
Estado da Arte de Medidas de Desempenho de Algoritmos de Otimização Multiobjetivo

Jefferson Barbosa Amorim Silva

jeffauh@gmail.com

Universidade Federal de Ouro Preto, Ouro Preto, MG, Brasil

Thiago Fontes Santos

santostf@iceb.ufop.br

Universidade Federal de Ouro Preto, Ouro Preto, MG, Brasil

Sebastião Martins Xavier

semarx@iceb.ufop.br

Universidade Federal de Ouro Preto, Ouro Preto, MG, Brasil

Resumo

Em problemas de otimização multiobjetivo é comum o uso de procedimentos algorítmicos para se encontrar uma aproximação do conjunto solução, uma vez que determinar o conjunto exato de soluções é, em geral, inviável. Diante disso, com a popularização destes procedimentos nas últimas décadas é necessário distinguir, conforme for o objetivo do algoritmo, a qualidade das aproximações de tais procedimentos. As medidas existentes focam-se em duas características principais, a saber, 1) Convergência e 2) Distribuição das soluções. Dificilmente uma única métrica irá medir todos os aspectos mencionados acima. Além disso, para avaliar alguns desses aspectos é necessário o conhecimento do conjunto exato, e essa informação normalmente, não está disponível. Associado a isso, espera-se que tais medidas tenham um custo computacional baixo. Esta proposta visa estudar as mais diversas medidas, reunindo as principais delas em um único artigo, fazendo um comparativo entre as mesmas. Iremos estabelecer o estado da arte nessa área de pesquisa em otimização.

Palavras-chave

Modelagem matemática, Otimização Multiobjetivo, Medidas de Desempenho.

1 Introdução

Ao projetar um dado produto, procura-se minimizar o custo de produção e maximizar a qualidade deste. A necessidade de procurar uma solução que satisfaça as duas condições e a natureza oposta dela, fazem com que métodos de otimização sejam importantes [4].

Tratando de objetivos conflitantes, como no desenvolvimento de um produto, em que deseja-se minimizar o custo de produção enquanto se maximiza a quali-

dade do mesmo, encontrar o mínimo global para o custo significa ter custo 0, ou seja, não produzir o produto, enquanto maximizar a qualidade implica um custo de produção muito alto, o que pode tornar o valor de revenda do produto muito alto. Desta maneira, a resposta que minimiza apenas um dos objetivos, não minimiza o outro.

Desta forma, a solução deste problema levando em consideração estes dois objetivos não pode ser apenas um valor. Na realidade, é um conjunto de pares (Custo, Qualidade) que se relacionam entre si. Este conjunto representa as trocas a serem feitas em relação ao custo e a qualidade. Assim, apenas com este conjunto, não pode-se decidir qual dessas respostas é melhor que outra. Como Deb afirma em [4], problemas de otimização multi-objetivo, tem como objetivo buscar um conjunto de trocas a serem feitas em relação aos objetivos ao considera-los igualmente importantes. Uma vez obtido tal conjunto, pode-se, com informações externas, decidir qual delas é melhor. Em outras palavras, escolher uma solução que minimiza apenas um objetivo pode significar ignorar uma outra que é melhor de uma maneira mais geral [11], isso significa que na grande maioria dos casos, considerar a minimização de apenas um objetivo, significa obter uma resposta pior em um, ou vários outros objetivos. Ao se obter este conjunto de respostas não é possível decidir entre duas respostas, qual é a melhor, pois cada uma representa uma troca a ser feita em relação aos objetivos. Este conjunto, conhecido como *Frenteira Pareto Ótima*, nos dá o conjunto das melhores trocas a serem feitas. Porém, na grande maioria dos problemas, encontrá-lo não é computacionalmente viável devido à complexidade e quantidade das funções em questão. Desta maneira, utiliza-se algoritmos para obter uma aproximação boa o suficiente, em um tempo hábil, chamada de *Frenteira Pareto Aproximada*. Espera-se desta aproximação, que ela dê informações boas o suficiente para a solução do problema dado.

Dentre os diversos algoritmos propostos no campo de otimização, os *Algoritmos Evolutivos* (AE) são os quais imitam a evolução na natureza na procura da solução ótima [4]. Estes algoritmos utilizam operadores que imitam mutação, reprodução e seleção, assim como na natureza. Schaffer *apud* Deb [4], em 1984 propôs em sua tese de doutorado a primeira aplicação de um algoritmo evolutivo para problemas multiobjetivo, a *Vector-Evaluated Genetic Algorithm* (VEGA)

. Baseado nesta proposta e em notas escritas por David E. Goldberg, muitos outros algoritmos foram propostos, dentre eles podemos citar o *Non-Dominated Sorting Genetic Algorithm* (NSGA) proposto por Srinivas e Deb [4], este que a sua segunda versão, NSGAI foi considerado um dos melhores algoritmos por 10 anos e foi recentemente atualizado para a sua terceira versão, NSGAIII.

Com o surgimento de diversos AE, comparar suas respostas se tornou necessário, uma vez que dado um problema, deseja-se obter uma solução boa o suficiente em um intervalo de tempo razoável. Para isso, vários *Indicadores de Desempenho* são propostos. Estes, têm como objetivo avaliar a qualidade da resposta de um dado algoritmo. Cada indicador foi proposto para avaliar a qualidade do conjunto dado. Alguns avaliam por exemplo, a quantidade de elementos, outros a distância à um conjunto referência, caso conhecido, enquanto existe outros que avaliam o quanto eles estão distribuídos pelo espaço. Assim pode-se avaliar qual dos dois conjuntos representados na Figura 1 é melhor em aproxima a Fronteira Pareto Ótima.

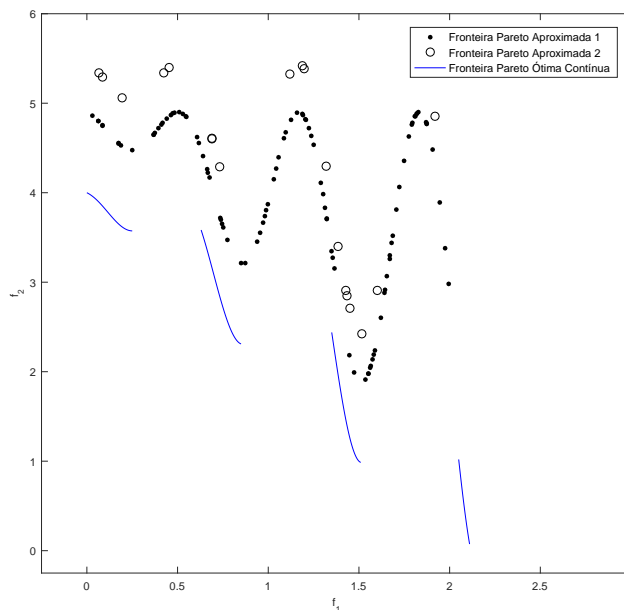


Figura 1: Comparação de duas Fronteiras Pareto

2 Problema de Otimização

Chamamos de *Problema de Otimização Mono-Objetivo* o problema que envolve apenas uma função objetivo [4]. Neste tipo de problema, procura-se o mínimo (ou máximo) global para este único objetivo. Considere o problema de minimização da função $f(x) = e^{x^3+2x^2+x}$ no intervalo $[-2, 0]$. Utilizando cálculo e derivando $f(x)$, encontramos que $x = -1$ e $x = -\frac{1}{3}$ são pontos críticos da função. Porém, para calcular o mínimo no intervalo, também precisamos levar em consideração os extremos do intervalo, ou seja, $x = -2$ e $x = 0$. Assim, obtemos: $f(-1) = 1$, $f(-\frac{1}{3}) = \exp^{-\frac{4}{27}} \approx 0.86$, $f(-2) = \exp -2 \approx 0.13$ e $f(0) = 1$. Conseqüentemente, o mínimo absoluto de $f(x)$ no intervalo em questão é atingido quando $x = -2$.

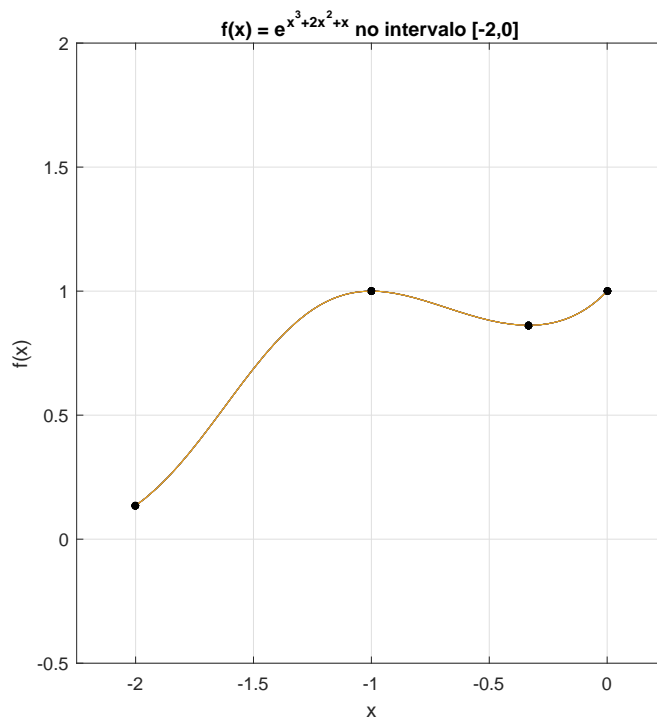


Figura 2: Pontos críticos de $f(x)$ assim como os extremos do intervalo

Nos interessa quando este problema envolve mais de um objetivo, chamamos este de *Problema de Otimização Multi-Objetivo* (PMO).

Deb [4] define problema de otimização Multi objetivo como:

$$\begin{aligned}
 \text{Minimizar/Maximizar} \quad & f_m(x), & m = 1, \dots, M; \\
 \text{sujeito à} \quad & g_j(x) \geq 0, & j = 1, \dots, J; \\
 & h_k(x) = 0 & k = 1, \dots, K; \\
 & x_i^{(L)} \leq x_i \leq x_i^{(U)} & i = 1, \dots, n.
 \end{aligned} \tag{1}$$

Nesta definição, as funções $f_m(x)$ são os M objetivos a serem considerados, cada um com até n variáveis. As funções g_j, h_k são as chamadas de funções de restrições e $x_i^{(L)}, x_i^{(U)}$ são cotas inferiores e superiores respectivamente para cada coordenada x_i .

