RESOLUÇÃO DE PROBLEMAS COMPLEXOS DE ENGENHARIA UTILIZANDO ALGORITMOS EVOLUTIVOS MULTIOBJETIVO

António Gaspar-Cunha
Institute for Polymers and Composites/I3N
University of Minho, Campus de Azurém, 4800-058 Guimarães, Portugal
agc@dep.uminho.pt

José A. Covas
Institute for Polymers and Composites/I3N
University of Minho, Campus de Azurém, 4800-058 Guimarães, Portugal
agc@dep.uminho.pt

RESUMO

Os problemas práticos de projeto e otimização em engenharia são normalmente complexos, multidisciplinares e difíceis de resolver em períodos razoáveis de tempo. Nalguns casos, podem ser simulados matematicamente utilizando ferramentas computacionais sofisticadas, que geralmente requerem recursos significativos. Os Algoritmos Evolutivos são particularmente adequados para lidar com a natureza multiobjetivo dos problemas reais, uma vez que trabalham com uma população (de vetores ou soluções) em vez de com uma única solução. Tal permite a geração de fronteiras de Pareto que representam o balanço entre os objetivos proporcionando, simultaneamente, uma ligação às variáveis de decisão. Este trabalho apresenta e discute uma abordagem possível para a resolução de problemas complexos de engenharia, utilizando ferramentas capazes de lidar com objetivos múltiplos, a tomada de decisão, a robustez das soluções e a redução do tempo de cálculo.

PALAVARAS CHAVE. Projeto e otimização multidisciplinar, Projeto em engenharia, Otimização multiobjectivo, Algoritmos evolutivos.

ABSTRACT

Real engineering design and optimization problems are complex, multidisciplinary and difficult to manage within reasonable timings. In some cases, they can, at least to some extent, be mathematically described by sophisticated computational tools, which generally require significant resources. Evolutionary Algorithms, EAs, are particularly adequate to deal with the multi-objective nature of real problems, as they work with a population (of vectors, or solutions) rather than with a single solution point. This feature enables the generation of Pareto frontiers representing the trade-off between the objectives, while simultaneously providing a link to the decision variables. This work presents and discusses a possible approach to solve such type of problems by employing tools that are able to deal with multiple objectives, decision making, robustness of the solutions and reduction of computation times.

KEYWORDS. Multidisciplinary design and optimization, Engineering design, Multi-Objective Optimization, Evolutionary Algorithms.

1. Introdução

Os problemas reais de projeto e otimização em engenharia são geralmente complexos, multidisciplinares e difíceis de resolver em prazos razoáveis. Nalguns casos, estes problemas podem, pelo menos parcialmente, ser descritos matematicamente por ferramentas computacionais sofisticadas, que geralmente exigem recursos significativos. De facto, o avanço científico e tecnológico nalgumas disciplinas (como por exemplo, a mecânica de fluidos e a mecânica estrutural), conjugado com o desenvolvimento de técnicas de computação de alto desempenho (nomeadamente, a computação paralela e/ou em rede) e a disponibilidade de instalações computacionais mais eficientes, tem vindo a tornar possível abordar os problemas complexos de forma mais completa. Um sistema Multidisciplinar de Projeto e Otimização, MDO, pode ser descrito como uma tecnologia, um ambiente ou uma metodologia usada na conceção integrada de sistemas complexos de engenharia, através da combinação de diferentes disciplinas e tendo em consideração de forma sinérgica a interação entre os vários subsistemas (Avriel (1979), Siddall (1982)). Como exemplos de aplicação prática, podem citar-se o design, projeto e otimização de aviões, automóveis, estruturas de edifícios e sistemas de fabrico (Bentley (1999), Oztemela (2009)).

