVE DESCRIÇÃO SOBRE PROGRAMAÇÃO MULTIOBJETIVO

Como é bem conhecido, Programação Matemática (PM) representa uma área de pesquisa que visa encontrar soluções eficientes para problemas da vida real. Esses problemas são formulados matematicamente de maneira clara e precisa. Eles necessitam ter a(s) função(ões) objetivo(s) a ser(em) otimizada(s), e eles podem ser do tipo irrestrito ou restrito. PM tem várias classes de problemas que podem ser encontrados em teoria de jogos, alocação de facilidades, problemas de logística, designação de tarefas, problemas de economia em geral, controle, processamento de sinais, entre outros. Várias aplicações e métodos podem ser encontrados em [7, 30, 37, 66, 79].

Os problemas de PM tratados neste trabalho têm duas ou mais funções objetivo, que são chamadas de problemas de programação multiobjetivo. Assim, primeiramente será mostrada a formulação matemática de um problema de programação multiobjetivo irrestrito

$$\begin{array}{ll} \min & F(\mathbf{x}) \\ \text{s.a} & \mathbf{x} \in \mathbb{R}^n \end{array} \tag{1}$$

sendo que $F = (f_1, f_2, \ldots, f_m),\ (m \geq 2)$ é um vetor de objetivos.

Diante da dificuldade em obter uma solução ótima única, o conceito de uma solução ótima eficiente ou não-dominada foi introduzida em [55] e [56]. Alguns métodos específicos para resolver problemas de programação multiobjetivo são propostos em [10]. A grande maioria destes métodos são descritos por modelos de otimização que usam frequentemente PM clássica, a qual tenta desenvolver um modelo exato para resolver os problemas de otimização em geral.

No caso do Problema (1) ter todas as suas funções objetivo com característica linear, o problema a ser otimizado é chamado de problema de programação multiobjetivo linear. Caso contrário, ele é chamado de problema de programação multiobjetivo não-linear. Entretanto, existem outras formas de identificarmos ou separarmos os problemas de otimização, como por exemplo os de Programação Convexa (PC), que é um conjunto de problemas de programação matemática que engloba todos os problemas lineares e alguns casos particulares de não-lineares.

Escolher o objetivo para ser otimizado é um passo crítico no processo de modelagem de um problema do mundo real e a solução ótima a ser obtida depende totalmente dessa escolha. Em um subgrupo de problemas práticos, vários objetivos podem ser definidos e muitas vezes eles são conflitantes e/ou não-mensuráveis. No intuito de solucionar esse tipo de problema é que surgiu a otimização multiobjetivo, que é um ramo da área de otimização matemática. A otimização multiobjetivo foi desenvolvida para resolver problemas de programação matemática com várias funções objetivo. Devido a esse ponto de conflito, o conceito clássico de otimalidade não pode ser aplicado, pois agora será encontrado um conjunto de soluções ótimas que otimizam esse tipo de problema.

Os trabalhos clássicos de Vilfredo Pareto, vide [55, 56], introduziu o conceito de Pareto-otimalidade e iniciou o campo de otimização multiobjetivo. Depois desses trabalhos, a solução de um Problema de Otimizaçao Multi-objetivo (POM) é caracterizada por um conjunto de pontos chamados não-dominados ou eficientes. Atualmente, o enfoque multiobjetivo encontra aplicações em qualquer área que o processo de tomada de decisão esteja centrado em problemas de otimização.

Métodos específicos para resolver problemas de programação multiobjetivo foram propostos em [10, 29]. Esses métodos são classificados de acordo com o instante que o decisor aplica os seus critérios. Três métodos são propostos: (i) Métodos a-Priori, onde o decisor atribui esses critérios antes de realizar a execução de qualquer método; (ii) Métodos a-Posteriori, onde o decisor permite que os métodos encontrem um conjunto de soluções eficientes para depois ele escolher uma solução diante desses critérios; e (iii) Métodos interativos, onde esses critérios são aplicados no decorrer da execução dos mesmos com a tentativa de guiar a uma solução eficiente.

2.1 Conceitos básicos

Alguns conceitos envolvidos em otimização são importantes de serem apresentados.

- **Problema geral multiobjetivo**

  O problema de otimização multiobjetivo foi definido por [54] como sendo o problema que visa encontrar um vetor de variáveis de decisão que satisfaz as restrições e otimiza uma função vetor cujos elementos representam as funções objetivo. Estas funções formam uma descrição matemática dos critérios de performance que normalmente são conflitantes entre eles. O termo otimizar significa encontrar uma solução tal que forneça os valores de todas as funções objetivo aceitáveis pelo designer.

  Formalmente podemos dizer que um problema multiobjetivo minimiza $F(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_k(\mathbf{x}))$ sujeito a m restrições de desigualdade $g_i(\mathbf{x}) \leq 0,\ i = 1, \ldots, m$ e p restrições de igualdade $h_i(\mathbf{x}) = 0,\ i = 1, \ldots, p,\ \mathbf{x} \in \Omega$. Uma solução do problema multiobjetivo minimiza as componentes do vetor $(\mathbf{x} = x_1, \ldots, x_n)$ de algum universo Ω.

- **Dominância de Pareto**

  O espaço de busca Ω é parcialmente ordenado no sentido de que duas soluções arbitrárias são relacionadas de duas possíveis maneiras: ou uma domina a outra ou nenhuma delas domina. Este conceito pode ser formalizado da seguinte forma. Sejam $\mathbf{a}$ e $\mathbf{b} \in \Omega$, dizemos que $\mathbf{a}$ *domina* $\mathbf{b}$ se e somente se:

  $$\forall i \in \{1, 2, ..., k\}, f_i(\mathbf{a}) \geq f_i(\mathbf{b}) \quad \text{e} \quad \exists j \in \{1, 2, ..., k\}, f_j(\mathbf{a}) > f_j(\mathbf{b}). \tag{2}$$

  Em outras palavras, "**a** não é pior que **b** em nenhum dos objetivos e é melhor em pelo menos um". Normalmente exitem um conjunto de soluções Pareto-ótimas conhecidas como soluções não-dominadas. A figura 2 representa o conjunto dessas soluções.

- **Fronteira de Pareto**

  No sentido de Pareto está no limite da região de design ou no local dos pontos tangentes às funções objetivo. A região dos pontos definido por esse limite é conhecida como fronteira de Pareto, ou seja, os valores de $\mathcal{F}(\mathbf{x})$ cujo $\mathbf{x}$ está no conjunto Pareto-ótimo formam a fronteira de Pareto.

2.2 Métodos para resolver de problemas otimização multiobjetivo

Os métodos para resolução de POMs podem ser classificados de acordo com o momento em que o decisor aplica seus critérios.

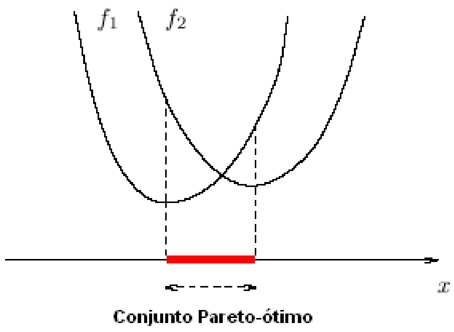

Figura 2: Conjunto Pareto-ótimo

2.2.1 Métodos A Priori

O decisor define preferências antes da solução do POM. Desta forma, o problema multiobjetivo original é transformado em problemas mais simples que podem ser resolvidos através de métodos conhecidos.

- **Métodos lexicográficos:** o decisor deve listar os objetivos em ordem decrescente de importância. A primeira função da lista é maximizada no espaço de busca e este resultado é considerado como uma restrição na maximização da segunda função; e assim por diante. Generalizando temos:

$$\text{Maximizar} \quad f_{ij}(\mathbf{x}) \tag{3}$$
$$\text{sujeito a:} \quad \mathbf{x} \in \Omega_j \tag{4}$$

  onde $\Omega_j = \{x : x \in \Omega_{j-1}, x = \arg\max(f_{ij-1})\}$. $\{ij\} = \{i_1, i_2, ..., i_k\}$, para $i = 1, 2, ..., k$, é o conjunto dos objetivos i segundo a ordem de importância j. Este método é inapropriado quando existe um grande número de objetivos, até porque o decisor deve ter informações sobre as funções antes da resolução do problema.

- **Métodos baseados em limitantes:** o decisor deve selecionar um objetivo de preferência e indicar limitantes para os demais objetivos. Formalmente temos:

$$\text{Maximizar} \quad f_j(\mathbf{x}), \ \text{com} \ l_i \leq f_i(\mathbf{x}) \leq L_i \tag{5}$$
$$\text{sujeito a:} \quad \mathbf{x} \in \Omega \ \text{com} \ i = 1, 2, ..., k \qquad i \neq j \tag{6}$$

2.2.2 Métodos A Posteriori

Nos métodos à posteriori se busca gerar o conjunto solução para depois escolher uma solução de compromisso. Em outras palavras, o método fornece como resultado um conjunto de soluções. A partir desse conjunto de soluções, o decisor poderá escolher a solução que melhor se adeque aos recursos disponíveis.

- **Métodos das ponderações:** obtem uma solução Pareto-ótima resolvendo um problema formulado a partir da soma ponderada de todas as funções objetivo do POM original. O problema ponderado pode ser generalizado da seguinte maneira:

$$\text{Maximizar} \quad \sum w_i . f_i(\mathbf{x}) \tag{7}$$
$$\text{sujeito a:} \quad \mathbf{x} \in \Omega \tag{8}$$

  com $\sum w_i = 1$, $w_i \geq 0$, $i = 1, 2, ..., k$. Os pesos w_i representam a relativa importância de cada função objetivo no problema.

- **Método das ϵ-restrições:** é baseado na maximização de uma função objetivo (a mais importante) e transformando as outras funções objetivo em restrições de desigualdade. O problema ϵ-restrito pode ser descrito como:

$$\text{Maximizar} \quad f_j(\mathbf{x}) \ \text{com} \ f_i(\mathbf{x}) \leq \epsilon_i \tag{9}$$
$$\text{sujeito a:} \quad \mathbf{x} \in \Omega \ \text{para} \ i = 1, 2, ..., k \qquad i \neq j \tag{10}$$

- **Método simplex multiobjetivo:** este método é restrito para problemas lineares na forma:

$$\text{Maximizar} \quad C\mathbf{x} \tag{11}$$
$$\text{sujeito a:} \quad A\mathbf{x} \leq b \quad x \geq 0 \tag{12}$$

  com $\mathrm{C} \in \mathbb{R}^{k \times n}, x \in \mathbb{R}^n, b \in \mathbb{R}^m, A \in \mathbb{R}^{m \times n}$.