A maximização de um objetivo pode ser convertida em um problema de minimização apenas multiplicando tal objetivo por -1 [4]. Desta maneira, trataremos apenas de problemas de minimização.

A solução de um problema de otimização é um vetor x com n variáveis $x = (x_1, \dots, x_n)$. Um vetor x que satisfaz todas as restrições é chamado de *Solução Aceitável* caso contrário, o vetor é dito *Não-Aceitável*. O conjunto de todas as soluções aceitáveis é chamado de *Região Aceitável* e analogamente os vetores não-aceitáveis determina uma região chamada de *Região Não-Aceitável*.

Podemos estar interessados em minimizar as funções $f(x) = x^2$ e $g(x) = x^3 - x^2 + 1$ restritas ao intervalo $[0, 2]$ ilustradas na Figura 3, simultaneamente. Desta forma, estamos interessados em procurar um valor $x \in \mathbb{R}$ tal que tenhamos o menor valor possível para $f(x)$ e $g(x)$. Repare que tanto o valor de x tal que $f(x)$ é o mínimo absoluto no intervalo não minimiza g e tanto o valor de x tal que $g(x)$ é o mínimo absoluto no intervalo não minimiza $f(x)$. Diremos que estes dois objetivos são conflitantes, pois para minimizar uma objetivo, temos que abrir mão de um valor melhor no outro. Este exemplo simples ilustra a dificuldade encontrada ao tentar minimizar múltiplas funções conflitantes.

Ter vários objetivos a serem otimizados, quase sempre conflitantes, implica que existe uma ordem parcial invés de total no espaço [11]. Ou seja, dada um conjunto, não se pode definir uma resposta como sendo absolutamente melhor que todas as outras, mas sim, um conjunto de soluções que são preferíveis em relação as outras. Assim, a ideia de resposta “ótima” muda, sendo o objetivo

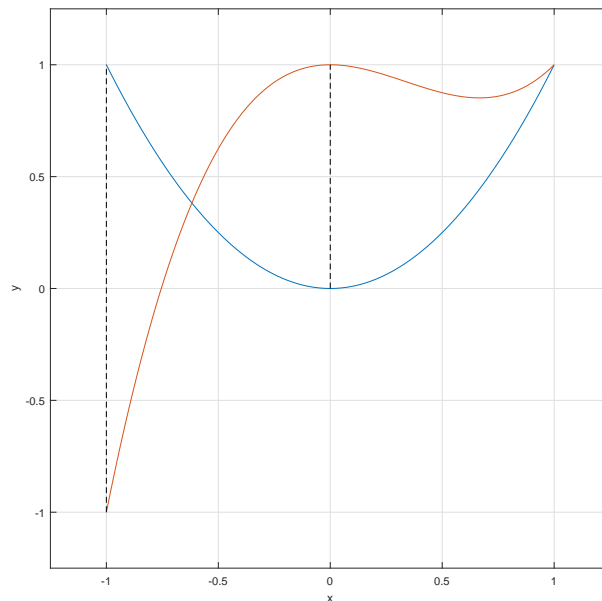


Figura 3: Problema de Minimização

obter uma relação entre os objetivos ao invés de uma única solução como em otimização com apenas um objetivo [11]. A notação de “ótimo” mais adotadas foi originalmente proposta por Francisco Ysidro Edgeworth e depois generalizada por Vilfred Pareto [3].

Primeiramente, é utilizado o conceito de *Dominância*. Diremos que um vetor $x \in \mathbb{F}$ *domina* um elemento $y \in \mathbb{F}$ se x se desempenhar “melhor” ou igual a y em todos os objetivos e ser estritamente “melhor” que y em pelo menos um objetivo, denotaremos por $x \succ y$.

Dado os pontos (f_1, f_2) da Figura 4, caso o objetivo seja minimizar f_1 e f_2 , podemos ver, por exemplo que $A \succ B$, $D \succ C$ mas não podemos relacionar B e D .

$x^* \in \mathbb{F}$ é dito ser Pareto-Ótimo se $\forall y, x^* \succ y$, em outras palavras x^* é dito Pareto-Ótimo se não existir outro valor possível x que diminua algum objetivo sem simultaneamente causar aumento em pelo menos um outro objetivo.[3]. O conjunto de todos estes pontos, não dominados por nenhum outro é chamada *Frenteira Pareto, FP*. O conjunto dos $x \in \Omega$ tais que $F(x)$ é um elemento da

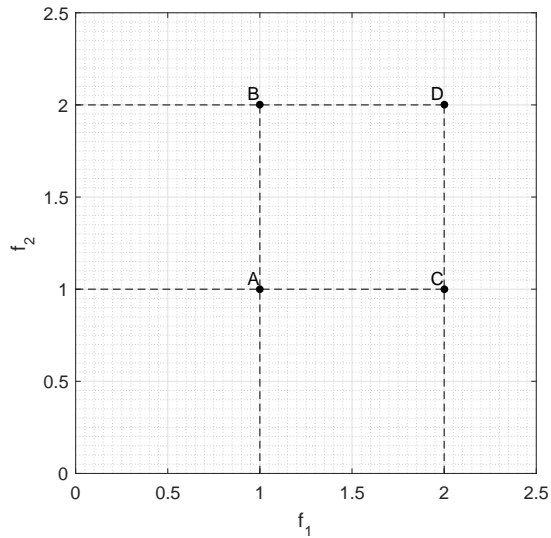


Figura 4: Exemplo de Dominância

Fronteira Pareto é chamado de *Conjunto Pareto*, denotado por CP .

De acordo com Deb [4], pode-se conjecturar que existem dois objetivos em otimização multi-objetivo:

1. Encontrar um conjunto de soluções o mais próximo possível à Fronteira Pareto Ótima.
2. Encontrar um conjunto o quão diverso quanto o possível.

A primeira propriedade garante que, ao se aproximar da Fronteira Pareto Ótima, o conjunto obtido está próximo de ser ótimo. Enquanto a diversidade da resposta garante que esta fornece um bom conjunto de trocas a serem feitas em relação a cada objetivo. Dado um problema de minimização, é desejado encontrar tais conjuntos, Fronteira e Conjunto Pareto. Obter tais conjuntos explicitamente, quando se trata de diversas funções de várias variáveis torna-se computacionalmente inviável. Desta forma, são utilizados Algoritmos para se obter uma aproximação destes conjuntos *Fronteira e Conjunto Pareto Reais*, também chamados de conjunto referência.

Coello *et al* [2, 3] listam diversos trabalhos em que diversos problemas reais em que a otimização multi-objetivo é aplicada. Estes problemas vão desde

algoritmos para navegação de veículos autônomos à problemas de descrição da sequência de DNA.

3 Algoritmos de Otimização

Enquanto o objetivo de um algoritmo para um problema de otimização com apenas um objetivo é simples, obter o mínimo absoluto da função dada. Porém, dado certo problema, pode-se estar interessado não em obter o mínimo absoluto, pelo tempo requerido para o algoritmo alcançá-lo, mas encontrar uma solução boa o suficiente, mesmo que não seja a ideal, em um tempo razoável. Não apenas isto, mas pode-se estar interessado em obter um conjunto com diversas possíveis respostas “aceitáveis”. Quando se trata de otimização multiobjetivo, o objetivo não é encontrar uma única solução, mas sim uma aproximação boa para o conjunto ótimo em um tempo razoável. Assim, como Dan Simon [9] menciona, espera-se de um algoritmo de otimização multiobjetivo que ele maximize o número de indivíduos à uma certa distância do conjunto Pareto Ótimo, minimize a distância entre a resposta do algoritmo e o conjunto Pareto Ótimo, maximize a diversidade desta resposta, ou seja, que os pontos deste conjunto estejam “bem espalhados” pelo espaço.

Os algoritmos utilizados em otimização podem ser basicamente divididos em dois tipos: os clássicos e os evolucionários.

3.1 Algoritmos Clássicos

Algoritmos clássicos foram os primeiros a serem propostos na tentativa de se resolver um P.M.O.. Estes podem ser divididos basicamente em três tipos [3]: Os *Métodos Enumerativos*, que dado um espaço finito discreto de possíveis respostas, avalia qual é a melhor. Além de ineficientes, estes se tornam inutilizáveis quando o espaço se torna muito amplo ou ainda é contínuo; Os *Métodos Determinísticos* implementam algum conhecimento do espaço de pesquisa pra encontrar a resposta do problema. Dentre estes, podemos citar, os algoritmos *gananciosos*, que procuram por mínimos locais na busca pelo mínimo global, porém, ao lidar com funções que possuam muito mínimos locais, estes podem ser pouco efetivos e os algoritmos de escalada, que buscam o mínimo a partir de um ponto aleatório no espaço e avaliando a direção de maior decréscimo e assim avançando para

um próximo ponto. Porém, a existência de mínimos locais ou de pontos onde a função estabiliza, diminui a efetividade destes algoritmos. Estes são alguns dos métodos Determinísticos; Os *Métodos Estocásticos*, como *Busca Aleatória*, ou *Caminhada Aleatória* avaliam aleatoriamente valores no espaço, enquanto a segunda leva em consideração o ponto avaliado na etapa anterior. Apesar de eles eventualmente encontrarem um mínimo, a maioria não pode garantir alcançar o mínimo absoluto. Para obter mais informações a cerca dos métodos clássicos, ler Coello *et al* [3], onde são apresentados diversos métodos clássicos.

3.2 Algoritmos Evolutivos

Os algoritmos evolutivos são divididos em dois tipos, os clássicos e os evolutivos.