Têm sido propostas na literatura várias estratégias de MDO. Uma característica comum desses métodos é a utilização de técnicas de aproximação e decomposição, que consistem em dividir o problema em sistemas mais pequenos e, portanto, mais passíveis de serem resolvidos isoladamente aplicando modelos mais simples (Bentley (1999), Avriel (1979). No entanto, perde-se a visão unificada do problema e, provavelmente, o custo total será substancialmente superior ao correspondente a resolver o problema no seu todo. Mais recentemente, têm sido propostas abordagens integradas de MDO (Siddall (1982), Oztemela (2009)). Infelizmente, algumas dessas técnicas parecem ignorar o fato de os problemas reais conterem vários objetivos, que são frequentemente contraditórios, mas que devem ser tidos em conta simultaneamente (Deb (2001)).

Os Algoritmos Evolutivos, EAs, são particularmente adequados para lidar com a natureza multiobjetivo dos problemas reais, devido ao fato de usarem uma população de vetores ou soluções em vez de um único ponto ou solução. Esta característica permite a geração de fronteiras de Pareto que representam o compromisso existente entre os diversos objetivos, proporcionando simultaneamente uma ligação às variáveis de decisão (Deb (2001), Branke (2008), Gaspar-Cunha (2004a), García-Martínez (2007)). Assim, o resultado de um algoritmo evolutivo multi-objetivo (*Multi-Objective Evolutionary Algorithm*, MOEA) será um conjunto de soluções tão próximas quanto possível da frente ótima de Pareto (Deb (2001)). Foram desenvolvidos outros tipos de algoritmos multi-objetivo que podem ser usados em situações específicas, como por exemplo os algoritmos de colónia de formigas (*Ant Colony Optimization*, ACO) (García-Martínez (2007)), ou a busca local estocástica (*Stocastic Local Search*, SLS) (Paquete (2007)).

Em todos os casos, será sempre necessário dispor de informação sobre a importância relativa de cada objetivo no problema. Tal é frequentemente realizado através da introdução no sistema de otimização das preferências de um tomador de decisão (*Decision Maker*, DM) (Ferreira (2008)). Dependendo da estratégia adotada para a tomada de decisão, essas informações podem ser introduzidas antes, durante ou após a otimização (Branke (2008), Gaspar-Cunha (2004b), García-Martínez (2007), Ferreira (2008)). Além disso, uma vez que em aplicações reais podem frequentemente ocorrer pequenas instabilidades das variáveis de decisão, ou de parâmetros ambientais (por exemplo, temperatura), deve procurar assegurar-se que o desempenho das soluções potenciais é apenas ligeiramente afetado por essas variações, ou seja, as soluções devem ser robustas (Ray (2002), Jin (2005), Gaspar-Cunha (2008), Ferreira (2008)). Apesar da sua importância prática, a robustez é raramente incluída nos algoritmos tradicionais (Jin (2005), Gaspar-Cunha (2008), Ferreira (2008)).

Uma das principais dificuldades associadas à aplicação de MOEAs a problemas reais de engenharia é o elevado número de avaliações da função objetivo necessário à obtenção de uma solução aceitável - normalmente, da ordem de vários milhares. Além disso, estas avaliações são com frequência computacionalmente caras, uma vez que usam códigos baseados em métodos numéricos, tais como elementos finitos ou volumes finitos. Por conseguinte, a redução do número de avaliações necessárias para resolver um problema real reveste-se de grande importância prática (Jin (2005)). A tarefa é difícil, devido à existência de vários objetivos e às possíveis interações entre eles. De qualquer modo, tem sido propostas várias abordagens para lidar com este problema, envolvendo frequentemente a hibridização de MOEAs com métodos de procura local, conhecidos como algoritmos Miméticos (Gaspar-Cunha (2004b)).

Finalmente, é também fundamental definir/controlar a dimensão do problema no que respeita ao número de objetivos individuais, uma vez que à medida que esse número aumenta o número de soluções não dominadas aumenta também. O problema torna-se muito mais difícil de resolver (pelo menos do ponto de vista dos EAs) porque a pressão de seleção diminui. Simultaneamente, torna-se mais complexo visualizar as correlações existentes entre as diferentes soluções. Deste modo, sempre que possível deve considerar-se a adoção de métodos para reduzir o número de objetivos, como por exemplo técnicas estatísticas (Deb (2006a), Costa (2007)).