2.2.3 Métodos interativos

Nos métodos interativos, informações sobre preferências são passadas para o decisor durante a resolução do POM.

- **STEM - STEP Method:** procedimento interativo direcionado à resolução de POM lineares:

$$\text{Maximizar} \quad [C_1\mathbf{x}, C_2\mathbf{x}, ..., C_k\mathbf{x}] \tag{13}$$
$$\text{sujeito a:} \quad A\mathbf{x} \leq b \quad x \geq 0 \tag{14}$$

  com $C_i \in \mathbb{R}^{1\times n}, x \in \mathbb{R}^n, i = 1, 2, ..., k$. O decisor deve classificar cada solução encontrada com satisfatória ou insatisfatória quando comparadas com uma solução utópica.

- **Método de Geoffrion, Dyer e Feinberg:** baseia-se no algoritmo de Frank-Wolfe e é destinado para POM na forma:

$$\text{Maximizar} \quad U[f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_k(\mathbf{x})] \tag{15}$$
$$\text{sujeito a:} \quad \mathbf{x} \in \Omega \tag{16}$$

  onde U é uma função côncava crescente que reflete as preferências gerais de um decisor em relação aos k objetivos. Como U não é totalmente conhecida algumas informações devem ser fornecidas pelo decisor durante o processo.

Atualmente, existem várias técnicas de programação para resolução de POM. Entretanto, estas técnicas tendem a gerar somente um elemento do Conjunto Pareto-ótimo a cada rodada. Além disso, elas são sensíveis ao formato da Fronteira de Pareto.

Por um lado, muitos problemas práticos não podem ser adequadamente representados de maneira linear devido à natureza das funções que são não-lineares. Por outro lado, os problemas de PM necessitam ser formulados de forma clara e precisa, mas na grande maioria dos casos seus dados são incertos, imprecisos ou mal-definidos. Existem algumas maneiras de conseguir descrever matematicamente essas incertezas, como programação estocástica, lógica nebulosa, caos ou até mesmo com uma aproximação numéricas destes dados imprecisos. Nas próximas seções serão apresentados alguns métodos baseados no conceito estocástico para obter a fronteira de Pareto e como os dados incertos podem ser tratos usando lógica nebulosa.

3 ALGORITMOS EVOLUTIVOS PARA OTIMIZAÇÃO MULTIOBJETIVO

O termo algoritmo evolutivo (AE) se refere a uma classe de métodos de otimização estocástica que simulam o processo da evolução natural. Sua origem ocorreu no final da década de 1960 e desde os anos 70 muitas metodologias evolutivas vêm sendo propostas, principalmente por algoritmos genéticos, programação genética e estratégias evolutivas.

Dentre as várias áreas emergentes nas quais os Algoritmos Evolutivos (AEs) [5] se destacaram, a otimização multiobjetivo é uma das que apresentou maior crescimento nos últimos anos [16]. Um problema de otimização multiobjetivo difere de um problema de otimização com objetivo único por possuir vários objetivos, muitas vezes conflitantes, que necessitam ser otimizados. Ao contrário de um problema mono-objetivo, os problemas multiobjetivos não possuem uma solução ótima única. Uma solução adequada deve obter um desempenho adequado para todos os objetivos [52].

O potencial dos algoritmos evolutivos para resolver problemas de otimização mutiobjetivos foi sugerido nos anos 60 por [58]. Porém a primeira implementação de um algoritmo evolutivo multiobjetivo só foi desenvolvido em meados dos anos 80 [64, 65]. A partir daí, uma quantidade considerável de pesquisas vêm sendo feitas nessa área [20]. A crescente importância desse campo é refletida pelo aumento significante dos artigos técnicos em conferências internacionais. Usando o pacote Bibliometrix no R [3], foi possível realizar uma compilação das buscas feitas nas bases de dados SCOPUS e Web of Science com as palavras chave: (multiobjective OR multi-objective) AND (programming OR optimization) AND evolutionary AND algorithm. O resultado obtido pode ser observado na Figura 3.

A maior motivação para usar algoritmos evolutivos (principalmente os algoritmos genéticos, baseados em [34,35,38]) para resolver problemas de otimização multiobjetivos é devido ao fatos dos AEs lidarem simultaneamente com um conjunto de possíveis soluções (população) que nos permite encontrar vários membros do conjunto Pareto-ótimo em apenas uma execução do algoritmo ao invés de ter que realizar uma série de execuções separadas como no caso das técnicas de programação matemática tradicionais [52]. Além disso os AEs são menos susceptíveis à forma ou continuidade da fronteira de Pareto, já que esses são problemas conhecidos para as técnicas de programação matemática [11, 12, 21, 90].

Vários problemas de otimização admitem múltiplas (e muitas vezes, conflitantes) funções objetivo. Um Problema de Otimização Multiobjetivo (POM) pode ser definido como o problema de encontrar um vetor de variáveis de decisão que satisfaça as restrições e otimize a função vetorial cujos elementos representam as funções objetivo. Nestes problemas, otimizar significa encontrar todos os valores aceitáveis para as funções objetivos para uma tomada de decisão.

Os Algoritmos Evolutivos (AE), em especial os Algoritmos Genéticos (AG), têm demonstrado bom desempenho na resolução de POMs e nos últimos anos várias abordagens foram apresentadas. A maioria das propostas de Algoritmos Evolutivos para Otimização Multiobjetivos (AEOM) considera mecanismos de preservação de diversidade, *fitness*

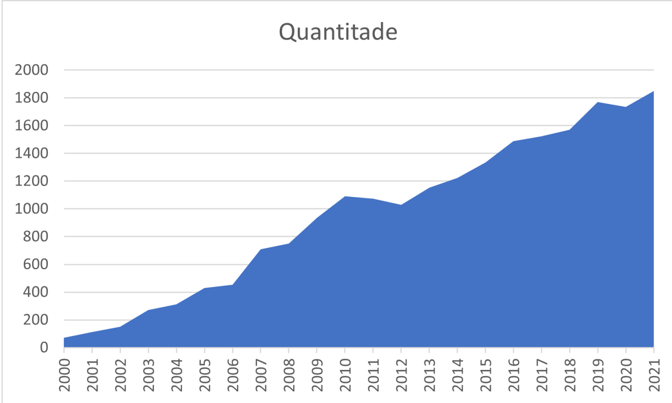

Figura 3: Quantidade de artigos publicados com Algoritmos Evolutivos para Otimização Multiobjetivo de 2000 até 2021.

sharing, restrições de *mating* e *crowding*. Estes mecanismos são empregadas com o intuito de resolver o problema da convergência prematura e garantir também a uniformidade das amostras da fronteira de Pareto.

O tamanho da população influencia fortemente na capacidade de convergência do AEOM através da fronteira de Pareto, pois populações pequenas não provem diversidade suficiente entre os indivíduos.

O conceito de elitismo também é incorporado e têm garantido melhor desempenho aos algoritmos. Em algumas das abordagens estudadas, é comum manter um conjunto externo de soluções não-dominadas entre todos os indivíduos gerados até a população corrente. Uma desvantagem na utilização deste conceito é quando informações de preferências são incluídas na atribuição de *fitness* e essas preferências mudam com o tempo.

A avaliação da função de *fitness* e o critério se seleção são especialmente importantes na busca pelo Conjunto Pareto-ótimo. Três enfoques são normalmente abordados: seleção por critério (alternação dos objetivos durante a seleção), seleção por agregação (múltiplos objetivos são combinados em um só) e seleção baseada em Pareto-dominância.

Finalmente, no decorrer do processo de otimização multiobjetivo é comum que soluções não dominadas em um determinado estágio tornem-se dominadas por soluções superiores em um estágio posterior. Logo, um monitoramento da população envolvida deve ser realizado: quão boa ou superior é uma geração em comparação à anterior.

Na comparação entre as abordagens apresentadas alguns *benchmarks* foram propostos por diversos pesquisadores. Os conceitos gerais de POM serão apresentados, juntamente com os métodos de resolução. As diferentes abordagens de AEOMs serão discutidas e várias aplicações serão apresentadas.

3.1 Visão Geral de Algoritmos Evolutivos para Otimização Multiobjetivo

Considerando que a solução de um POM é constituída por um conjunto de pontos, a utilização de AEs parece imediata, pois considera simultaneamente uma população de soluções. Isto permite encontrar vários pontos do conjunto Pareto-ótimo em uma única rodada do algoritmo (paralelismo), ao contrário das técnicas tradicionais de programação matemática, que necessitam de uma série de rodadas separadas. Adicionalmente, os AEs são menos susceptíveis à forma ou à continuidade da fronteira de Pareto. Os Algoritmos Evolutivos para Otimização Multiobjetivo (AEOM) são classificados como métodos à posteriori, visando a geração de pontos uniformemente distribuídos que representem o Conjunto Pareto-ótimo.

O campo de pesquisa em AEOM ficou praticamente inativo durante anos. Começou a crescer em meados dos anos 90, quando várias técnicas e aplicações foram desenvolvidas. Na figura 2 apresentamos o número de publicações sobre AEOM por ano.

3.2 Classificação dos Algoritmos Evolutivos para Otimização Multiobjetivo

As diferentes técnicas de AEOM podem ser classificadas em:

O VEGA (Vector Evaluated Genetic Algorithm) foi o primeiro trabalho em algoritmos evolutivos para otimização multiobjetivo proposto por Schaffer [64].

O MOGA (Multi-Objective Genetic Algorithm), foi proposto por [31, 32]. O *rank* de um indivíduo é igual ao número de indivíduos da população corrente pelos quais ele é dominado e, todos os indivuos não dominados têm o mesmo *rank*. Utiliza *fitness sharing* e *mating*.