Deb [4], exhibe diversos métodos clássicos. Métodos clássicos são basicamente maneiras de transformar um problema multiobjetivo em um com apenas um objetivo baseada em parâmetros definidos pelo usuário. Como exemplo, o *Método das Somas Ponderadas* que substitui o problema de minimizar múltiplos objetivos pelo problema de minimizar a soma ponderada destes. Ao realizar esta substituição, e solucionar um problema com apenas um objetivo, a solução (única) obtida depende dos parâmetros utilizados pelo usuário, assim, para obter um conjunto com N vetores, seria necessário rodar o algoritmo pelo menos N vezes com parâmetros diferentes.

Dentre os algoritmos clássicos, existem Os *Métodos Sem Preferência* estes que não utilizam nenhuma preferência sobre os objetivos na busca por soluções enquanto *Métodos a posteriori* utilizam informações sobre a importância de cada objetivo para gerar o conjunto de soluções Pareto-Ótimas. *Métodos a priori* utilizam informações exteriores ao problema sobre a importância de cada objetivo. Finalmente, os *Métodos Interativos* usam a importância de cada objetivo progressivamente durante o processo de otimização. Todos estes geralmente obtêm apenas uma solução.

A maioria dos algoritmos clássicos sofrem de vários problemas como a incapacidade de gerar múltiplas respostas em apenas uma execução, conhecimento prévio do problema para escolha de parâmetros, terem dificuldade com certos tipos de problemas e funções envolvidas.

Dentre os diversos algoritmos propostos no campo de otimização, os *Algoritmos Evolutivos* são os quais imitam a evolução na natureza na procura da solução ótima [4]. Schaffer *apud* Deb [4], em 1984 propôs em sua tese de doutorado a primeira aplicação de um algoritmo evolutivo para problemas multiobjetivo, a *Vector-Evaluated Genetic Algorithm* (VEGA). Este algoritmo opera avaliando os vetores em relação a um objetivo de cada vez e após avaliados todos os objetivos, ele recombina eles para gerar uma nova geração.

Os algoritmos clássicos quanto os primeiros algoritmos evolutivos como o VEGA, citados acima, não utilizam diretamente, em sua implementação, o conceito de Pareto Dominância. Já os próximos utilizam este conceito para dividir as possíveis respostas em conjuntos dependendo da relação de dominância e se ela domina ou não outras respostas. Dan Simon [9] e Coello *et. al* [3] citam e descrevem diversos destes algoritmos. Em todos eles, assim como na natureza, são utilizados operadores evolutivos na tentativa de gerar soluções cada vez melhores. Estes algoritmos começam gerando uma “população” inicial de possíveis respostas, chamadas de indivíduos. Esta população inicial, sofre ação de operadores evolutivos como Mutação, Recombinação e Seleção para gerar uma nova população e continua assim até uma quantidade pré determinada de iterações. A Mutação é controlada pelo usuário, determinado a probabilidade com que, ao final da geração de uma geração, um indivíduo tenha seus valores modificados por um parâmetro também pré-determinado. Recombinação, como o próprio nome diz, recombina os indivíduos com uma probabilidade predeterminada pelo usuário. Finalmente, após aplicada a mutação e recombinação, a seleção dá a maneira na qual os indivíduos serão selecionados para fazer gerar a próxima geração. É natural dar à indivíduos melhores, maior probabilidade de serem geradores de indivíduos na próxima geração. Esta seleção pode ser feita por "Torneios" onde os indivíduos são separados dois a dois e comparados, o “melhor” é selecionado para a próxima geração e o outro descartado, ou então pode ser feita aleatoriamente apenas associando maior probabilidade para indivíduos melhores.

Vamos descrever a estrutura básica de um algoritmos evolutivo baseada na descrição apresentada por Coello *et. al* [3].

- O algoritmo começa gerando aleatoriamente N indivíduos em uma população P ;

- Remove de P indivíduos que são Pareto-Dominados, fazendo $P \rightarrow P^i$;
- Limita a quantidade de indivíduos de P^i que estão muito próximos com parâmetros definidos pelo usuário;
- Aplica operadores evolutivos (Recombinação, Mutação) para gerar novos indivíduos, $P^i \rightarrow P^{ii}$;
- Seleciona indivíduos para a próxima geração. Como mencionado acima, esta seleção pode ocorrer de diversas maneiras. $P^{ii} \rightarrow P^{iii}$;
- Caso o critério de parada não seja atingido (Quantidade de gerações, ou critério de convergência), faz-se $P^{iii} \rightarrow P$;
- Remove indivíduos Pareto Dominados ou não aceitáveis de P^{iii} , faz-se $P^{iii} \rightarrow P_A$, onde P_A é a população atual;
- Mantém arquivados em P^{iv} os indivíduos não dominados e aceitáveis de P^{iii} . Quando uma nova geração P^{iii} é criada, ela é unida com P^{iv} e o operador para avaliar Pareto-Dominância é avaliado novamente;
- Busca local utilizada em algoritmos híbridos ou meméticos, podem proporcionar um bom desempenho ao procurar respostas ao mover a buscar em direção à certas regiões da Fronteira Pareto.

O interessante deste algoritmos é que eles passam a rotina a ser implementada, porém, como estas serão implementadas ficam à critério do usuário. A maneira que mutação ou a recombinação vai ser implementada, por exemplo, pode variar dependendo do problema a ser resolvido. Dan Simon [9] mostra diversas maneiras nas quais Seleção, Recombinação e Mutação podem ser implementadas.

Para esclarecer como estes algoritmos funcionam, vamos apresentar o código de alguns deles. Primeiramente, a Figura 5 mostra o SPEA2, *Strength Pareto Evolutionary Algorithm* proposto por Zitzler **et al** [15]. Este avalia o "quanto" um dado vetor domina os outros medindo a quantidade de vetores que este domina e quantos vetores os vetores que ele domina domina. Assim, um vetor que domina

diversos vetores que também dominam outros é melhor para a evolução em direção à uma resposta ótima.

```

 $N$  = Tamanho da População;
 $N_A$  = Tamanho do Arquivo;
Gera uma população de possíveis soluções  $P = \{x_j\}$  para  $j \in [1, N]$ ;
Gera um arquivo  $A$  sendo vazio;
while O critério de parada não for atingido do
  Copia os indivíduos não dominados de  $P$  para  $A$ ;
   $A \leftarrow A \cup \{x \in P : \nexists y \in \{P, A\} : y \succ x\}$ ;
  Remove indivíduos dominados de  $A$ ;
  while  $|A| > N_A$  do
    | Remove indivíduos que estão muito próximos entre si de  $A$ ;
  end
  while  $|A| > N_A$  do
    | Adiciona o "melhor" resultado de  $P$  não duplicado à  $A$ ;
  end
  Calcula o custo de cada indivíduo em  $P$ ;
  Seleciona geradores de  $A$ ;
  Usa algum método de recombinação para criar "crianças"  $C$  dos geradores;
  Aplica mutação sobre as "crianças" de  $C$  com dada probabilidade;
  Usa algum método para substituir indivíduos de  $P$  por indivíduos de  $C$ ;
end
Passa para a próxima geração;

```

Figura 5: Base para o algoritmo SPEA2 baseado no apresentado por Dan Simon [9]

Na Figura 6 apresentamos outro algoritmo de otimização que possui uma proposta diferente, o MO-PSO, *Multi Objective Particle Swarm Optimization*, proposto por Alvarez-Benitez **et al** [1]. Este algoritmo imita a natureza ao reproduzir o movimento de grupos de animais como pássaros e peixes. Para cada indivíduo, sua posição é alterada através de um vetor velocidade o qual é atualizado a cada etapa do processo levando em consideração o deslocamento de todos os outros indivíduos. Além disto, assim como no deslocamento de pássaros, por exemplo, um vetor, que possui a melhor resposta quando comparado com os

outros, serve como guia para os outros.

```

N = Número de indivíduos;
G = Quantidade de gerações ;
K = Quantidade de dimensões ;
A =  $\emptyset$  - Inicializa um arquivo vazio ;
 $\{x_n, v_n, G_n, P_n\}_{n=1}^N$  - Gera um conjunto de possíveis soluções com suas
respectivas posições e velocidades iniciais;
for  $t=1:G$  do
  for  $n=1:N$  do
    for  $k=1:K$  do
      Atualiza a Velocidade e Posição das partículas baseado em
      funções pré-determinadas;
    end
    Se certifica que os candidatos se encontram dentro da região de
    pesquisa;
    Avalia os pontos  $x_n$  obtidos;
    Adiciona os vetores  $x_n$  não dominados à A;
  end
  Atualiza a melhor posição dos vetores comparando a geração atual
  com a anterior;
  Seleciona o guia global baseado na melhor posição de todos os
  vetores
end

```

Figura 6: Base para o algoritmo MO-PSO baseado no apresentado por Alvarez-Benitez **et al** [1]

A existência de diversos algoritmos, evolutivos ou não, e as diferenças nas suas maneiras de resolver o problema e serem implementados, implicam em diferenças nas suas respostas para um mesmo problema. Para avaliar a resposta de um problema dado por um algoritmo, precisa-se saber a resposta “verdadeira” deste problema para que se possa compará-los. Desta maneira, foram propostos diversos problemas testes, os quais suas respostas são conhecidas. Coello [3] mostra em seu livro diversos destes problemas teste. Dentre eles, podemos citar o conjunto ZDT, composto por 6 problemas cada um com características diferentes. Dentre estes, o ZDT1 possui a Fronteira Ótima contínua e convexa, enquanto no ZDT2 sua Fronteira Ótima não é convexa e ZDT3 possui sua Fronteira Ótima

descontínua.

Desta maneira, conhecida a resposta do problema, podemos avaliar a proximidade da resposta do algoritmo em relação à resposta esperada, como esta aproximação se distribui ao longo da curva esperada e dentre outros.