Em conclusão, a aplicação eficaz de uma metodologia de otimização multi-objetivo à resolução de problemas de projeto em engenharia requer a existência de conjunto de alargado de características. Assim, estas devem englobar a seleção dos algoritmos mais adequados, o funcionamento de um sistema de suporte de decisão, a análise da robustez das soluções, a redução do número de avaliações exigidas pela rotina de otimização (ou seja, dos tempos de cálculo) e a possibilidade de redução do número de objetivos. O presente texto apresenta e discute uma possível abordagem contemplando estes pressupostos.

2. Metodologia

2.1. Estrutura Geral

Embora a metodologia aqui proposta não possua a dimensão de um MDO integral, ela tenta incorporar os diferentes aspetos associados a uma otimização multi-objetivo (incluindo, como discutido acima, a tomada de decisão, a robustez das soluções e a redução do número de objetivos e tempos de computação) utilizando tanto os cálculos necessários à avaliação de aspetos específicos do problema a resolver (modelação do escoamento ou da transferência de calor, de velocidades e pressões aerodinâmicas, das tensões estáticas de uma estrutura sujeita a um determinado carregamento, etc), como estimativas semi-quantitativas ou mesmo qualitativas (por exemplo, avaliação do conforto/ergonomia, atributos estéticos). A articulação entre as diversas disciplinas do conhecimento é conseguida através dos objetivos selecionados para avaliar as soluções.

Como se mostra na Figura 1, a metodologia envolve as etapas de análise, modelação e otimização. Inicialmente, devem identificar-se as características gerais do problema a resolver, especialmente quais os principais objetivos e restrições, quais os parâmetros do processo principal, se existem ferramentas de cálculo disponíveis (e automáticas) capazes de fornecer soluções para posterior avaliação, se deve também ser utilizado conhecimento empírico e qualitativo, etc. Procede-se então a uma etapa inicial de otimização, empregando um MOEA com o objetivo de se obter uma boa aproximação à frente ótima de Pareto. O DM define a importância relativa dos vários objetivos através de pesos relativos e, dependendo do grau de confiança nessa decisão (que pode ser definida no próprio algoritmo), o MOEA será capaz de avançar no sentido de se obterem fronteiras de Pareto de dimensões diferentes (Ferreira (2008)). A análise de robustez pode sugerir um número mais restrito de soluções. Se o problema tem muitos objetivos, deve-se também tentar reduzir o seu número, como se discutiu anteriormente. Dependendo do tempo gasto na avaliação de cada função objetivo, podem aplicar-se algoritmos híbridos, por exemplo MOEA acoplados a funções de aproximação (Gaspar-Cunha (2004b)). O resultado desta fase consistirá num conjunto de fronteiras de Pareto que explicitam o balanço entre os diferentes objetivos e as variáveis de decisão (ou seja, os parâmetros a otimizar). As soluções são então apresentadas ao DM sob forma gráfica. Este passo é fundamental quando existem objetivos não quantificáveis, ou quando é necessário considerar saber empírico. O DM identifica regiões de espaço de procura (no domínio dos objetivos) que podem satisfazer as necessidades em termos de projeto (no domínio das variáveis de decisão). Estas regiões estão assinaladas por DM1 e DM2 na Figura 1.

Na etapa seguinte, resolve-se o processo inverso, isto é, determina-se o conjunto de pesos (um por cada objetivo) que corresponde às regiões do espaço de procura selecionadas pelo DM. Esta informação é incorporada novamente no MOEA para gerar novas soluções. O processo é repetido até que o DM fique satisfeito com os resultados obtidos.