O NPGA (Niched-Pareto Genetic Algorithm) foi proposto por [39, 40]. Utiliza seleção por torneio baseado na Pareto-dominância e *fitness sharing*. NPGA2 foi proposto in [26] e utiliza Pareto ranking, mas matem seleção por torneio e *fitness sharing* do NPGA.

O NSGA (Nondominated Sorting Genetic Algorithm) foi proposto por [73]. Baseado em fronteiras de classificação dos indivuos; aos indivíduos não-dominados são atribuidos valores de *fitness dummy*, e eles são removidos da população;

o processo se repete até toda a população estar classificada. Utiliza *fitness sharing*. O NSGA 2, proposto em [22], é computacionalmente mais eficiente do que o NSGA, além de utilizar elitismo e comparação por *crowding*.

O PAES (Pareto Archived Evolution Strategy) foi proposto por [14]. Utiliza a estratégia evolutiva (1+1), basicamente uma busca local baseada em mutação e Pareto-dominância, além de uma memória externa com todos os indivíduos não dominados encontrados anteriormente.

O PESA (Pareto Envelope-based Selection Algorithm) foi proposto por [15]. Utiliza uma pequena população interna e uma grande população externa. O mecanismo de seleção é baseado em uma medida de *crowding*, que também é utilizada para decidir quais indivíduos entrarão na população externa (guarda os indivíduos não dominados encontrados durante o processo evolutivo). O PESA2 é a versão revisada do PESA, na qual uma seleção baseada em regiões é adotada [14].

O MOMGA (Multi-Objective Messy Genetic Algorithm) foi proposto in [75] com o intuito de estender os algoritmos genéticos *messy* para a resolução de problemas de otimização multiobjetivo [76]. O MONGA2 em [93] pode ser visto como uma versão revisada do MOMGA.

O SPEA (Strength Pareto Evolutionary Algorithm) foi proposto em [92]. Utiliza seleção por torneio e *fitness sharing*. Nichos não são definidos em termos de distância mas em Pareto-dominância. Indivíduos não dominados são mantidos em uma população externa. SPEA2 [91] é a versão revisada do SPEA, com melhora na associação de *fitness* e incorporação da informação de densidade da vizinhança. O Micro-GA proposto em [13] é similar ao PAES.

Algumas destas abordagens são discutidas mais detalhadamente em [11]. Observamos que os algoritmos SPEA 2 e NSGA II representam as técnicas mais promissoras para a resolução de algoritmos evolutivos para otimização multiobjetivo.

4 PROGRAMAÇÃO MULTI-OBJETIVO FUZZY

A lógica nebulosa e a teoria dos conjuntos *fuzzy* foram desenvolvidas por L. A. Zadeh [80], na tentativa de refletir matematicamente a imprecisão do mundo real . Hoje em dia, a lógica nebulosa, ou melhor, *Soft Computing* (SC) é empregada com grande sucesso na concepção, construção, formulação e utilização de uma ampla gama de produtos e sistemas em que o funcionamento é diretamente baseado na forma de raciocínio do ser humano. Um problema de programação multiobjetivo *fuzzy* irrestrito pode ser formulado como

$$\begin{array}{ll} \widetilde{\min} & F(\tilde{\mathbf{c}};\tilde{\mathbf{x}}) \\ \text{s.a} & \tilde{\mathbf{x}} \in \widetilde{\Omega} \end{array} \tag{17}$$

sendo que $F = (f_1, f_2, \ldots, f_m)(m \geq 2)$ um vetor de objetivos, $\tilde{\mathbf{c}} \in \mathbb{F}(\mathbb{R}^{m\times p})$ representa os parâmetros *fuzzy* nas funções objetivo e $\widetilde{\Omega} = \mathbb{F}(\mathbb{R}^n)$ é o conjunto de solução factíveis. $\mathbb{F}(\mathbb{R})$ define o conjunto de números *fuzzy*, $\mathbb{F}(\mathbb{R}^n)$ define o conjunto de vetores n-dimensionais com parâmetros *fuzzy* e $\mathbb{F}(\mathbb{R}^{m\times p})$ define o conjunto de matrizes $(m \times n)$-dimensionais com parâmetros *fuzzy*. Dentre os parâmetros incertos que foram descritos no Problema (17), podemos observar que as imprecisões podem estar presentes em vários pontos do problema.

Existem casos de problemas do mundo real em que os seus parâmetros são raramente conhecidos com exatidão e necessitam ser estimados. Assim, a aplicação de métodos que consigam descrever matematicamente essas imprecisões é necessária, como descrito em [2, 49, 53, 77]. Outras técnicas de resolução de problemas de otimização *fuzzy* podem ser encontradas em [4, 23, 42, 67–69, 72, 74, 86]

Nessa seção são apresentadas algumas abordagens propostas para resolver problemas de programação multiobjetivo restritos em um ambiente *fuzzy*. Os parâmetros *fuzzy* podem estar presentes nas funções objetivo, como também nos coeficientes, variáveis e/ou relação de ordem no conjunto de restrições. Um problema de programação multiobjetivo *fuzzy* restrito pode ser formulado da seguinte maneira:

$$\begin{array}{ll} \widetilde{\min} & F(\tilde{\mathbf{c}};\tilde{\mathbf{x}}) \\ \text{s.a} & G(\tilde{\mathbf{a}};\tilde{\mathbf{x}}) \leq^f \tilde{\mathbf{b}} \\ & \tilde{\mathbf{x}} \in \tilde{\Omega} \end{array} \tag{18}$$

onde $F = (f_1, f_2, \ldots, f_m)$ $(m \geq 2)$ é um vetor de objetivos, $G = (g_1, \ldots, g_l)$ é um vetor de funções que restringem a região factível e $\leq^f$ representa uma relação de ordem *fuzzy* que compara números *fuzzy*. $\tilde{\mathbf{c}}_i \in \mathbb{F}(\mathbb{R}^{p_i})$ (para $i = 1, \ldots, m$) representa a quantidade de custos *fuzzy* em cada função objetivo i, $\tilde{\mathbf{a}}_j \in \mathbb{F}(\mathbb{R}^{o_j})$ (para $j = 1, \ldots, l$) representa a quantidade de coeficientes *fuzzy* em cada função de restrição j, e $\tilde{\mathbf{b}} \in \mathbb{F}(\mathbb{R}^l)$ representam os parâmetros *fuzzy* do termo independente, enquanto $\tilde{\mathbf{x}}$ representa as variáveis de decisão *fuzzy*, sendo $\tilde{\Omega} \in \mathbb{F}(\mathbb{R}^n)$ o conjunto de soluções factíveis *fuzzy*. Neste trabalho, toda imprecisão é representada por funções de pertinência definidas pelo decisor.

A incerteza pode estar presente em diferentes pontos no conjunto de restrições ou simplesmente não existir imprecisões. Assim, é possível dividir em duas abordagens principais. A primeira abordagem é um conjunto de restrições sem imprecisão, isto é, as restrições são clássicas e usam os critérios tradicionais para garantir a exequibilidade das soluções. No momento que surgem as imprecisões, essas podem ser de dois tipos. O primeiro é definido quando a incerteza está presente na relação de ordem do conjunto de restrições e o segundo quando os coeficientes de um dos lados da relação de ordem ou de ambos os lados são incertos.

Em [8] é definida a **decisão *fuzzy*** como a intersecção das várias metas, e é formalizada como segue:

Definição 4.1 *Assuma que as funções de pertinências que descrevem o objetivo* fuzzy μ_G *e a restrição* fuzzy μ_C *são dados em um espaço de alternativas* X*. Então,* μ_G *e* μ_C *são combinados para formar a* ***decisão* fuzzy***,* μ_D*, que é um conjunto* fuzzy *resultante da intersecção de* μ_G *e* μ_C*:*

$$\sup_{\mathbf{x}\in X} \mu_D(\mathbf{x}) = \sup_{\mathbf{x}\in X} [\mu_G(\mathbf{x}) \wedge \mu_C(\mathbf{x})]$$

que é basicamente uma escolha ou um conjunto de escolhas de objetivos e restrições das alternativas viáveis no conjunto fuzzy X*.*

Note que na Definição 4.1 os objetivos e as restrições *fuzzy* são inseridos na expressão que define uma outra função de pertinência, que denominamos por μ_D. Assim, é possível encontrar uma decisão máxima para um problema extremo por uma função escalar, como descrito em [53]. Seja $\phi[0,1] \to [0,1]$, onde $\phi(\alpha) = \sup_{\mathbf{x}\in\mathbf{X}(\alpha)} \mu_G(\mathbf{x})$, com $X(\alpha) = \{\mathbf{x} \in \mathbb{R}^n \mid \mu_X(\mathbf{x}) \geq \alpha\}$. Se ϕ é contínuo em [0,1] então tem um **ponto fixo** $\overline{\alpha}$ e, portanto,

$$\sup_{\mathbf{x}\in X} \mu_D(\mathbf{x}) = \sup_{\mathbf{x}\in X(\overline{\alpha})} \mu_G(\mathbf{x}) = \overline{\alpha}.$$

A partir do que foi exposto acima, a investigação sobre programação multiobjetivo *fuzzy* restrita pode definir o conjunto Pareto ótimo *fuzzy*, assim como as condições de otimalidade de KKT para soluções não-dominadas *fuzzy* são estabelecidas. Os enfoques mostrados nessa seção são baseados nos trabalhos [17–19, 68–72]

4.1 Levantamento bibliográfico

Usando novamente o pacote Bibliometrix no *R* [3], foi possível realizar uma compilação das buscas feitas nas bases de dados SCOPUS e Web of Science com as palavras chave: (multiobjective OR multi-objective) AND (programming OR optimization) AND evolutionary AND algorithm AND fuzzy. O resultado obtido pode ser observado na Figura 4.1.

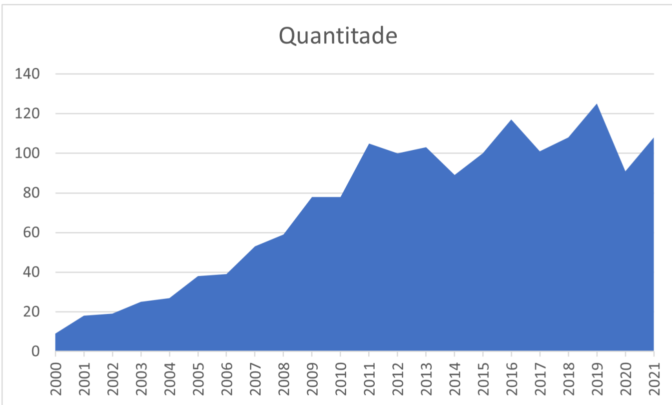

Figura 4: Quantidade de artigos publicados com Algoritmos Evolutivos para Otimização Multiobjetivo em Ambiente Fuzzy de 2000 até 2021.