4 Indicadores de Desempenho

Como discutido anteriormente, com a popularização dos algoritmos evolutivos, a necessidade de comparar suas respostas se tornou algo tão relevante quanto seu desenvolvimento. Para avaliar estas respostas, utilizamos o que chamamos de Indicadores de Desempenho. Estes indicadores também são conhecidos como medidas de desempenho, ou métricas, e tem como objetivo quantificar a qualidade de um dado conjunto, ou comparar dois conjuntos. Utilizaremos o termo Indicadores ao invés de Medidas ou Métricas para evitar confusão com a definição matemática. Para indicadores que comparam dois conjuntos, normalmente são utilizados um conjunto referência e a resposta de um algoritmo. Para isto, denotaremos por FA a *Fronteira Pareto Aproximada* e por FR o *Conjunto Referência* ou *Fronteira Real*. Além disto, estaremos sempre preocupados em avaliar a qualidade de FA e compará-lo com FR. Utilizando a divisão utilizada por Jiang et al.[6], dividiremos os indicadores em 4 grupos:

- Indicadores de Capacidade: Indicadores deste grupo medem o número de vetores em um dado conjunto ou a razão entre a quantidade de elementos de FA e FR.
- Indicadores de Convergência: Avaliam a proximidade entre os conjuntos FA e FR.
- Indicadores de Diversidade: São divididos em dois grupos: 1) Distribuição: mede o quão uniformemente distribuídos estão os pontos de FA, e 2) Cobertura: indica o quanto os vetores em FA se aproximam dos extremos de FR.
- Indicadores de Convergência-Diversidade: Avaliam tanto a convergência quanto a diversidade em uma única escala.

Para evitar confusão com a tradução do nome dos indicadores e facilitar a referência aos livros e artigos utilizados, utilizaremos seus nomes em inglês, porém sempre daremos um significado geométrico do que estes indicadores representam, para facilitar sua compreensão.

Além disto, outra suposição que assumiremos será de que toda Fronteira Aproximada é de fato uma boa aproximação para a Fronteira Real do problema, sendo esta conhecida ou não. Desta forma, evitaremos lidar com situações onde a Fronteira Aproximada contenha vetores que destoem demais dos outros e possam influenciar negativamente a avaliação de tal conjunto.

Todos os indicadores estudados, foram implementados em Matlab e serão disponibilizados gratuitamente. Ao disponibilizar estes códigos, espera-se auxiliar pesquisadores e estudantes que estejam trabalhando em áreas relacionadas poupando-os do trabalho de ter que implementar algum desses indicadores, além de facilitar a aplicação de tais indicadores em possíveis trabalhos futuros.

4.1 Indicadores de Capacidade

Os indicadores de capacidade são aquelas que quantificam o número de vetores em FA ou a relacionam com a quantidade de vetores em FR .

4.1.1 ONVG e ONVGR

Van Veldhuizen [11] define a *ONVG*, do inglês *Overall Non-dominated Vector Generation*, e a *ONVGR*, do inglês, *Overall Non-dominated Vector Generation Ratio*. A primeira, mede a quantidade de vetores em FA , enquanto a segunda toma a razão entre a quantidade de elementos de FA e a de FR .

$$\text{ONVG} := |FA| \quad (2)$$

$$\text{ONVGR} := \frac{|FA|}{|FR|} \quad (3)$$

Schott [8] define indicadores semelhante à *ONVG* e à *ONVGR* porém, ao invés de calculá-la sobre FA , ele a calcula sobre CA . A diferença sendo que múltiplos valores em CA podem ter a mesma imagem sobre F em FA .

Independente de qual conjunto for utilizado, é difícil obter um valor desejado

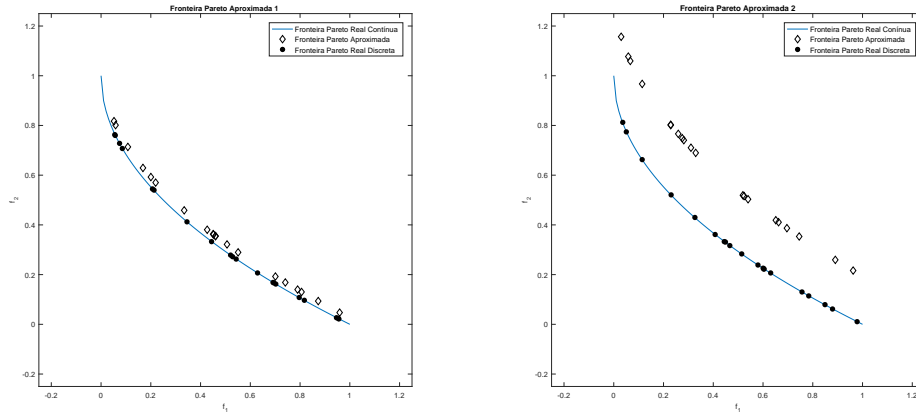


Figura 7: Duas Fronteiras Pareto Aproximadas sendo comparadas com a uma mesma Fronteira Pareto Real

para tais métricas, especialmente devido ao fato de que normalmente nos AE, o usuário determina a quantidade de vetores não dominados que serão obtidos. Da mesma forma, utilizando a razão, a quantidade de vetores em cada conjunto são variáveis que o usuário pode controlar.

Na Figura 7, consideramos duas fronteiras Pareto distintas, com 20 vetores cada, que quando comparadas com a Fronteira Real também com 20 vetores, são avaliadas da mesma maneira por ambos os Indicadores. Porém, a primeira fronteira é claramente preferível sobre a segunda por estar mais próxima da Fronteira Real.

Independente de qual conjunto for utilizado, é difícil obter um valor desejado para tais indicadores, especialmente devido ao fato de que normalmente nos AE o usuário determina a quantidade de vetores não dominados que serão obtidos. Da mesma forma, utilizando a razão, a quantidade de vetores em cada conjunto são variáveis que o usuário pode controlar.

4.1.2 ER

A *ER*, do inglês, *Error Ratio*, também definida por Van Veldhuizen [11] é dada por:

$$E := \frac{\sum_{i=1}^n e_i}{n} \tag{4}$$

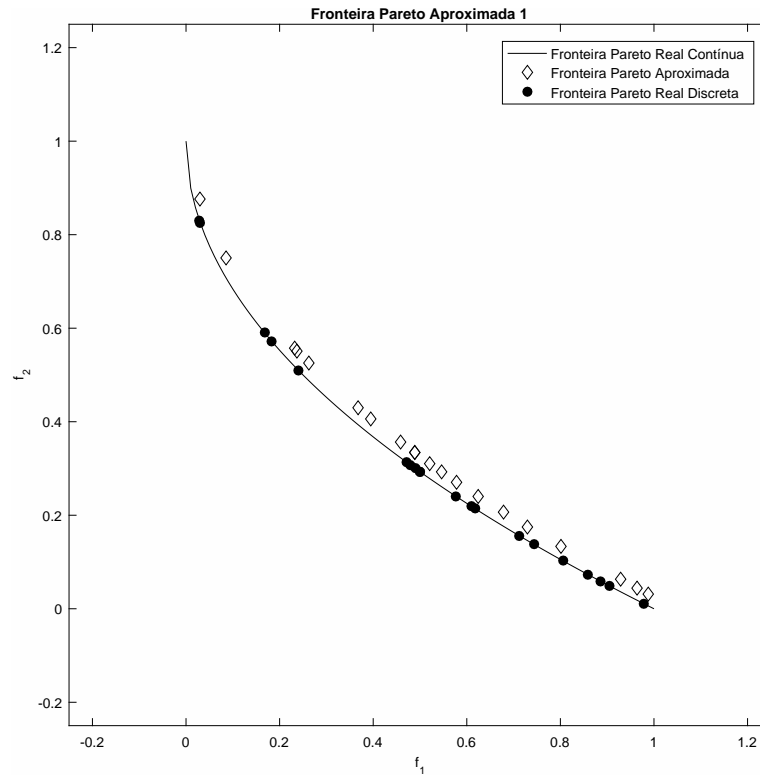


Figura 8: Fronteira Pareto Aproximada

onde n é o número de vetores de FA e

$$e_i = \begin{cases} 0 & , \text{ caso o vetor } i \in FR \\ 1 & , \text{ caso contrário} \end{cases}$$

Este indicador mede a razão entre a quantidade de pontos de FA que também está em FR e a quantidade total de pontos de FR . Caso $FA \subseteq FR$, temos $E = 0$, que seria um valor desejado para tal indicador. O pior caso, seria $E = 1$, o que significaria que $FA \cap FR = \emptyset$. Pode-se pensar que seria bom caso encontrássemos algum ponto de FR , porém, qualquer diferença nas coordenadas dos pontos, mesmo que pequena, é o suficiente para que estes não sejam iguais. Portanto, mesmo obtendo uma aproximação muito boa para FR , pode se ter $E = 1$ e ainda assim ter um bom conjunto.

Porém, não Figura 8 temos $E = 1$ e ainda temos uma aproximação muito boa para FR .

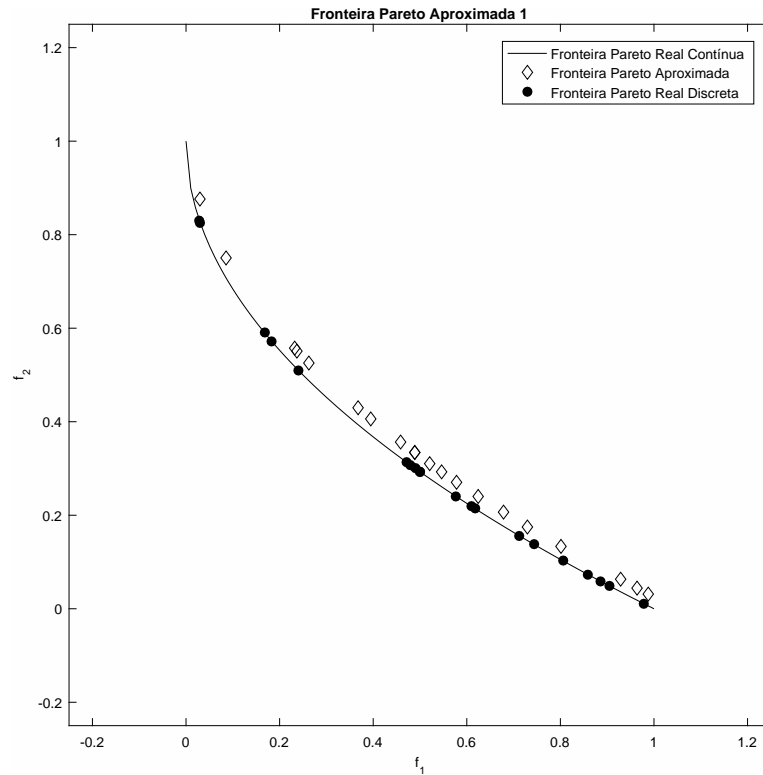


Figura 9: Fronteira Pareto Aproximada

4.1.3 FC

A *FC*, do inglês *Fraction of Pareto Front Covered*, definida por Ulungu et al.[10], considera primeiramente a cardinalidade da interseção de FA e FR e toma a razão deste valor pela cardinalidade de FR .

$$FC := \frac{|FA \cap FR|}{|FR|} \tag{5}$$

É desejado um valor próximo à 1, significando que vários elementos de FA também pertencem à FR . Porém, pode-se obter $FC = 0$, ou seja, termos $FA \cap FR = \emptyset$ mas ainda FA ser uma boa aproximação de FR .