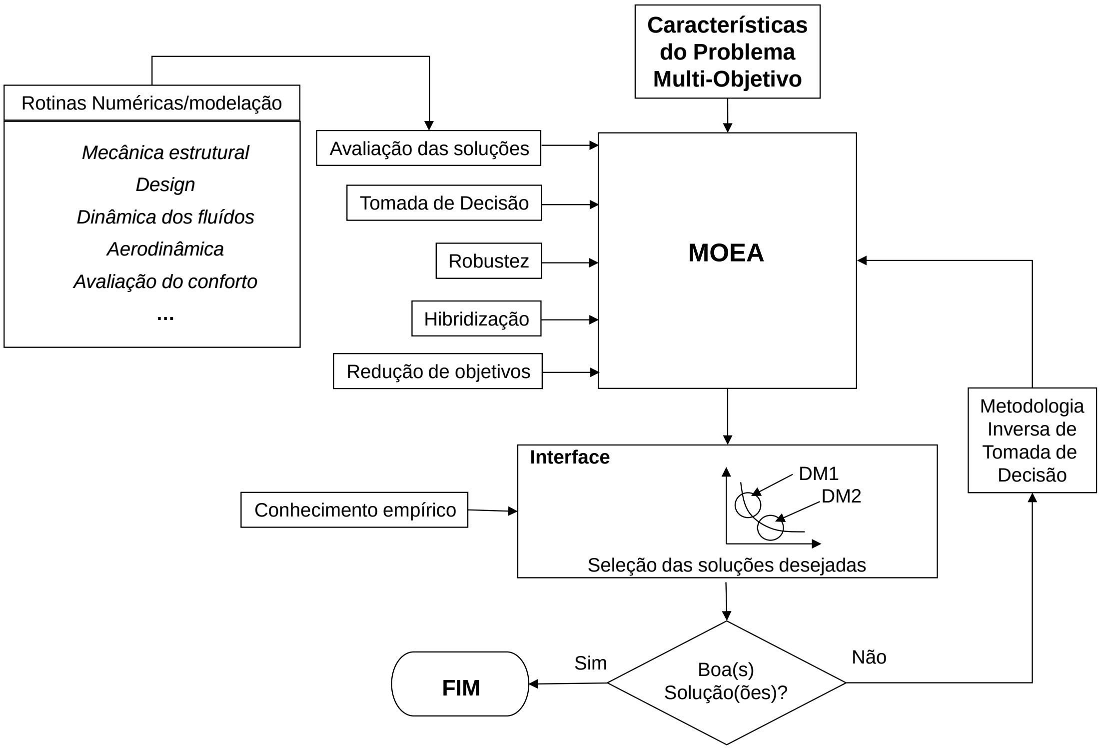

Figura 1 - Sistema Multi-Objetivo de Projeto e Otimização (MO-MDO).

2.2. Algoritmo Evolutivo Multiobjectivo

O papel da otimização é encontrar o melhor conjunto de variáveis de decisão que minimizam ou maximizam uma função objetivo. Como a maioria dos problemas reais envolve a satisfação simultânea de vários objetivos, o resultado de um processo de otimização é um conjunto de soluções de compromisso que leva em conta todos os objetivos considerados. Por exemplo, se se pretender minimizar o custo e maximizar o desempenho de um sistema específico, existem duas soluções ótimas quando cada um dos objetivos é otimizado individualmente. No entanto, se os objetivos são otimizados simultaneamente, o resultado do processo é um conjunto de soluções que explicitam a relação de conflitualidade entre os objetivos considerados. Este conjunto é designado como conjunto ótimo de Pareto e é constituído por todas as soluções não dominadas. Assim, uma solução não-dominada é aquela para a qual não existe outra solução que seja melhor nos objetivos considerados. Na prática, os MOEA permitem obter uma boa aproximação à frente ótima de Pareto. Estes conceitos podem ser explicitados mais formalmente usando as definições seguintes:

Definição 1: Problema de Otimização Multi-Objetivo

Maximizar $F(X)=\left(f_1(X),\ldots,f_N(X)\right)$ sujeito a $\boldsymbol{X}=(x_1,\ldots,x_L)\in S$, onde $S=\left\{X\in\Re^L : g_r(X)\geq 0, r=1,\ldots,K\right\}$, L é o número de variáveis de decisão e K é o número de restrições.

Definição 2: Dominância de Pareto
Um vetor $\boldsymbol{u}$ = (u_1, …, u_N) domina um vetor $\boldsymbol{v}$ = (v_1, …, v_N) se e só se $\forall i \in \{1, \ldots, N\}, u_i \geq v_i$ e $\exists i \in \{1, \ldots, N\}: u_i > v_i$.