O assunto escolhido para o presente artigo foi o de programação matemática multiobjetivo em um ambiente *fuzzy*, onde as imprecisões podem estar presentes nas funções objetivo e/ou nas funções que geram o conjunto restrição.

Os métodos de otimização usam frequentemente programação matemática clássica, que tenta desenvolver um modelo exato para o problema que estamos interessados em otimizar. Alguns destes métodos podem ser encontrados em [6, 10, 33, 36, 51]. Contudo, a modelagem desses problemas podem conter ambiguidades em seus dados, o que frequentemente ocorre em problemas do mundo real. Em recentes anos, a teoria de números *fuzzy* [80] mostrou grande potencial para lidar com sistemas que são não-lineares, complexos, mal-definidos e não bem entendidos. Ela encontrou numerosas aplicações devido à sua fácil implementação, flexibilidade e tolerância natural aos dados imprecisos. Essas qualidades podem ser úteis para representar comportamentos de complexidade arbitrária porque a teoria de números *fuzzy* está baseada em termos de linguagem natural. Em [85] é discutido o uso de teoria de conjuntos *fuzzy* que é uma teoria precisa de argumentos imprecisos e aproximados. Algumas aplicações podem ser encontradas em [24, 45, 89], e estão inseridas nos campos de reconhecimento de padrões, análise de dados, controle, economia, pesquisa operacional, entre outros.

As representações e manipulações aritméticas das quantidades numéricas incertas podem ser feitas usando a teoria de números *fuzzy*. Entretanto, a comparação entre dois ou mais números, intervalos e conjuntos *fuzzy* não é fácil. Existem na literatura vários enfoques que foram desenvolvidos para conseguir compará-los (veja alguns exemplos em [9, 24, 44, 45, 57]), sendo que cada um é baseado em um ponto de vista diferente ou leva em consideração possíveis necessidades específicas do problema a ser resolvido. Diante desta variedade de opções, os autores aqui destacam a teoria de possibilidade, que é análoga à teoria de probabilidade e foi proposta por Zadeh [81] para agregar o conceito de uma distribuição de possibilidade à teoria de conjuntos *fuzzy*. Alguns índices de comparação para ordenar números e intervalos *fuzzy* empregam a teoria de possibilidade e eles foram propostos em [25]. Um outro ponto de vista é descrito em alguns trabalhos que transformam um problema de programação *fuzzy* em um ou vários problemas clássicos e, nesse momento, pode-se usar métodos clássicos para encontrar a melhor solução ou um conjunto ótimo de soluções. Um exemplo desse procedimento é descrito em [43], que transforma um problema de otimização *fuzzy* com um único objetivo em um problema multiobjetivo clássico e o número de funções objetivo é definido pela quantidade de coeficientes imprecisos presentes no problema *fuzzy* original.

4.2 Abordagens desenvolvidas para resolver problemas multiobjetivo em ambiente *fuzzy*

Várias noções de Pareto-otimalidade *fuzzy* foram descritas baseadas em diferentes métodos de comparação entre números com certa imprecisão. Estes métodos são de grande importância, pois são usados para ordenar as soluções candidatas e obter o conjunto de soluções não-dominadas. As definições de Pareto para problemas *fuzzy* podem ser divididas de três maneiras: (i) o uso de funções de pertinência para descrever as incertezas inseridas nas funções objetivo; (ii) coeficientes *fuzzy* como parâmetros somente nas funções objetivo; (iii) coeficientes *fuzzy* como parâmetros nas funções objetivo e no conjunto de restrições.

4.2.1 Função de pertinência das funções objetivo

O trabalho pioneiro em usar funções de pertinência para representar as funções objetivos e funções de restrição foi desenvolvido por Bellmann e Zadeh [8] e foi usado como referência para muitos outros trabalhos. Contudo, outros pesquisadores descrevem maneiras diferentes de usar as funções de pertinência para obter um conjunto de soluções eficientes do problema multiobjetivo a ser otimizado, como descrito no decorrer dessa subseção.

Enfoque de Bellmann a Zadeh: Em [8], encontramos o primeiro desenvolvimento na direção de resolver problemas de programação multiobjetivo com dados imprecisos. O objetivo principal está em representar as incertezas por números *fuzzy* usando funções de pertinência para permitir um certo tipo de flexibilização dos dados imprecisos, tanto quando o objetivo é incerto como também quando a formação do conjunto de restrição é *fuzzy*. Em um segundo momento é realizado uma agregação de todas essas funções de pertinência no intuito de encontrar um valor que melhor satisfaça todas as funções de pertinência ao mesmo tempo, que é chamado de nível de satisfação. Alguns trabalhos adotam essa abordagem para resolver problemas de programação multiobjetivo linear, dentre os quais se destacam [78, 87, 88].

Esse nível de satisfação pode ser dado por uma escolha ou por um conjunto de escolhas realizadas pelo decisor. É possível destacar que essa decisão pode ser definida por um conjunto *fuzzy* de alternativas, resultantes da intersecção dos objetivo e restrições. Dessa maneira, esse conceito é definido da seguinte forma:

Definição 4.2 *Assuma que são dados um objetivo* fuzzy G *e uma restrição* fuzzy C*, que estão em um espaço de alternativas* X*. Então,* G *e* C *são combinados para formar a decisão* D*, a qual é um conjunto* fuzzy *resultante da intersecção de* G *e* C*, isto é,*

$$D = G \bigcap C.$$

Pode ser observado que nesse caso foram considerados para um único objetivo e uma única restrição *fuzzy*, mas essa definição pode ser generalizada para múltiplos objetivos e restrições. Assim, se existem m objetivos $G_1, \ldots, G_m$ e p restrições $C_1, \ldots, C_p$ sobre um espaço de alternativas X, então a decisão resultante é a intersecção dos objetivos e restrições *fuzzy* que foram dados, isto é,

$$D = G_1 \bigcap \ldots \bigcap G_m \bigcap C_1 \bigcap \ldots \bigcap C_p.$$

Note que na equação acima os objetivos e restrições *fuzzy* são inseridos da mesma maneira, de uma forma mais genérica, como foi descrito na Definição 4.2 na obtenção da decisão *fuzzy*. Essa é a ideia básica, que indentifica uma regra para tratar os objetivos e as restrições na obtenção de um processo de decisão em um ambiente *fuzzy*.

Enfoque de Farina e Amato: Em [27] e [28], uma sequência de definições de otimalidade é proposta, que pode ser dividida em quatro partes. Essa sequência começa com a definição de Pareto-otimalidade clássica e depois eles apresentam três definições envolvendo conjunto de números *fuzzy*. Cada uma delas é uma extensão da definição clássica. Usando exemplos, os autores mostram que a definição de Pareto-otimalidade pode ser insatisfatória devido essencialmente a três razões:

1. Número de valores objetivos melhorados e iguais não é levado em conta;
2. Magnitudes das melhoras não são levadas em conta;

3. Nenhuma preferência entre os objetivos é considerada.

A sequência segue com uma definição de k-otimalidade, que introduz três funções para cada par $\mathbf{v}_1, \mathbf{v}_2 \in \Omega$. A primeira função, n_b, determina o número de funções objetivo para as quais $\mathbf{v}_1$ domina $\mathbf{v}_2$; a segunda, n_e, retorna o número de funções objetivo para as quais as duas soluções candidatas são equivalentes; a última função, n_w, é o contador de funções objetivo para as quais $\mathbf{v}_1$ é dominado por $\mathbf{v}_2$. Seja M o número de funções objetivo. Para $k \in [0, 1]$, pode-se definir o conceito de $(1-k)$-dominância e k-otimalidade como segue

Definição 4.3 ((1-k)-dominância) $\mathbf{v}_1$ *é dito ser* $(1-k)$*-dominado por* $\mathbf{v}_2$ *se*

$$\begin{cases} n_e < M \\ n_b \geq \frac{M-n_e}{k+1} \end{cases} \tag{19}$$

Definição 4.4 (k-otimalidade) $\mathbf{v}^*$ *é k-ótimo se não existe nenhum $v \in \Omega$ tal que $\mathbf{v}$ k-domina $\mathbf{v}^*$.*

O terceiro elemento na sequência é a definição de k-otimalidade *fuzzy*, que é uma definição estendida da k-otimalidade com as três funções n_b, n_e e n_w substituídas por funções de pertinência n_b^F, n_e^F, n_w^F. Os conceitos de $(1-k_F)$-dominância e (k_F)-otimalidade também são extensões e são redefinições dos conceitos anteriores. Essa sequência termina com uma definição de otimalidade *fuzzy* mais geral. Uma definição *fuzzy* da relação de dominância muda ela mesma as funções n_b^F, n_e^F, n_w^F. Essa definição é dada abaixo.

Definição 4.5 (Dominância *fuzzy*) *Seja $\mu_D(\mathbf{v}_1, \mathbf{v}_2)$ uma função de pertinência definida como segue.*

$$\mu_D(\mathbf{v}_1, \mathbf{v}_2) \triangleq f_{\mu_D}(n_b^F(\mathbf{v}_1, \mathbf{v}_2), n_e^F(\mathbf{v}_1, \mathbf{v}_2), n_w^F(\mathbf{v}_1, \mathbf{v}_2)) \tag{20}$$

onde f_{μ_D} pode ser uma função de pertinência ou um sistema fuzzy.