Na Figura 9, temos $FC = 0$, pois $|FA \cap FR| = 0$, porém temos uma boa aproximação.

4.1.4 Conclusão sobre os Indicadores de Capacidade

Todos os indicadores de capacidades são medidas grosseiras, dando pouca informação sobre os conjuntos em questão. Estes, não refletem a distribuição ou convergência do conjunto em questão, além de terem como entrada, a cardinalidade dos conjuntos, que podem ser manipulados pelo usuário.

Assim, características associadas aos indicadores de capacidade, como a necessidade de se conhecer FR ou ter um conjunto de referência, avaliar de maneira igual conjuntos com características diferentes, levando em consideração apenas a quantidade de vetores que este possui, aliado ao fato de que a quantidade de vetores na resposta dos algoritmos pode ser controlada pelo usuário, tornam estes indicadores pouco efetivas na avaliação de respostas de algoritmos.

4.2 Indicadores de Convergência

Indicadores de convergência medem a proximidade do conjunto FA do conjunto FR . Esta proximidade pode ser avaliada de diversas maneiras, como veremos a seguir. Esta proximidade é calculada avaliando a distância Euclideana entre os pontos dos dois conjuntos e operando com os valores obtidos

Percebemos que estes indicadores sofrem de um mesmo problema, a dificuldade de avaliar a distribuição dos conjuntos em questão, que será discutido com mais detalhe ao final desta seção.

4.2.1 GD e IGD

Em [11], Van Veldhuizen propões a GD é definida, do inglês *Generational Distance*. Esta medida avalia a proximidade entre FA e FR calculando a menor distância entre cada vetor de FA e o conjunto FR , e tomando a média destas distâncias.

$$GD := \frac{\left(\sum_{i=1}^n d_i^2 \right)^{\frac{1}{2}}}{n} \quad (6)$$

onde n é o número de elementos FA e $d_i = \min_j \{|x_i - y_j|\}$ onde $x_i \in FA$ e $y_j \in FR$, ou seja, d_i é a distância entre o ponto $x_i \in FA$ e o ponto de FR mais próximo.

Caso $GD = 0$, podemos concluir que cada ponto de FA é igual a um ponto em FR ($d_i = 0 \quad \forall i$), portanto $FA \subseteq FR$. Note que a igualdade só acontecerá

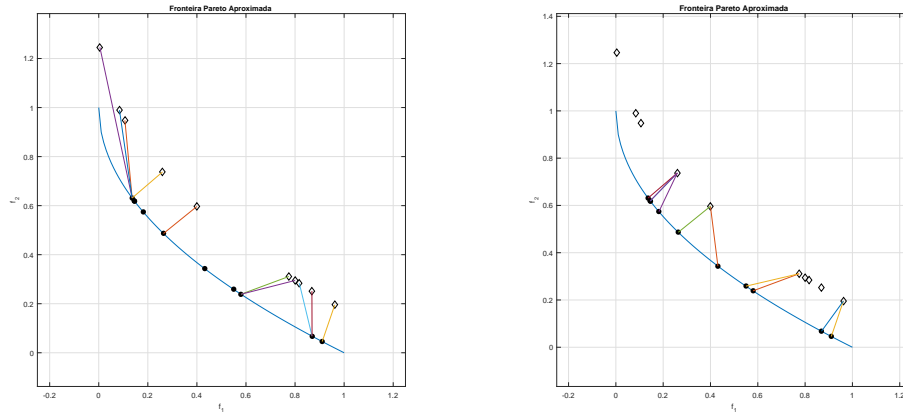


Figura 10: Distâncias levadas em consideração ao calcular GD e IGD, respectivamente

se $|FA| = |FR|$. Com base nisto, deseja-se encontrar um valor próximo a zero para esta medida, o que significa que na média, os elementos de FR estão próximos aos de FA .

De maneira bem similar, temos definida a *IGD* (FALTA REFERENCIA), *Inverted Generational Distance* que leva em consideração as distâncias mínimas dos vetores de FR aos vetores de FA .

$$IGD := \frac{\left(\sum_{i=1}^n d_i^2 \right)^{\frac{1}{2}}}{n^*} \tag{7}$$

neste caso, n^* é o número de elementos FR e $d_i = \min_j \{|y_j - x_i|\}$ onde $x_i \in FA$ e $y_j \in FR$.

A diferença entre a GD e a IGD é a quantidade de elementos levadas em consideração. Enquanto a GD leva em consideração a quantidade de elementos em FA , a IGD leva a quantidade de elementos de FR .

Como podemos ver na Figura 10, mesmo calculando as distâncias em relação aos mesmos conjuntos, os pontos de partida sendo diferentes, levam a respostas diferentes. Para a aproximação em questão, temos $GD = 0.2647$ e $IGD = 0.1860$.

4.2.2 Distance from Reference Set

Jaszkiewicz e Cyzak [5] propuseram dois indicadores para medir a distância entre dois conjuntos. Eles levam em consideração a variação que ocorre em cada objetivo, desta maneira, eliminando possíveis interferências que um objetivo possa ter no resultado por assumir valores em um intervalo mais extenso do que outro objetivo. Assim, definimos a distância entre dois pontos por:

$$c(x, y) = \max_{i=1, \dots, n} \{0, w_i(f_i(y) - f_i(x))\}$$

onde $x \in FR$ e $y \in FA$, $w_i = \frac{1}{\Delta_i}$ com Δ_i sendo a variação no objetivo f_i . Assim, $c(x, y) = 0$ somente se $f_i(y) - f_i(x) \geq 0$ para todo i . Caso contrário, $0 < c(x, y) \leq 1$. Esta função c calcula a distância entre dois vetores x e y considerando apenas o máximo da diferença em cada objetivo, e não a distância Euclideana como nos outros exemplos.

Utilizando isto, temos os seguintes indicadores:

$$Dist_1 = \frac{1}{|FR|} \sum_{y \in FA} \min_{x \in FR} \{c(x, y)\} \tag{8}$$

$$Dist_2 = \max_{x \in FR} \{\min_{y \in FA} \{c(x, y)\}\} \tag{9}$$

Desta forma, $Dist_1$ mede a distância mínima média entre os vetores de FA e FR utilizando a função $c(x, y)$. Enquanto $Dist_2$ nos dá o maior valor destas distâncias mínimas, nos dando uma cota superior para estas distâncias mínimas.

4.2.3 ME

Definida por Van Veldhuizen [11], ME , *Maximum Pareto Front Error*, mede a maior dentre as menores distâncias de pontos em FA aos pontos de FR . Geometricamente, este valor nos dá uma cota superior para a "largura" de uma faixa ao redor de FA tal que para todo ponto, existirá um ponto de FR dentro desta faixa. Matematicamente:

$$ME = \max_{y \in FR} \min_{x \in FA} \sqrt{\sum_{i=1}^n |f_i(x) - f_i(y)|^2} \tag{10}$$

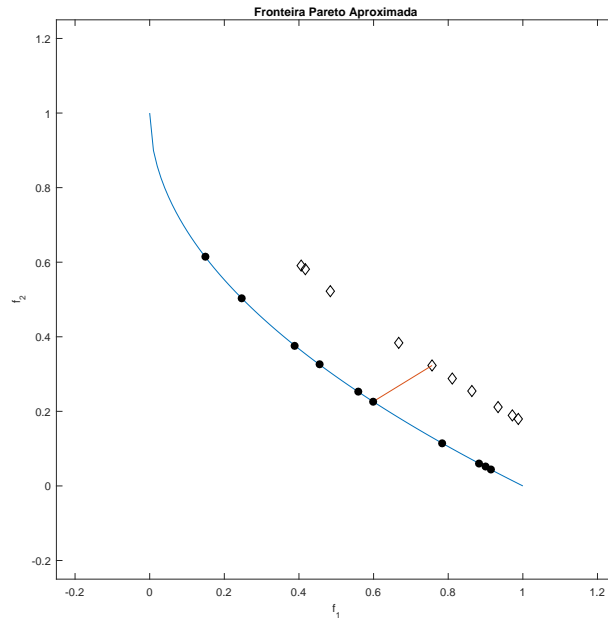


Figura 11: Maior dentre as distâncias mínimas entres os vetores de FA e FR

Da mesma maneira, Sayin em [7] define o indicador ε do inglês *Coverage Error* como sendo

$$\varepsilon = \max_{y \in FR} \min_{x \in FA} d(x, y) \tag{11}$$

A qual tem a mesma ideia em tomar o máximo das distâncias mínimas, porém leva em consideração qualquer distância, não necessariamente só a Euclideana.

Na Figura 11 vemos destacada a maior dentre as menores distâncias entre os vetores de FA e FR . Esta então, é uma cota superior para a distância entre os conjuntos FA e FR .

4.2.4 ε -indicator

Zitzler **et. al** [14] define o ε -indicator como:

$$I_\varepsilon(FA) = \inf_{\varepsilon \in \mathbb{R}} \{ \forall x \in FR, \exists x^* \in FA : x^* \succ_\varepsilon x \} \tag{12}$$

Onde $x^* \succ_\varepsilon x$ significa que $\forall i = 1, 2, \dots, n$, temos $(x_i)^* \leq \varepsilon x_i$. Ou seja, o

ε – *indicator* nos dá o menor fator pelo qual precisamos multiplicar todas as coordenadas de todos os pontos de FA para que estes dominem os pontos de FR .