Neste trabalho utiliza-se um MOEA desenvolvido pelos autores, designado por *Reduced Pareto Set Genetic Algorithm*, RPSGA (Gaspar-Cunha (2004a)). Neste algoritmo, tanto a distribuição homogénea da população ao longo da fronteira de Pareto, como a melhoria das soluções ao longo de gerações sucessivas, são promovidas por uma técnica de *clustering*, que é aplicada em cada geração no sentido de reduzir o número de soluções na fronteira de Pareto. Esta técnica de *clustering* é baseada na ideia de que as soluções que estão próximas a menos de uma distância pré-definida podem ser agrupadas em torno de uma solução que as possa representar em termos dos valores das várias funções objetivo.

A estrutura do RPSGA está ilustrada no algoritmo 1 abaixo. Inicialmente, criam-se uma população externa vazia, p_e e um arquivo vazio (linha 1) e gera-se aleatoriamente uma população interna, p_i, com n soluções (linha 2). Em cada geração, ou seja, enquanto a condição de terminação não for satisfeita, serão executadas as operações seguintes:

i) as soluções candidatas, p_i, são avaliadas pela rotina de modelação (linha 4);
II) emprega-se uma técnica de *clustering* (algoritmo 2) para reduzir o número de soluções existentes na fronteira de Pareto e calcular a aptidão de cada indivíduo pertencente a p_i (linha 5);
III) um número fixo dos melhores indivíduos é copiado para p_e (linha 6);
IV) se a população p_e estiver completa, emprega-se novamente a técnica de *clustering* (algoritmo 2) para classificar os indivíduos pertencentes a p_e (linha 8) e um número pré-definido dos melhores indivíduos de p_e é incorporado na população p_i para que aqueles que têm menor aptidão sejam substituídos (linha 9);
v) se a população p_e não estiver completa, selecionam-se alguns indivíduos de p_i (linha 11) para utilizar os operadores de cruzamento e mutação (linha 12), a fim de se obterem novos indivíduos para a próxima geração;
VI) todas as soluções não dominadas identificadas durante os cálculos são copiadas para o arquivo (linha 13);
VII) todas as soluções não-dominadas do arquivo são convertidas em resultado final após filtragem.

O Algoritmo 2 (técnica de *clustering*) inicia-se com a definição do número de classes, N_{Ranks}, (linha 1) e cada classe, Rank[i] é classificada como 0 (linha 2). Para cada classe, a população é reduzida para *NR* indivíduos usando a técnica de *clustering* (linhas 4 e 5), ou seja, são escolhidos os *NR* (melhores) indivíduos como representando a população. De seguida, atribui-se a classificação *r* a estes *NR* indivíduos (linha 6) até que o número de classificações pré-definidas seja atingido (linha 7). Finalmente, calcula-se a aptidão de cada indivíduo, F_i, usando uma função de classificação linear (linha 8).

Algoritmo 1: RPSGA

```
Inicializar p_e (população externa) e o Arquivo como conjuntos vazios
A população inicial (interna) p_i é gerada aleatoriamente
enquanto (a condição de terminação não for satisfeita)
    Avaliar p_i
    Avaliar a aptidão dos indivíduos usando a técnica de clustering
    Copiar os melhores indivíduos para p_e
    se (população externa está completa)
        p_e ← Clustering p_e
        Copiar os melhores indivíduos de p_e para p_i
    fim
    Selecionar os indivíduos para reprodução
    Aplicar os operadores de cruzamento e mutação para se obter novos
    indivíduos
    Adicionar as soluções não-dominadas ao Arquivo
fim
Filtar o Arquivo
Retornar o Arquivo
Fim
```

Algoritmo 2: Clustering

```
Definição de N_Ranks
Rank[i] = 0
r = 1
fazer
    NR = r (N/NR_anks)
    Reduzir a população até restarem NR indivíduos
    r = r + 1
enquanto (r < N_Ranks)
Calcular a aptidão, F_i
fim
```

3. Tomada de Decisão

3.1. Métodos Existentes

Em problemas reais, o resultado final de um processo de otimização multiobjetivo deve consistir numa solução única e não no conjunto de soluções que pertencem à fronteira de Pareto porque, obviamente, somente uma solução poderá ser implementada. Para tal, torna-se necessário que o DM introduza, nalgum momento ao longo do processo de otimização, a identificação da importância relativa dos vários objetivos (Deb (2001)). Portanto, o desfecho de um processo de otimização multiobjetivo decorre não só da fase de otimização, mas também de tomadas de decisão.