Definição 4.6 (Otimalidade *fuzzy*) *Uma função de pertinência μ_O representa a relação da otimalidade* fuzzy *se $\mathbf{v}^*$ pertence ao conjunto definido pelo corte k_F em μ_O para qualquer $k_F \in [0, 1]$ e somente se não existe nenhum $\mathbf{v} \in \Omega$ tal que*

$$\mu_D(\mathbf{v}, \mathbf{v}^*) > k_F \tag{21}$$

Enfoque de Köppen e Nickolay: Uma abordagem baseada na inserção de imprecisão, o que chamaremos de "fuzzificação" da relação de dominância de Pareto é proposta em [46] e [47]. Essa nova abordagem determina valores que ajudam na ordenação de um conjunto de vetores usando graus de imprecisão de dominância, o que resulta na definição abaixo:

Definição 4.7 *É dito que o vetor $\vec{a}$ domina o vetor $\vec{b}$ com grau μ_a de acordo com a divisão*

$$\mu_a(\vec{a}, \vec{b}) = \frac{\prod_i \min(a_i, b_i)}{\prod_i a_i} \tag{22}$$

e que o vetor $\vec{a}$ é dominado pelo vetor $\vec{b}$ com o grau μ_p de acordo com a divisão

$$\mu_p(\vec{a}, \vec{b}) = \frac{\prod_i \min(a_i, b_i)}{\prod_i b_i} \tag{23}$$

Observe que as definições não são simétricas, porque elas diferem no denominador. Esses graus de dominância são usados para ordenar um determinado conjunto de vetores de dimensão M, que representam o espaço das funções objetivo. O valor desses vetores pode ser o valor de adaptação de um problema de otimização multiobjetivo. Os M elementos são ordenados em uma ordem crescente dos valores calculados pela definição acima pela seguinte equação:

$$r_M(\vec{a}) = \max_{\vec{b} \in M \setminus \{\vec{a}\}} \mu_p(\vec{a}, \vec{b}) \tag{24}$$

Note que essa definição está relacionada com um conjunto. Um valor de ordenação do vetor $\vec{a}$ de dimensão M pode somente ser designado com referência aos M dados contidos em $\vec{a}$.

Enfoque de Sakawa usando função de pertinência: Outro enfoque bastante conhecido para tratar modelos de otimização imprecisos está descrito em [59], e assume meta *fuzzy* para cada função objetivo de um problema de programação multiobjetivo, as quais são descritas pelo decisor. Assim, é possível deixar mais flexível os requisitos rígidos dos problemas de programação multiobjetivo para a minimização das m funções objetivo diante de um conjunto de restrições. Logo, o problema multiobjetivo flexibilizado pode ser escrito da seguinte forma:

$$\begin{array}{ll} \widetilde{\min} & f(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})) \\ \text{s.a} & \mathbf{x} \in \Omega = \{\mathbf{x} \in \mathbb{R}^n | g_j(\mathbf{x}) \le 0,\ j = 1, \ldots, p\} \end{array} \tag{25}$$

sendo que o símbolo $\widetilde{\min}$ denota uma versão relaxada da função min tradicional e essa relaxação é a responsável em interpretar uma relação que não é bem conhecida, isto é, as k funções objetivo devem ser minimizadas tanto quanto possível, dentro do espaço factível.

Essas imprecisões podem ser descritas por conjuntos *fuzzy* e quantificados por funções de pertinência, $\mu_i(f_i(\mathbf{x})),\ i = 1, \ldots, m$, para todas as funções objetivo. Em geral nos problemas de minimização, a meta *fuzzy* pode ser descrita como "substancialmente menor que ou igual a algum valor p_i", onde d_i é a violação máxima permitida, ou seja, é o menor valor que a função de pertinência pode assumir e o seu nível de satisfação é zero. Essas funções de pertinência são funções monotonicamente decrescentes e podem ser formuladas da seguinte maneira:

$$\mu_i(f_i(\mathbf{x})) = \begin{cases} 1 & f_i(\mathbf{x}) \le f_i \\ p_i(\mathbf{x}) & f_i < f_i(\mathbf{x}) \le f_i + d_i \\ 0 & f_i(\mathbf{x}) > f_i + d_i \end{cases} \tag{26}$$

onde f_i representa o i-ésimo valor da função objetivo, tal que o grau da função de pertinência $\mu_i(f_i(\mathbf{x}))$ seja igual a 1, enquanto $f_i + d_i$ assume valor igual a 0 nessa função de pertinência. Os demais graus dessa função de pertinência, que está dentro de intervalo $[0, 1]$, são expressos pela função monotonicamente crescente $d_i(\mathbf{x})$, para cada $i = 1, \ldots, m$.

Diante desse contexto, a definição básica de uma solução não-dominada pode ser modificada para que atenda a condição mais flexível da função min. Logo, a definição modificada, que é chamada solução ótima M-Pareto, de uma solução não-dominada pode ser descrita como segue:

Definição 4.8 $\mathbf{x}^* \in \Omega$ *é dito ser uma solução ótima M-Pareto para um problema de programação multiobjetivo com meta* fuzzy *se, e somente se, não existe um outro* $\mathbf{x} \in \Omega$ *tal que* $\mu_i(f_i(\mathbf{x}^*)) \ge \mu_i(f_i(\mathbf{x}))$ *para todo* $i = 1, \ldots, m$ *e* $\mu_j(f_j(\mathbf{x})) \ne \mu_j(f_j(\mathbf{x}^*))$ *para ao menos um* j.

Assumindo que estamos tratando problemas não-lineares, a definição acima é somente válida para problemas que podemos garantir uma solução global para cada função objetivo. Entretanto, muitos problemas reais não permitem isso e necessitamos garantir uma solução local. Assim, uma solução ótima M-Pareto local pode ser definida como:

Definição 4.9 $\mathbf{x}^* \in \Omega$ *é dito ser uma solução ótima M-Pareto local para um problema de programação multiobjetivo com meta* fuzzy *se, e somente se, dado um número real* $\delta > 0$ *não existe um outro* $\mathbf{x} \in \Omega \bigcap \mathcal{N}(\mathbf{x}^*, \delta)$ *tal que* $\mu_i(f_i(\mathbf{x}^*)) \ge \mu_i(f_i(\mathbf{x}))$ *para todo* $i = 1, \ldots, m$ *e* $\mu_j(f_j(\mathbf{x})) \ne \mu_j(f_j(\mathbf{x}^*))$ *para ao menos um* j*, sendo que* $\mathcal{N}(\mathbf{x}^*, \delta)$ *representa uma vizinhança de tamanho* δ *do ponto* $\mathbf{x}^*$.

Infelizmente, muitas vezes as soluções ótimas M-Pareto (locais) consistem de um número elevado de pontos e, portanto, o tomador de decisão deve selecionar uma solução final (local) que pertence ao conjunto ótimo M-Pareto (local) como uma solução satisfatória.

4.2.2 Coeficientes *fuzzy* nas funções objetivo

Nesta subseção o foco está em abordagens que tratam de parâmetros imprecisos, que é o mesmo que *fuzzy* nesse trabalho, somente nas funções objetivo, mas foi encontrado somente um trabalho com essa característica e o mesmo está descrito abaixo.

Enfoque de Hussein e Maaty: O artigo [41] usa uma função de distribuição de possibilidade para derivar uma relação de ordem entre números imprecisos. Esse trabalho introduz o conceito de solução eficiente *α-fuzzy* , em que a solução eficiente ordinária é estendida baseada no α-corte de números *fuzzy*.Uma condição necessária e suficiente para definir uma solução candidata é estabelecida. Ele usa a seguinte definição:

Definição 4.10 $x^* \in X$ *é dito ser uma solução eficiente* α-fuzzy *para o problema de programação multiobjetivo não-linear se não existe nenhum outro* $x \in \Omega$ *tal que*

$$\begin{aligned} &\mu_{\tilde{a}}\{a \in \mathbb{R}^{\sum_{j=1}^{n} p_j} | f_j(x, a_j) \le f_j(x^*, a_j)\} = \\ &= \sup_{(a_1,\ldots,a_n)\in A} \min\{\mu_{\tilde{a}_{11}}(a_{11}), \ldots, \mu_{\tilde{a}_{1p_1}}(a_{1p_1}), \\ &\qquad \ldots, \mu_{\tilde{a}_{n1}}(a_{n1}), \ldots, \mu_{\tilde{a}_{np_1}}(a_{np_1})\} \ge \alpha \end{aligned}$$

sendo que $\alpha \in [0,1]$ *e*

$$\begin{aligned} A = \{a \in \mathbb{R}^{\sum_{j=1}^{n} p_j} | f_1(x, a_1) \leq f_1(x^*, a_1), \ldots, \\ f_{j-1}(x, a_{j-1}) \leq f_{j-1}(x^*, a_{j-1}), \\ f_j(x, a_j) < f_j(x^*, a_j), \\ f_{j+1}(x, a_{j+1}) \leq f_{j+1}(x^*, a_{j+1}), \\ \ldots, f_n(x, a_n) \leq f_n(x^*, a_n)\}. \end{aligned}$$

Diante da finalidade de caracterizar a solução eficiente α-*fuzzy* para o problema de programação multiobjetivo não-linear, é possível considerá-lo como um problema multiobjetivo paramétrico. Logo, pode-se analisar as mudanças globais somente nas restrições e nas funções objetivo.

4.2.3 Coeficientes *fuzzy* nas funções objetivo e restrições

Os enfoques descritos nesta subseção tratam de parâmetros incertos nas funções objetivo e no conjunto de restrições. Esses métodos são mais abrangentes que os anteriormente expostos. Isso ocorre porque se não houver incerteza no conjunto de restrições, os mesmos ainda podem ser aplicados.

Enfoque de Sakawa: Um dos primeiros trabalhos sobre Pareto-otimalidade *fuzzy*, que está descrito em [59], usa conjuntos α-corte para resolver problemas de programação multiobjetivo com parâmetros imprecisos. Esses conjuntos são definidos como segue:

$$\begin{array}{ll} \min & f(x,a) \triangleq (f_1(x,a_1), f_2(x,a_2), \ldots, f_k(x,a_k)) \\ \text{s. a} & x \in X(b) \triangleq \{x \in \mathbb{R}^n | g_i(x,b_i) \leq 0, j = 1, \ldots, m\} \\ & (a,b) \in (A,B)_\alpha \end{array} \tag{27}$$

sendo que $A \in \mathbb{F}(\mathbb{R}^k)$ e $B \in \mathbb{F}(\mathbb{R}^j)$ representam, respectivamente, um vetor de parâmetros *fuzzy*, os quais são encontrados nas funções objetivo e no conjunto de restrições. Esses coeficientes são representados por $(a,b) \in \mathbb{R}^k \times \mathbb{R}^j$, que pode ser arbitrariamente definido por qualquer valor pertencente a $(A,B)_\alpha$. Esse subconjunto *fuzzy* é criado por todos os vetores, cujos graus de cada função de pertinência excedem o nível α. O conceito de α-Pareto-otimalidade global e/ou local é introduzida abaixo.