4.2.5 Indicador χ^2

Este indicador, definido por Deb [4], utiliza um parâmetro ε para calcular n_i , o numero de vetores de FA que estão no máximo ε distantes dos vetores de FR . A partir disto, a diferença entre dois conjuntos e dada por:

$$i = \sqrt{\sum_{i=1}^{|FR|+1} \left(\frac{n_i - \bar{n}_i}{\sigma^2} \right)^2} \tag{13}$$

É comum utilizar FR com distribuição normal. Desta forma, espera-se ter $n_i = \frac{|FA|}{|FR|}$ vetores localizados próximos de cada vetor de FR . Além disto, o parâmetro $\sigma_i^2 = \bar{n}_i \left(1 - \frac{\bar{n}_i}{|FA|} \right)$ é sugerido para $i = 1, 2, \dots, |FR|$. Já para o índice $i = |FR| + 1$, que representa os vetores que não pertencem a nenhuma ε -vizinhança dos vetores de FR , utiliza-se os seguintes valores para o índice $|FR| + 1$: $\bar{n}_i = 0$ e $\sigma_i^2 = \sum_{i=1}^{|FR|} \sigma_i^2$.

Na Figura 12 utilizando $\varepsilon = 0.05$, temos $n_1 = 1, n_2 = 1, n_3 = 2, n_4 = 1, n_5 = 1, n_6 = 3$. Além disto, $\bar{n}_i = \frac{10}{5} = 2$ e $\sigma_i^2 = 1.6$ para $i = 1, \dots, 5$ enquanto obtemos $n_i = 0$ e $\sigma_i^2 = 8$ para $i = 6$. Assim, $i = 1.3463$.

Como este indicador calcula o "desvio"entre dois conjuntos, espera-se um valor próximo de zero, enquanto se for obtido uma aproximação FA que possui a mesma distribuição de FR , significando $n_i = \bar{n}_i$ para todo i , teremos exatamente o indicador igual a zero.

Apesar de dar uma boa ideia da distribuição do vetores de FA em relação aos vetores de FR , este indicador pode levar a conclusões equivocadas dependendo do valor de ε utilizador pelo usuário.

4.2.6 Conclusões sobre Indicadores de Convergência

Ao considerar apenas a distância entre os vetores dos dois conjuntos, seja lá como esta distância é calculada, os indicadores de convergência não consideram

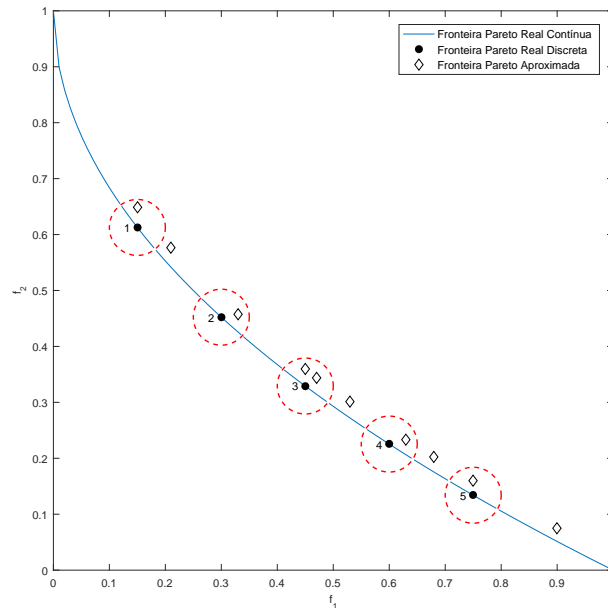


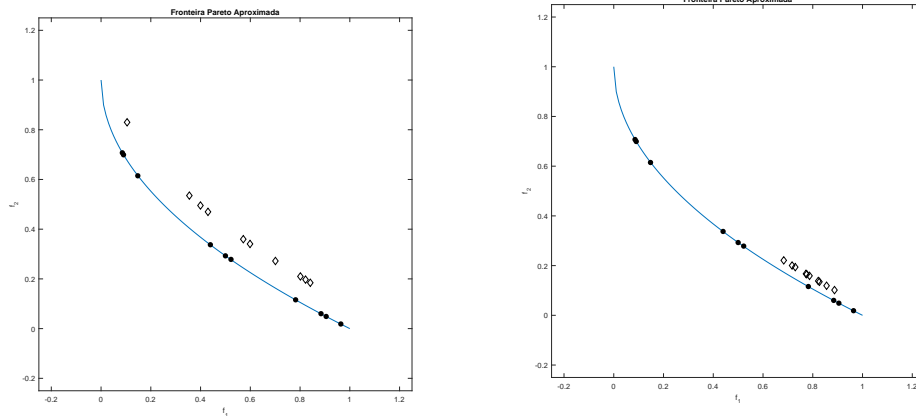
Figura 12: Exemplo de Fronteira Pareto Aproximada e $\varepsilon = 0.05$

como os vetores de FA estão distribuídos pelo espaço. Além disto, a necessidade de se conhecer a fronteira real, ou ter um conjunto referência torna estes indicadores difíceis de serem utilizados em problemas reais.

Na Figura 13 podemos ver duas Fronteiras Aproximadas sendo comparada com a mesma Fronteira Real, e mesmo que a primeira esteja "mais distante" da fronteira real, esta é preferível por nos dar informação sobre uma extensão maior do conjunto real. Porém, ao avaliá-las com o indicador GD , por exemplo, obtemos $GD = 0.0924$ para o primeiro conjunto e $GD = 0.0402$, para o segundo e este é um problema que se repete para as outras medidas. Para a $Dist_1$ obtivemos 0.1803 e 0.2066 para o primeiro e segundo conjunto respectivamente, e $Dist_2$ 0.3397 e 0.4337. Para ME obtivemos 0.3863 e 0.0398 respectivamente.

4.3 Indicadores de Diversidade

Os indicadores de diversidade medem a distribuição dos vetores de FA dentro do próprio conjunto, ou seja, avaliam as distância dos vetores do mesmo conjunto,



(a) Fronteira Pareto Aproximada com boa Distribuição (b) Fronteira Pareto Aproximada com distribuição ruim

Figura 13: Duas Fronteiras Pareto Aproximadas sendo comparadas com a uma mesma Fronteira Pareto Real

Estes indicadores, como o próprio nome sugere, avaliam o quão bem o conjunto em questão cobre uma determinada região do espaço, no sentido de que eles delimitam uma região na qual todos os pontos estão contidos ou então calculam a área determinada por este conjunto quando limitado por uma certa área.

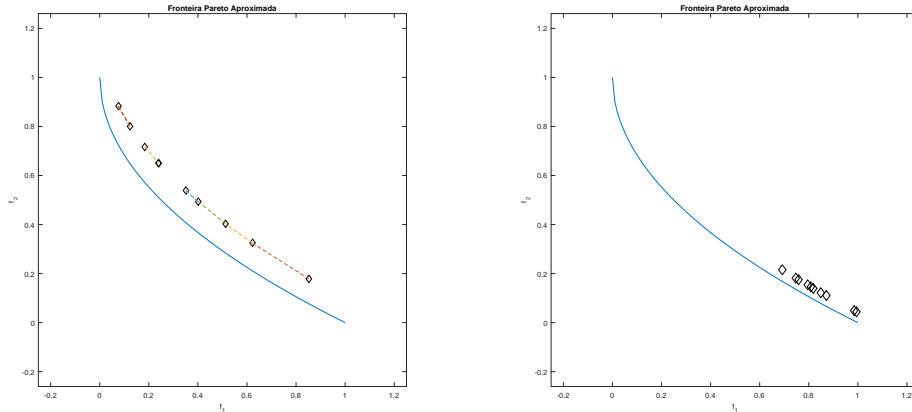
Estes indicadores são os mais completos, avaliando tanto a distribuição quando a cobertura de um conjunto

4.3.1 Maximum Spread

Zitzler [13] propõe um indicador que leva em consideração o máximo dentre as distâncias entre os pontos de FA .

$$M_3^* = \sqrt{\sum_{i=1}^n \max\{|u_i - v_i| : u, v \in FA\}} \tag{14}$$

Assim, este indicador leva em consideração as distâncias em cada objetivo para estimar a extensão do conjunto FA no espaço de soluções. Quando avaliando dois conjuntos S_1, S_2 , a priori, se $M_3^*(S_1) > M_3^*(S_2)$, então S_1 é preferível. Porém, este indicador, por si só não deve ser tomado como resposta definitiva.



(a) Fronteira Pareto Aproximada bem distribuída (b) Fronteira Pareto Aproximada acumulada

Figura 14: Duas Fronteiras Pareto Aproximadas sendo avaliadas

Por não levar em consideração nenhum conjunto de referência (FR), esta medida não nos dá informação sobre a convergência dos conjuntos em questão, conseqüentemente, pode levar a conclusões equivocadas.

4.3.2 Spacing

Van Veldhuizen [11], propõe um Indicador que leva em consideração as menores distâncias entre os pontos de FA aos outros pontos de FA e calcula a variância destes valores.

$$S = \sqrt{\frac{1}{n_1} \sum_{i=1}^{|n|} (\bar{d} - d_i)^2} \tag{15}$$

onde $d_i = \min_j \{|x_i - y_j|\}$, $x_i, y_j \in FA$ e \bar{d} é a média os d_i .

Note que se todos os pontos estiverem equidistantes do ponto mais próximo, ou seja $d_i = d \forall i$, $\bar{d} - d_i = 0$, teremos $S = 0$. Qualquer valor diferente de 0 indica que a distribuição dos pontos varia, além disto, quanto maior o valor de S , maior a variação nas distâncias entre os pontos de FA.

Podemos ver na Figura 14 temos dois conjuntos no qual o primeiro obtivemos $S = 0.0778$ e o segundo $S = 0.0162$, porém podemos ver que mesmo obtendo um valor menos no indicador, o segundo conjunto, por ter todos os seus pontos

acumulados em uma pequena região, o primeiro conjunto é preferível sobre o segundo.