Sob este aspeto, os métodos de otimização multi-objetivo podem ser classificados como de não-preferência ou baseados em preferência, dependendo do momento em que é tomada a decisão sobre a importância relativa dos objetivos. No primeiro caso, o problema é resolvido independentemente da definição dessa importância e a(s) solução(ões) obtida(s) é(são) disponibilizada(s) ao DM, que as aceita ou rejeita. No segundo caso, as preferências do DM são introduzidas no processo de procura e, em princípio, será obtida a solução que melhor satisfaz as suas preferências (Miettinen (1999)).

Durante a sequência de otimização, as preferências do DM podem ser introduzidas de três formas distintas: i) os objetivos individuais são agregados numa função objetivo única, sendo necessário neste caso decidir *a priori* como essa combinação é realizada; ii) os processos de decisão e de otimização decorrem alternadamente, ou seja, depois de cada etapa de otimização o DM indica as suas preferências relativamente ao conjunto de soluções disponíveis, por forma a que o algoritmo prossiga com a procura; iii) o grupo de objetivos é otimizado simultaneamente de modo a obter-se o conjunto de soluções não dominadas, sendo que no fim a melhor solução é selecionada integrando as preferências do DM (Deb (2001)).

Na literatura existem diversos métodos para combinar o processo de procura com as preferências do DM. Por exemplo, a estratégia de tomada de decisão proposta por Fonseca e Fleming (1998) baseia-se em prioridades e objetivos pré-definidos. No entanto, a sua aplicação exige a articulação progressiva das preferências com as consequentes alterações na localização das soluções no espaço dos objetivos, bem como uma interação permanente com o DM. Além disso, o método exige a definição de dois conjuntos de parâmetros, objetivos e correspondentes prioridades, o que não é uma tarefa fácil. Métodos alternativos incluem métricas ponderadas (Miettinen (1999)), taxa marginal de substituição (Miettinen (1999)), vetor de pesos (Deb (2001)), funções de utilidade (Keeney (1976)), partilha tendenciosa (Deb (2001)), dominação guiada (Branke (2004)), dominação ponderada (Parmee (2000) e ponto base referência (Deb (2006b)). A sua aplicação a problemas reais pode gerar diversas dificuldades, como sejam: i) a inexistência de algumas soluções ideais de Pareto propostas por métricas ponderadas, dependendo do grau de não-convexidade do problema (Miettinen (1999)); ii) um grande esforço computacional no caso da taxa marginal de substituição (Deb (2001)), e iii) a incapacidade do vetor de pesos resolver satisfatoriamente problemas com frentes não-convexas, especialmente quando se atribui importância preponderante a um dos objetivos (Deb (2001)). Em geral, na maioria destes métodos o DM tem de definir vários parâmetros do algoritmo, o que requer um bom conhecimento *a priori* das características da frente de Pareto (Miettinen (1999)).