Definição 4.11 (Solução ótima de α-Pareto (local)) $x^* \in X(b)$ *é dito ser uma solução α-Pareto-ótimo (local) para o α-problema de programação multiobjetivo não-linear se e somente se não existe nenhum outro $x \in X(b)(\bigcap \mathcal{N}(x^*, r))$ e $(a,b) \in (A,B)_\alpha(\bigcap \mathcal{N}(a^*, b^*, r'))$ tal que $f_i(x,a_i) \leq f_i(x^*, a_i^*)$, $i = 1, 2, \ldots, k$, sendo que pelo menos uma dessas i desigualdades seja estrita, onde os valores correspondentes dos parâmetros a^* e b^* são chamados parâmetros ótimos para um certo nível α (e $\mathcal{N}(x^*, r) \triangleq \{x \in \mathbb{R}^n | \|x - x^*\| < r\}$ denota a r vizinhança de x^*).*

As soluções α-Pareto-ótimas podem ser obtidas através de uma aplicação direta dos métodos usuais para resolver problemas com um único objetivo. Contudo, esse conjunto de soluções geralmente compreende um número infinito de pontos e a decisão deve selecionar uma única solução global e/ou local baseada em critérios subjetivos.

Em [59], são apresentadas algumas definições para obter soluções eficientes em problemas de otimização multiobjetivo com incertezas nas diferentes formas. As três principais definições descritas por [59] estão a seguir:

Definição 4.12 (solução M-pareto-ótima (local)) $\mathbf{x}^* \in X$ *é dito ser uma solução M-pareto-ótima (local) para o PNLMO generalizado se, e somente se, não existe outro $\mathbf{x} \in X$ $(\cap N(\mathbf{x}^*, \delta))$ tal que $\mu_i(f_i(x)) \geq \mu_i(f_i(\mathbf{x}^*))$ para todo i e $\mu_i(f_j(x)) \neq \mu_i(f_j(\mathbf{x}^*))$ para ao menos um j, sendo que $N(\mathbf{x}^*, \delta)$ denota a δ vizinhança de $\mathbf{x}^*$.*

Definição 4.13 (solução α-pareto-ótima (local)) $\mathbf{x}^* \in X(\mathbf{b})$ *é dito ser uma solução α-pareto-ótima (local) para o α-PNLMO se, e somente se, não existe outro $\mathbf{x} \in X(\mathbf{b})$ $(\cap N(\mathbf{x}^*, \delta))$ e $(\mathbf{a}, \mathbf{b}) \in (\tilde{\mathbf{a}}, \tilde{\mathbf{b}})_\alpha$ $(\cap N(\mathbf{a}^*, \mathbf{b}^*, \delta))$ tal que $f_i(\mathbf{x}, \mathbf{a}_i) \leq f_i(\mathbf{x}^*, \mathbf{a}_i^*)$, $i = 1, \ldots, k$, com uma inequação estrita para ao menos um i, sendo que os valores correspondentes dos parâmetros $\mathbf{a}^*$ e $\mathbf{b}^*$ são chamados parâmetros α-cortes ótimos (locais).*

Definição 4.14 (solução M-α-pareto-ótima (local)) $\mathbf{x}^* \in X(\mathbf{b})$ *é dito ser uma solução α-pareto-ótima (local) para o α-PNLMO generalizado se, e somente se, não existe um outro $\mathbf{x} \in X(\mathbf{b})$ $(\cap N(\mathbf{x}^*, \delta))$ $(\cap N(\mathbf{a}^*, \mathbf{b}^*, \delta))$ e $(\mathbf{a}, \mathbf{b}) \in (\tilde{\mathbf{a}}, \tilde{\mathbf{b}})_\alpha$ $(\cap N(\mathbf{a}^*, \mathbf{b}^*, \delta))$ tal que $\mu_i(f_i(\mathbf{x}, \mathbf{a}_i)) \geq \mu_i(f_i(\mathbf{x}^*, \mathbf{a}_i^*))$, para todo i e $\mu_j(f_j(\mathbf{x}, \mathbf{a}_j)) \neq \mu_j(f_j(\mathbf{x}^*, \mathbf{a}_j^*))$ para ao menos um j, sendo que os valores correspondentes dos parâmetros $\mathbf{a}^*$ e $\mathbf{b}^*$ são chamados parâmetros α-cortes ótimos (locais).*

Em [61] é apresentado um algoritmo genético para um problema de *Job Shop* multiobjetivo, com tempo de processamento e data de entrega imprecisos. As soluções são representadas por matrizes tarefas *versus* máquinas dos tempos *fuzzy* de finalização e dos operadores genéticos mantêm a diversidade da população através da geração de indivíduos com baixa similaridade.

Em [63], problemas multiobjetivo não-convexos com coeficientes *fuzzy* nos objetivos e restrições são resolvidos através de um algoritmo genético baseado no conceito de soluções **α-Pareto-ótimas**: conjunto de pontos não-dominados para um α-corte dos coeficientes *fuzzy*. O algoritmo faz uso de uma população de referência (constituída por soluções totalmente factíveis) e uma população convencional (com atendimento parcial do conjunto de restrições). A primeira solução de referência é gerada com a resolução de um problema de minimização dos desvios das restrições. Ao longo das gerações, novas soluções factíveis são geradas através do Método de *bissecção*, aplicado na direção formada por uma solução de referência e um indivíduo da população convencional. Este último procedimento é executado após a aplicação dos operadores genéticos.

Em [62] e [60] são apresentados algoritmos genéticos para problemas multiobjetivo com diferentes níveis de prioridades entre os objetivos. Os problemas são resolvidos de forma interativa, permitindo ao decisor informar os níveis de satisfação dos objetivos prioritários durante a execução do algoritmo.

Enfoque de Ammar: Em [1] é mostrada as soluções eficientes dos problemas de programação multiobjetivo quadrática *fuzzy*. Alguns teoremas são usados para encontrar uma solução ótima da programação multiobjetivo quadrática escalar relativa a coeficientes *fuzzy* usando vetor de decisão como variáveis *fuzzy*. Considere o seguinte problema de programação multiobjetivo quadrática aleatória *fuzzy*:

$$\begin{array}{ll} \min_{\tilde{X}\in\tilde{M}} & \tilde{f}^r(\tilde{X}) = \tilde{X}^T\tilde{D}^r\tilde{X}, \quad r = 1,2,\ldots,k \\ \text{sujeito a:} & \tilde{M} = \left\{\tilde{X}\in\mathbb{R}^n | \tilde{A}\tilde{X} \lesssim \tilde{B}, \tilde{X} \gtrsim 0\right\} \end{array} \tag{28}$$

sendo que $\tilde{D}^r = (\tilde{d}^r_{ii})_{n\times n}, \quad r = 1,2,\ldots,k$, $\tilde{A} = (\tilde{a}_{ji})_{m\times n}, \tilde{B} = (\tilde{b}_1,\ldots,\tilde{b}_m)^T, \tilde{X} = (\tilde{x}_1,\ldots,\tilde{x}_n), \tilde{d}^r_{ii}, \tilde{a}_{ji}, \tilde{b}_j \in \mathbb{F}(\mathbb{R})$. $\tilde{X}$ não é um vetor aleatório *fuzzy* e $\tilde{X} \gtrsim 0$ significa $(x_i)_\alpha \geq 0$ para qualquer $i = 1,\ldots,n$. As relações $\lesssim$ e $\gtrsim$ denotam as inequações *fuzzy*.

Então, a técnica desenvolvida para resolver o Problema 28 o divide em vários modelos para cada um dos diferentes níveis de α. É claro que para qualquer valor de $\alpha \in [0,1]$, os modelos são problemas de programação quadrática com números aleatórios. Os autores mostram que as condições de Karush-Kuhn-Tucker para o caso quadrático o reduzem a um problema linear complementar. Pode ser provado que se eles têm soluções factíveis, então existem soluções ótimas aleatórias, e valores ótimos são variáveis aleatórias em (Ω, A).

Enfoque de Kuwano: O trabalho realizado por [48] usou uma função de distribuição de probabilidade para derivar uma relação de ordem entre números *fuzzy*. É usado nesse artigo o grau de possibilidade para transformar funções objetivo do problema original em restrições do novo problema, sendo que esses dois problemas são equivalentes. O objetivo do novo problema de otimização multiobjetivo é maximizar o grau de possibilidade das novas restrições ou minimizar o conjunto de nível de aspiração de cada uma das funções objetivo do problema original, que são fornecidos pelo decisor. É interessante ressaltar que esse trabalho emprega as definições clássicas de Pareto-otimalidade, pois as novas funções objetivo são funções que pertencem ao conjunto de números reais e a função de distribuição de possibilidade é somente usada na formulação de um problema de otimização multiobjetivo *fuzzy*.

Enfoque de Li, Ida e Gen: Em [50] é apresentado um algoritmo genético para o problema multiobjetivo de distribuição dos produtos com veículos heterogêneos e custos de transporte *fuzzy*. No algoritmo, as soluções são representadas por matrizes tridimensionais e as comparações entre os objetivos *fuzzy*, base dos processos de avaliação e seleção das soluções, são realizadas com a aplicação de funções de *índices*. Como o modelo apresenta um número elevado de restrições de igualdade, a população é inicializada e evoluída utilizando procedimentos de factibilização.

Nesta seção foi apresentada uma revisão bibliográfica sobre algumas abordagens já publicadas que resolvem problemas de programação multiobjetivo. Muitos trabalhos descrevem enfoques específicos para resolver problemas multiobjetivos lineares, mas nesse trabalho estamos focando em abordagens mais gerais, isto é, que sirvam tanto para resolver problemas multiobjetivos lineares como não-lineares.