4.3.3 I_{HC} e I_O

Também definido por Zitzler **et. al** [14], o *Enclosing Hypercube Indicator* avalia o menor Hiper cubo contendo todos os pontos de FA . Está é uma medida binária, ou seja, tem dois valores a serem levados em consideração.

$$I_1^{HC}(FA) = \sup_{a \in \mathbb{R}} \{ \{(a, a, \dots, a)\} \triangleright FA \}$$

$$I_2^{HC}(FA) = \inf_{b \in \mathbb{R}} \{ \{(b, b, \dots, b)\} \triangleleft FA \}$$

onde $(a, a, \dots, a) \triangleright FA$ significa que todo $x \in FA$ é dominado por (a, a, \dots, a) Assim,

$$I_{HC}(FA) = (I_1^{HC}(FA), I_2^{HC}(FA)) \tag{16}$$

Os mesmos autores do indicador anterior, propuseram outro indicador muito semelhante, *Objective Vector Indicator* dado por:

$$I_i^O(FA) = \inf a \in \mathbb{R} \{ \forall (x_1, x_2, \dots, x_n) \in FA : x_i \leq a \} \tag{17}$$

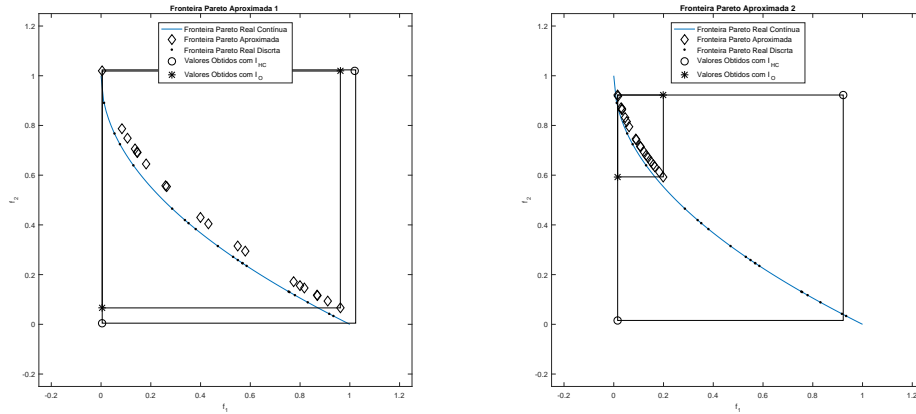
para todo $1 \leq i \leq n$ e

$$I_{n+1}^O(FA) = \begin{cases} 0, & \text{se FA contém dois ou mais elementos} \\ 1, & \text{caso contrário} \end{cases}$$

A ideia neste caso é considerar o menor hiper retângulo que cubra todo FA . Esta medida nos dá um ponto o qual é dominado por todos os pontos em FA .

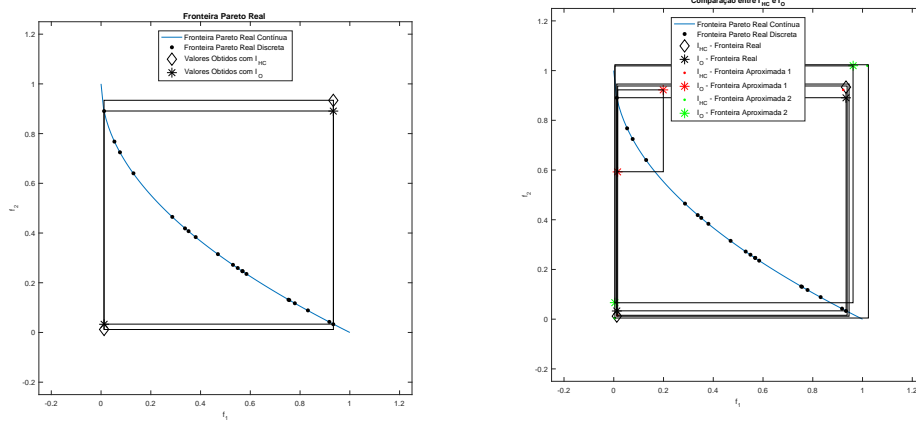
Estas duas, por si só não revelam muito sobre o conjunto em questão. Elas são melhor utilizadas quando comparam dois conjuntos. Dado um conjunto referência e duas aproximações, procura-se dentre as duas, a que define um hiper retângulo mais “semelhante” ao do conjunto referência.

Nas Figura 15a,15b, podemos ver duas Fronteiras Pareto aproximadas e seus valores para os indicadores I_{HC} e I_O e os hiper cubos definidos por tais valores, enquanto na Figura 15c temos uma Fronteira Pareto Ótima e I_{HC} , I_O e os hiper cubos definidos por tais valores.



(a) Fronteira Pareto Aproximada bem distribuída

(b) Fronteira Pareto Aproximada acumulada



(c) Fronteira Pareto Real

(d) Fronteira Pareto Real com todos os valores obtidos para I_{HC} e I_O

Figura 15: Avaliação dos indicadores I_{HC} e I_O

Podemos ver na Figura 15d a Fronteira Ótima e todos os hipercubos obtidos por I_{HC} e I_O para as duas fronteiras aproximadas obtidas em 15a e 15c. Podemos ver, por exemplo que I_{HC} pode levar a conclusões equivocadas, pois neste caso, para os três conjuntos ela retorna resultados muito próximos, o que faz com que os três hipercubos sejam bem semelhantes. Enquanto I_O retorna valores que mostram melhor o fato, por exemplo, de que um dos conjuntos está acumulado em uma pequena região do espaço, como podemos ver, o hipercubo definido pela resposta de I_O em 15b é bem menor que os outros dois.

4.3.4 Overall Pareto Spread

Wu e Azarm [12] utilizam dois pontos pré-determinados, que chamaremos de *Ponto Bom* e *Ponto Ruim* para definir o indicador *Overall Pareto Spread*. Diremos que um ponto P_1 , é bom para um dado conjunto S se for uma cota inferior para todos os pontos de (Ou seja $P_{1i} \leq x_i$ para todo $x \in S$), e um ponto P_2 será ruim se for uma cota superior para todos pontos do espaço(Ou seja $P_{2i} \geq x_i$ para todo $x \in S$).

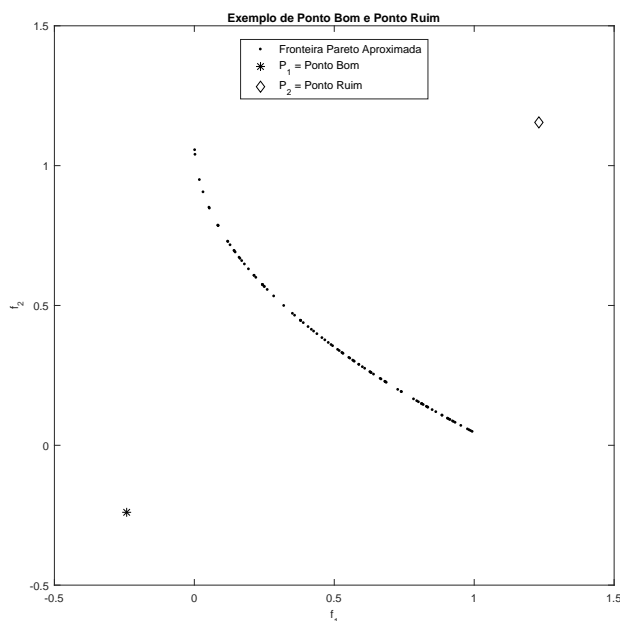


Figura 16: Uma Fronteira Pareto Aproximada e exemplos de Pontos Bom e Ruim.

Na Figura 16 temos P_1 como ponto bom e P_2 como ponto ruim, uma vez que P_1 domina todos os pontos de FA e P_2 é dominado por todos os pontos.

Além destes dois pontos, consideramos também outros dois pontos, estes sendo o “Mínimo” e o “Máximo” do conjunto em questão. Denotaremos por Min o ponto do conjunto FA tal que $Min_i \leq x_i$ para todo $x \in FA$ e $i = 1, \dots, n$ e Max o ponto tal que $Max_i \geq x_i$ para todo $x \in FA$ e $i = 1, \dots, n$. Note que os pontos Min e Max dependem do conjunto em questão, enquanto P_1 e P_2 são controlados pelo usuário.

Assim, a *Overall Pareto Spread*, é dado por:

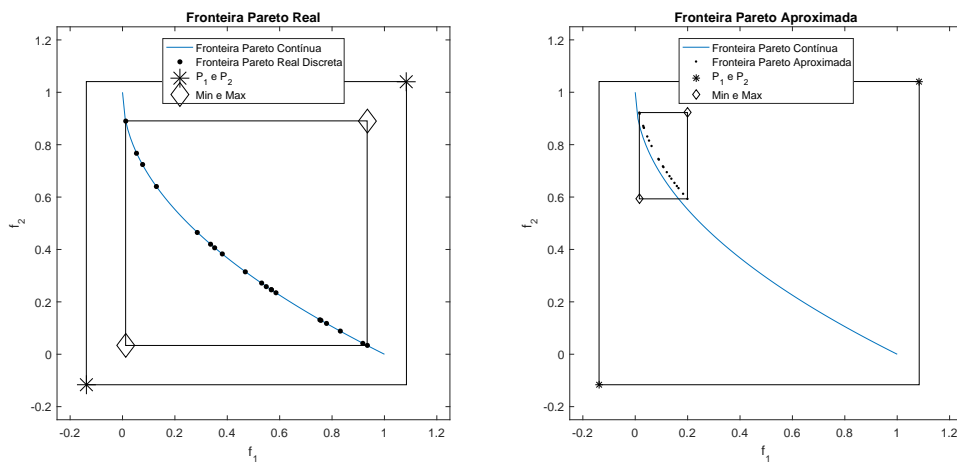
$$OS(FA) = \frac{HR_{ex}(FA)}{HR_{br}} \tag{18}$$

onde $HR_{ex}(FA)$ é o volume do hiper-retângulo formado pelos pontos Min e Max e HR_{br} é o volume do hiper-retângulo formado pelos pontos P_1 e P_2 .