5. CONCLUSÃO

No levantamento bibliográfico realizado, constatou-se a necessidade de desenvolver métodos baseados em Computação Flexível para resolver, de maneira satisfatórias, problemas práticos bastante complexos. Ademais, os problemas de programação multiobjetivo são muito importantes tanto do ponto de vista teórico como prático. As estratégias desenvolvidas nos Algoritmos Evolutivos descrito neste trabalho são baseadas no conceito de não-dominância, proposto nos trabalhos de Pareto, e buscam tornar a população ao final das gerações a própria fronteira de soluções não-dominadas.

Entretanto, os problemas de programação matemática que encontramos no mundo real não podem, muitas vezes, ser modelados de maneira clara e precisa porque os dados são ambíguos, vagos e imprecisos. Como elemento deste conjunto, os problemas de programação multiobjetivo também apresentam esses tipos de incertezas. Essa ambiguidade é natural e está presente em situações da vida real que requerem soluções precisas. Existem algumas técnicas para representar esses dados imprecisos, como por exemplo processos estocásticos, caos, aproximação a valores conhecidos, lógica *fuzzy*, etc. Essa última foi a escolhida para ser usada neste trabalho por se adaptar bem aos problemas de programação multiobjetivo *fuzzy*. Na literatura existem várias dessas técnicas e algumas delas foram mostradas neste trabalho. Os dados *fuzzy* podem estar presentes em várias partes do problema de otimização multiobjetivo, tais como:

custos das funções objetivo, coeficientes das funções de restrição e relações de ordem que geram o conjunto factível, e até mesmo nas variáveis. As abordagens de Pareto otimalidade que foram apresentadas neste trabalho usam diferentes técnicas de transformação de números *fuzzy* em clássico.

Agradecimentos

O autor agradece às agências CAPES e CNPq o apoio financeiro concedido para realização deste trabalho.

REFERÊNCIAS

[1] E. E. AMMAR. "On solutions of fuzzy random multiobjective quadratic programming with application in portfolio problem". *Information Sciences*, vol. 178, pp. 468–484, 2008.

[2] S. APPADOO, S. BHATT and C. BECTOR. "Application of possibility theory to investment decision". *Fuzzy Optimization and Decision Making*, vol. 7, pp. 35–57, 2008.

[3] M. ARIA and C. CUCCURULLO. "bibliometrix: An R-tool for comprehensive science mapping analysis". *Journal of Informetrics*, vol. 11(4), pp. 959–975, 2017.

[4] H. M. AYRES, L. M. QUEIROZ, L. C. B. SANTOS and R. C. SILVA. "Algoritmo evolutivo para problemas de otimização multiobjetivo com incertezas". In *XXXIX Simpósio Brasileiro de Pesquisa Operacional*, Fortaleza, CE, Setembro 2007.

[5] T. BÄCK. *Evolutionary algorithms in theory and practice,*. Oxford University Press, New York, 1996.

[6] M. S. BAZARAA, H. D. SHERALI and C. M. SHETTY. *Nonlinear programming - theory and algorithms*. Jonh Wiley & Sons, New York, USA, second edition, 1993.

[7] E. BEALE. "On quadratic programming". *Naval Research Logistics Quarterly*, vol. 6, pp. 227–244, 1959.

[8] R. E. BELLMAN and L. A. ZADEH. "Decision-marking in a fuzzy environment". *Management Science*, vol. 17, no. 4, pp. B141–B164, 1970.

[9] L. CAMPOS and J. L. VERDEGAY. "Linear programming problems and ranking of fuzzy numbers". *Fuzzy Sets and Systems*, vol. 32, pp. 1–11, 1989.

[10] V. CHANKONG and Y. Y. HAIMES. *Multiobjective decision making: Theory and Methodology*, volume 8 of *North Hollando series in system science and engineering*. North Holland, New York, USA, 1983.

[11] M. C. A. C. COELLO. "A comprehensive survey of evolutionary-based multiobjective optimization techniques". *Knowledge and Information Systems: An International Journal* , vol. 1(3), pp. 269–308, 1999.

[12] M. C. A. C. COELLO and A. H. AGUIRRE. "Design of combinational logic circuits through an evolutionary multiobjective optimization approach". *Artificial Intelligence for Engineering, Design, Analysis and Manufacture* , vol. 16(1), pp. 39–53, 2002.

[13] M. C. A. C. COELLO and G. P. TOSCANO. "A Micro-Genetic Algorithm for Multiobjective Optimization". In *First International Conference on Evolutionary Multi-Criterion Optimization*, pp. 126–140, 2001.

[14] D. W. CORNE, N, R. JERRAM, J. D. KNOWLES and M. J. OATES. "PESA-II: Regionbased Selection in Evolutionary Multiobjective Optimization". In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001)*, San Francisco, California, pp. 283–290, 2001.

[15] D. W. CORNE, J. D. KNOWLES and M. J. OATES. "The Pareto Envelope-based Selection Algorithm for Multiobjective Optimization". In *Proceedings of the Parallel Problem Solving from Nature*, Paris, France, pp. 839–848, 2000.

[16] M. C. A. C. COELLO, D. .A. VAN VELHUIZEN and G. B. LAMONT. *Evolutionaty algorithms for solving multi-objective problems.* Kluwer Academic Publishers, New York, 2002.

[17] C. CRUZ, R. C. SILVA, J. L. VERDEGAY and A. YAMAKAMI. "A survey of fuzzy quadratic programming". *Recent Patents on Computer Science*, vol. 1, pp. 182–193, 2008.

[18] C. CRUZ, R. C. SILVA, J. L. VERDEGAY and A. YAMAKAMI. "A parametric approach to solve quadratic programming problems with fuzzy environment in the set of constraints". In *IFSA/EUSFLAT 2009*, Lisbon, Portugal, July 2009.

[19] C. CRUZ, R. C. SILVA and J. L. VERDEGAY. "Extending and relating different approaches for solving fuzzy quadratic problems". *Fuzzy Optimization*, vol. 10, pp. 193–210, 2013.

[20] L. DE MOURA. *Um algoritmo genético para otimização multiobjetivo fuzzy.* Dissertação de mestrado, FEEC-UNICAMP, Campinas, 2002.

[21] K. DEB. *Multi-objective optimization using evolutionary algorithms.* John Wiley & Sons, Chichester, 2001.

[22] K. DEB, A. PRATAP, S. AGARWAL and T. MEYARIVAN. "A fast and elitist multiobjective genetic algorithm: NSGA-II". *IEEE Transaction on Evolutionary Computation*, vol. 6(2), pp. 182–197, 2002.

[23] M. DELGADO, J. L. VERDEGAY and M. A. VILA. "A General Model for Fuzzy Linear Programming". *Fuzzy Sets and Systems*, vol. 29, pp. 21–29, 1989.

[24] D. DUBOIS and H. PRADE. *Fuzzy sets and systems: Theory and Application*. Academic Press, San Diego, USA, 1980.

[25] D. DUBOIS and H. PRADE. "Ranking fuzzy numbers in the setting of possibility theory". *Information Sciences*, vol. 30, no. 3, pp. 183–224, 1983.

[26] M. ERICKSON, A. MAYER and J. HORN. "The Niched Pareto Genetic Algorithm 2 Applied to the Design of Groundwater Remediation Systems, in: Zitzler, E., Thiele, L., Deb, K., Coello Coello, C.A., Corne, D. eds., Evolutionary Multi-Criterion Optimization (EMO 2001)", Springer, Berlin, Heidelberg, pp. 681–695, 2002.

[27] M. FARINA and P. AMATO. "Fuzzy optimality and evolutionary multiobjective optimization". In *Second International Conference Evolutionary Multi-Criterion Optimization*, Faro, Portugal, pp. 58–72, 2003.

[28] M. FARINA and P. AMATO. "A fuzzy definition of "optimality"for many-criteria optimization problems". *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans*, vol. 34, no. 3, pp. 315–326, 2004.

[29] P. A. V. FERREIRA. "Otimização multiobjetivo: Teoria e aplicações". Livre docência, Universidade Estadual de Campinas, Abril 1999.

[30] C. A. FLOUDAS, P. M. PARDALOS, C. ADJIMAN, W. R. ESPOSITO, Z. H. GÜMÜS, S. T. HARDING, J. L. KLEPEIS, C. A. MEYER and C. A. SCHWEIGER. *Handbook of test problems in local and global optimization*, volume 33 of *Nonconvex Optimization and Its Applications*. Kluwer Academic Publishers, Dordrecht, 1999.

[31] C. M. FONSECA and P. j. FLEMING. "Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization". In *Proceedings of the Fifth International Conference on Genetic Algorithms*, San Mateo, California, pp. 416–423, 1993.

[32] C. M. FONSECA and P. j. FLEMING. "Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms - Part I: A Unifed Formulation". *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans*, vol. 28, no. 1, pp. 26–37, 1998.

[33] M. C. GOLDBARG and H. P. L. LUNA. *Otimização combinatória e programação linear: modelos e algoritmos*. Campus, Rio de Janeiro, BR, 2000.

[34] D. E. GOLDBERG. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1989.

[35] D. E. GOLDBERG and J. J. RICHARDSON. "Genetic algorithms with sharing for multimodal function optimization". In *Proceedings of the Second International Conference on Genetic Algorithms*, pp. 41–49, 1987.

[36] D. M. HIMMELBLAU. *Applied nonlinear programming*. McGraw-Hill Book Company, Reading, MA, USA, 1972.

[37] W. HOCK and K. SCHITTKOWSKI. *Test examples for nonlinear programming*, volume 187 of *Lecture Notes in Economics and Mathematical Systems*. Spring-Verlag, 1981.

[38] J. H. HOLLAND. *Adaptation in natural and artificial systems*. MIT Press, Ann Arbor, 1975.

[39] J. HORN. "Finite markov chain analysis of genetic algorithms with niching". In *Proceedings of the Fifth International Conference on Genetic Algorithms*, San Mateo, California, pp. 110–117, 1993.

[40] J. HORN, N. NAFPLIOTIS and D. GOLDBERG. "A Niched Pareto Genetic Algorithm for Multiobjective Optimization". In *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, Piscataway, New Jersey, pp. 82–87, 1994.

[41] M. L. HUSSEIN and M. ABDEL AATY MAATY. "The stability notions for fuzzy nonlinear programming problem". *Fuzzy Sets and Systems*, vol. 85, pp. 319–323, 1997.

[42] M. INUIGUCHI, H. ICHIHASHI and H. TANAKA. "Fuzzy Programming: A survey of recent developments, in: Slowinski and Teghem eds., Stochastic versus Fuzzy Approaches to Multiobjective Mathematical Programming under Uncertainty". *Kluwer Academic Boston*, pp. 45–68, Dordrecht 1990.

[43] F. JIMÉNEZ, J. CADENAS, G. SÁNCHEZ, A. GÓMEZ-SKARMETA and J. L. VERDEGAY. "Multi-objective evolutionary computation and fuzzy optimization". *International Journal of Approximate Reasoning*, vol. 43, no. 1, pp. 59–75, 2006.

[44] A. KAUFMANN and M. M. GUPTA. *Introduction to fuzzy arithmetic: theory and applications*. Van Nostrand Reinhold, New York, USA, 1984.

[45] G. J. KLIR and B. YUAN. *Fuzzy sets and fuzzy logic: theory and applications*. Prentice Hall, New Jersey, USA, 1995.

[46] M. KÖPPEN, K. FRANKE and B. NICKOLAY. "Fuzzy-Pareto-dominance driven multiobjective genetic algorithm". In *10th IFSA World Congress*, pp. 450–453, Istanbul, Turkey, 2003.

[47] M. KÖPPEN, R. V. GARCIA and B. NICKOLAY. "Fuzzy-pareto-dominance and its application in evolutionary multi-objective optimization". In *Evolutionary Multi-Criterion Optimization, Third International Conference*, pp. 399–412, Guanajuato, Mexico, 2005.

[48] H. KUWANO. "Inverse problems in fuzzy multiobjective linear programming". In *Second International Conference on Knowledge-Based Intelligent Eletronic Systems*, pp. 559–563, Adelaide, Australia, 2000.

[49] Y. J. LAI and C. L. HWANG. *Fuzzy mathematical programming: methods and applications*, volume 394 of *Lecture Notes in Economics and Mathematical Systems*. Springer, Berlin, 1992.

[50] Y. LI, K. IDA and M. GEN. "Improved genetic algorithm for solving multiobjective solid transportation problem with fuzzy numbers". *Computers in Industrial Engineering*, vol. 33, no. 3-4, pp. 589–592, 1997.

[51] D. G. LUENBERGER. *Linear and nonlinear programming*. Addison-Wesley, Massachusetts, USA, second edition, 1989.

[52] K. M. MIETTINEN. *Nonlinear multiobjective optimization*. Kluwer Academic Publishers, Massachussets, 1999.

[53] C. V. NEGOITA and D. A. RALESCU. *Applications of fuzzy sets to systems analysis*. Birkhauser Verlag, Stuttgard, 1975.

[54] A. OSYCZKA. "Multicriteria optimization for engineering design". In *J. S. Gero Ed. Design Optimization*, pp. 193–227. Academic Press, 1985.

[55] V. PARETO. *Cours d'economique politique*, volume I. Macmillan, Paris, FR, 1897.

[56] V. PARETO. *Le cours d'economique politique*, volume II. Macmillan, London, UK, 1897.

[57] W. PEDRYCZ and F. GOMIDE. *An introduction of fuzzy sets: analisys and design*. MIT press, Cambridge, MA, 1998.

[58] R. S. ROSENBERG. *Simulation of genetic populations with biochemical properties*. Tese de doutorado. Universidade de Michigan, Michigan, 1967.

[59] M. SAKAWA. *Fuzzy sets and interactive multiobjective optimization*. Plenum Press, New York, USA, 1993.

[60] M. SAKAWA. *Genetic algorithms and fuzzy multiobjective optimization*. Klumer Academic Publishers, London, UK, 2002.

[61] M. SAKAWA and R. KUBOTA. "Fuzzy programming for multiobjective job shop scheduling with fuzzy processing time and fuzzy duedate through genetic algorithms". *European Journal of Operational Research*, vol. 120, no. 2, pp. 393–407, 2000.

[62] M. SAKAWA and I. NISHIZAKI. "Interactive Fuzzy Programming for Two-Level Nonconvex Programming Problems with Fuzzy Parameters Through Genetic Algorithms". *Fuzzy Set and Systems*, vol. 127, pp. 185–197, 2002.

[63] M. SAKAWA and K. YAUCHI. "Interactive Decision Making for Multiobjective Nonconvex Programming Problems with Fuzzy Numbers Through Coevolutionary Genetic Algorithms". *Fuzzy Set and Systems*, vol. 114, pp. 151–165, 2000.

[64] J. D. SCHAFFER. *Multiple objective optimization with vector evaluated genetic algorithms*. Tese de doutorado. Vanderbilt University, 1984.

[65] J. D. SCHAFFER and J. J. GREFENSTETTE. "Multiobjective learning via genetic algorithms". In *Proceedings of the 9th International Joint Conference on Artificial Intelligence*, California, 593-595, 1985.

[66] K. SCHITTKOWSKI. *More test examples for nonlinear programming codes*. Spring-Verlag, 1987.

[67] R. C. SILVA and T. A. ALMEIDA. "Programação multiobjetivo irrestrita com incertezas". In *XLI Simpósio Brasileiro de Pesquisa Operacional*, Porto Seguro, BA, Setembro 2009.

[68] R. C. SILVA, C. CRUZ and A. YAMAKAMI. "A parametric method to solve quadratic programming problems with fuzzy costs". In *IFSA/EUSFLAT 2009*, Lisbon, Portugal, July 2009.

[69] R. C. SILVA and A. YAMAKAMI. "Definition of fuzzy Pareto-optimality by using possibility theory". In *IFSA/EUSFLAT 2009*, Lisbon, Portugal, July 2009.

[70] R. C. SILVA. "O uso de abordagens paramétrica e possiblística em programação multiobjetivo com incertezas". In *XLI Simpósio Brasileiro de Pesquisa Operacional*, Porto Seguro, BA, Setembro 2009.

[71] R. C. SILVA, C. CRUZ, J. L. VERDEGAY and A. YAMAKAMI. *Fuzzy optimization: Recent developments and applications*, chapter A survey of fuzzy convex programming models. Springer-Verlag, 2009.

[72] R. C. SILVA and A. YAMAKAMI. "The use of possibility theory in the definition of fuzzy Pareto-optimality". *Fuzzy Optimization and Decision Making*. 2011.

[73] N. SRINIVAS and K. DEB. "Multiobjective optimization using nondominated sorting in genetic algorithms". *Evolutionary Computation* , vol. 2(3), pp. 221–248, 1994.

[74] H. TANAKA, T. OKUDA and K. ASAI. "On fuzzy-mathematical programming". *Journal of Cybernetics*, vol. 1, no. 4, pp. 37, 46 1974.

[75] D. A. VAN VELDHUIZEN and G. b. LAMONT. "Multiobjective Evolutionary Algorithm Test Suites". In *Proceedings of the 1999 ACM Symposium on Applied Computing*, San Antonio, Texas, pp 351–357, 1999.

[76] D. A. VAN VELDHUIZEN and G. b. LAMONT. "Multiobjective Optimization with Messy Genetic Algorithms". In *Proceedings of the 2000 ACM Symposium on Applied Computing*, Villa Olmo, Italy, pp. 470–476, 2000.

[77] S. WANG and S. ZHU. "On fuzzy portfolio selection problems". *Fuzzy Optimization and Decision Making*, vol. 1, pp. 361–377, 2002.

[78] B. WERNERS. "Interative multiple objective programming suject to flexible constraints". *European Journal of Operational Research*, vol. 31, pp. 342–349, 1987.

[79] P. WOLFE. "The simplex method for quadratic programming". *Econometrica*, vol. 27, pp. 382–398, 1959.

[80] L. A. ZADEH. "Fuzzy sets". *Information and Control*, vol. 8, pp. 338–353, 1965.

[81] L. A. ZADEH. "Fuzzy sets as a basis for a theory of possibility". *Fuzzy Sets and Systems*, vol. 1, no. 1, pp. 3–28, 1978.

[82] L. A. ZADEH. "Fuzzy logic, neural networks, and soft computing". *Communication of the ACM*, vol. 37, no. 3, pp. 77–84, 1994.

[83] L. A. ZADEH. "Soft computing and fuzzy logic". *IEEE Software*, pp. 48–55, 1994.

[84] L. A. ZADEH. "A new direction in AI: Toward a computational theory of perceptions". *AI Magazine*, vol. 22, no. 1, pp. 73–84, 2001.

[85] L. A. ZADEH. "Is there a need for fuzzy logic?" *Information Sciences*, vol. 178, pp. 2751–2779, 2008.

[86] H. J. ZIMMERMANN. "Description and Optimization of Fuzzy Systems". *International Journal of General Systems*, vol. 2, pp. 209–215, 1976.

[87] H. J. ZIMMERMANN. "Fuzzy programming and linear programming with several objective functions". *Fuzzy Sets and Systems*, vol. 1, pp. 45–55, 1978.

[88] H. J. ZIMMERMANN. “Fuzzy mathematical programming”. *Computer & Operation Research*, vol. 10, no. 4, pp. 291–298, 1983.

[89] H.-J. ZIMMERMANN. *Fuzzy set theory and its applications*. Kluwer Academic Publishers, Massachusetts, USA, third edition, 1996.

[90] E. ZITZLER, M. LAUMANNS and S. BLEULER. *A tutorial on evolutionary multiobjective optimization*, In *Gandileux, X., Sevaux, M., Sörensen, K. e T'kindt, V. (Eds.) Metaheuristics for multiobjective optimization*, volume 353 of *Lecture Notes in Economics and Mathematical Systems*. Springer, Berlin, 3–37, 2004.

[91] E. ZITZLER, M. LAUMANNS and and L. THIELE. *SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimizatio*, In *In CIMNE*, Barcelona, Spain, pp. 19–26, 2002.

[92] E. ZITZLER and L. THIELE. “Multiobjective Evolutionary Algorithms: A comparative case study and the strength pareto approach”. *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.

[93] J. B. ZYDALLIS, D. A. VAN VELDHUIZEN and G. B. LAMONT. *A Statistical Comparison of Multiobjective Evolutionary Algorithms Including the MOMGA-II*, In *In First International Conference on Evolutionary Multi-Criterion Optimization*, pp. 226–240, 2001.