Esta medida também pode ser escrita como:

$$OS(FA) = \frac{\prod_{i=1}^n |\max_{x \in FA}(x_i) - \min_{x \in FA}(x_i)|}{\prod_{i=1}^n |P_{2i} - P_{1i}|}$$

Quando comparando duas soluções, procuramos por aquela cuja OS seja mais próxima de 1, assim, representando que os pontos cobrem uma parte maior do espaço de soluções, dados os pontos Bom e Ruim em questão. Na Figura 17, podemos observar os hipercubos considerados para uma fronteira Pareto real e uma aproximada. Neste caso foram utilizados os mesmos P_1 e P_2 .



(a) Fronteira Pareto Real

(b) Fronteira Pareto Aproximada Acumulada

Figura 17: Hipercubos utilizados para avaliar o indicador OS

Utilizando o exemplo da Figura 17, temos que $HR_{ex}(FA) = 0.0605$ e $HR_{ex}(FR) = 0.7906$ enquanto $HR_{br} = 1.4144$ (Considerando $P_1 =$

$(-0.1381, -0.1164)$ e $P_2 = (1.0840, 1.0409)$. Assim, $OS(FA) = 0.0427$ e $OS(FR) = 0.5589$. Desta maneira, podemos ver que o fato de FA estar acumulado faz com que o valor de OS diminua consideravelmente.

4.3.5 HV e HVR

O Indicador *Hypervolume* leva em consideração o volume compreendido entre os pontos da Fronteira Pareto e um dado ponto de referência.

$$HV(FA) := \{\cup_i area_i : x_i \in FA\} \tag{19}$$

onde $x_i \in FA$ e $area_i$ é a área entre x_i e o ponto referência.

Enquanto o *Hypervolume Ratio* leva em consideração o quociente $HV(FA)$ por $HV(FR)$ onde FR é uma Fronteira Pareto Ótima.

$$HVR(FA) := \frac{HV(FA)}{HV(FR)} \tag{20}$$

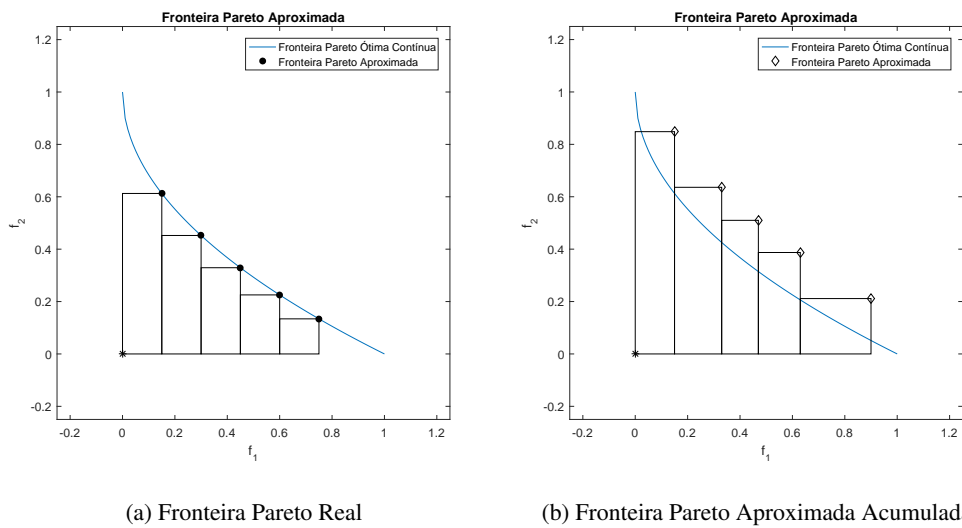


Figura 18: Áreas utilizadas para calcular os indicadores HV e HVR

Na Figura 18a temos um conjunto cujo $HV(FR) = 0.263$ enquanto na Figura 18b, $HV(FA) = 0.4322$, o que nos leva a $HVR(FA) = 1.643$. Em ambos os casos, foi levado em consideração $(0, 0)$ como ponto de referência.

O Indicador HV por si só não oferece base para avaliar um conjunto, assim, é ideal conhecer um conjunto referência (Fronteira Ótima) e avaliar HVR . Ao se tratar de problema de minimização, $HVR > 1$ indica que FA não é tão boa quanto FR , enquanto em problemas de maximização $HVR < 1$ leva à mesma conclusão.

A principal dificuldade deste indiciador é seu custo computacional, ou seja, quantas operações são necessárias para calculá-lo, uma vez que para problemas com várias variáveis, as contas para calculo de cada volume cresce consideravelmente.

4.3.6 Conclusões sobre Indicadores de Divergência

Estes indicadores priorizam avaliar a distribuição dos vetores do conjunto em questão através do espaço. Enquanto ME consegue avaliar o quanto estes vetores estão espalhados pelo espaço mas falha em avaliar como estes vetores estão distribuídos pelo espaço, a *Spacing* sofre do problema inverso, ao avaliar a distribuição dos vetores mas sem informar nada sobre seu espalhamento. Os outros indicadores fornecem uma ideia melhor sobre o conjunto, principalmente quando comparado com um conjunto referência. Como fato interessante, podemos citar que mesmo os indicadores HV e HVR fornecendo valores exatos para o hiper-volume compreendido por FA , este vem com um custo computacional muito alto. Enquanto o indicador OS fornece informações relevantes o suficiente enquanto possui custo computacional muito menor. Desta maneira, assim mesmo estes indicadores, I_O , I_{HC} , OS , HV e HVR sendo os mais completos em relação à avaliar conjuntos, é sensato utilizá-los em conjunto com outros indicadores.

5 Conclusão

O estudo dos indicadores, me proporcionou uma oportunidade de começa a estudar otimização sem me prender aos detalhes dos algoritmos, detalhes estes que para um aluno sem uma boa base de programação seria difícil. Assim, ao realizar o estudo inicial, eu me familiarizei primeiro com o problema em si, e em maneiras de avaliar as respostas dadas por algoritmos que inevitavelmente eu me deparei ao decorrer dos estudos. Neste estudos os algoritmos foram mais um passo intermediário do que propriamente onde foi investido mais tempo.

Desta forma, estou preparado para futuros estudos que foquem nos algoritmos de otimização em si, por exemplo, além de me familiarizar com o programa Matlab e técnicas de programação.

Após a extensa revisão de bibliografia feita, pode-se chegar a conclusão que existem muitos empecilhos ao se propor um indicador para avaliar todas as características de um conjunto. Esta dificuldade surge na necessidade de avaliar conjuntos com muitos elementos, pois a quantidade de operações a serem feitas aumenta consideravelmente dependendo do indicador em consideração.

Outra dificuldade apontada com bastante detalhes por Zitzler et. al [14] é a incompatibilidade de certos indicadores com o conceito de dominância. Isto significa que em diversos casos, os indicadores não refletem qual conjunto é melhor que outro. Discutimos alguns destes casos ao decorrer do trabalho.

Desta maneira, o ideal para compreender bem o resultado do algoritmo é utilizar um conjunto de indicadores. Desta maneira, reduz-se o custo computacional de calcular um indicador mais robusto, mas ainda é possível obter uma boa noção das características do conjunto.

Este estudo se faz relevante, pois a implementação destes indicadores em Matlab irá auxiliar pessoas interessadas em seus estudos na área, fornecendo os códigos para utilização, desta maneira economizando tempo e trabalho para estas pessoas. Estes códigos serão disponibilizados na página do Professor Dr. Thiago Santos (<http://professor.ufop.br/santostf/home>).

Referências

- [1] Julio Alvarez-Benitez, Richard Everson, and Jonathan Fieldsend. A mopso algorithm based exclusively on pareto dominance concepts. In *Evolutionary Multi-Criterion Optimization*, pages 459–473. Springer, 2005.
- [2] Carlos A Coello Coello and Gary B Lamont. *Applications of multi-objective evolutionary algorithms*, volume 1. World Scientific, 2004.
- [3] Carlos A Coello Coello, Gary B Lamont, and David A Van Veldhuizen. *Evolutionary algorithms for solving multi-objective problems*, 2007.
- [4] Kalyanmoy Deb. *Multi-objective optimization using evolutionary algorithms*. Wiley, 2005.

- [5] A Jaszkievicz and P Czyzak. Pareto simulated annealing—a metaheuristic technique for multiple-objective combinatorial optimization. *J. Multi-Criteria Decis. Anal.*, 7(1):34–47, 1998.
- [6] Siwei Jiang, Yew-Soon Ong, Jie Zhang, and Liang Feng. Consistencies and contradictions of performance metrics in multiobjective optimization. *IEEE transactions on cybernetics*, 44(12):2391–2404, 2014.
- [7] Serpil Sayın. Measuring the quality of discrete representations of efficient sets in multiple objective mathematical programming. *Mathematical Programming*, 87(3):543–560, 2000.
- [8] Jason R Schott. Fault tolerant design using single and multicriteria genetic algorithm optimization. Technical report, DTIC Document, 1995.
- [9] Dan Simon. *Evolutionary optimization algorithms*. John Wiley & Sons, 2013.
- [10] EL Ulungu, JFPH Teghem, PH Fortemps, and D Tuytens. Mosa method: a tool for solving multiobjective combinatorial optimization problems. *Journal of multicriteria decision analysis*, 8(4):221, 1999.
- [11] David A Van Veldhuizen. Multiobjective evolutionary algorithms: classifications, analyses, and new innovations. Technical report, DTIC Document, 1999.
- [12] Jin Wu and Shapour Azarm. Metrics for quality assessment of a multiobjective design optimization solution set. *Journal of Mechanical Design*, 123(1):18–25, 2001.
- [13] Eckart Zitzler. *Evolutionary algorithms for multiobjective optimization: Methods and applications*. 1999.
- [14] Eckart Zitzler, Lothar Thiele, Marco Laumanns, Carlos M Fonseca, and Viviane Grunert Da Fonseca. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on evolutionary computation*, 7(2):117–132, 2003.

- [15] Eckart Zitzler, Marco Laumanns, and Lothar Thiele. Spea2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. *Evolutionary Methods for Design, Optimization, and Control*, pages 95–100, 2002.