3.2. Método da Função Tensão

Com o objetivo de contornar as limitações dos métodos citados na seção anterior, apresenta-se e discute-se aqui uma metodologia alternativa. O procedimento é baseado no uso de uma função tensão ponderada (*Weighted Stress Function Method*, WSFM) (Ferreira (2008)), que integra as preferências do DM após a conclusão do processo de procura. Assim, o método é usado *a posteriori*, significando que a procura e a decisão são sequenciais. O WSFM baseia-se em dois pressupostos: primeiro, que a solução que melhor satisfaz as preferências do DM pertence à fronteira de Pareto; segundo, que a fase de seleção considera um vetor objetivo ideal (denominado Z*) que maximiza cada uma das funções objetivo. A otimização individual de cada objetivo corresponde ao valor máximo do objetivo global. A importância relativa atribuída a cada objetivo através da definição de pesos induz uma *tensão*. A melhor solução minimiza a função tensão calculada para cada objetivo. O conceito baseia-se no comportamento mecânico de tensão-deformação típico dos termoplásticos vulcanizados, TPV. Um TPV é constituído por uma fração de partículas de elastómero reticulado suspensas numa matriz termoplástica (Coran (1980)). Para frações volúmicas de partículas até 1.0%, as curvas tensão-deformação apresentam três regiões com comportamento distinto: i) para pequenas deformações, as tensões aumentam drasticamente; ii) para deformações intermédias, as tensões são relativamente estáveis; iii) para deformações elevadas, o material endurece (Coran (1980)). O método proposto faz uma analogia entre esse comportamento e uma função tensão representando os pesos definidos para cada objetivo, sendo os pesos definidos pelo DM representados pela fração de material elastomérico (Ferreira (2008).

A Figura 2 representa uma fronteira de Pareto para um problema com dois objetivos, f_1 e f_2, a serem maximizados. Para cada solução pertencente a essa fronteira definem-se duas funções tensão (σ_{w1} e σ_{w2}), associadas a cada um dos objetivos. Se w_1 e w_2 são os pesos atribuídos, cada função tensão é proporcional à distância ponderada entre o objetivo *i* e o *i*-ésimo componente do vetor objetivo ideal, z*.

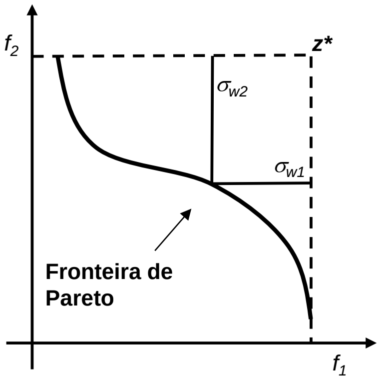

Figura 2 – Funções tensão associadas a uma solução na fronteira de Pareto, para um problema de otimização com dois objetivos a serem maximizados.

Uma vez que os objetivos estão em conflito, a solução do problema não é o vetor objetivo ideal e, simultaneamente, as tensões associadas à melhor solução não são nulas. De facto, a melhor solução possível é aquela para a qual as diferenças entre as tensões associadas a cada objetivo são mínimas. Tendo em conta estas considerações, pode definir-se a função tensão (que relaciona σ_{w1} com w_i) da seguinte forma (Ferreira (2008)):

$$\sigma_{w_i}(f_i) = \begin{cases} \dfrac{w_i}{2}\tan\left(-\dfrac{\pi}{\psi(w_i)}(f_i - w_i)\right) + \xi(w_i), & f_i \le w_i \\[2ex] -\dfrac{\xi(w_i)}{\tan\left(\dfrac{\pi}{\varphi(w_i)}(w_i - 1)\right)}\tan\left(-\dfrac{\pi}{\varphi(w_i)}(f_i - w_i)\right) + \xi(w_i), & f_i > w_i \end{cases} \qquad (1)$$

onde:

$$\varphi(w_i) = \frac{3}{4}(1 - w_i)^2 + 2(1 - w_i) + \delta_1 \qquad (2)$$

$$\psi(w_i) = \varphi(w_i) + 4w_i - 2 \qquad (3)$$

$$\xi(w_i) = -\frac{1}{\tan\left(-\dfrac{\pi}{2 + 2\delta_2}\right)}\tan\left(\frac{\pi}{1 + \delta_2}\left(w_i - \frac{1}{2}\right)\right) + 1\cdot \qquad (4)$$

Nestas equações, os objetivos estão normalizados no intervalo [0,1]. A resolução do problema de otimização multi-objetivo (com *N* objetivos) consiste em resolver o seguinte problema de um único objetivo:

$$\begin{aligned} &\text{Minimize} \quad T(X) = \max_i \ \sigma_{w_i}\left(f_i(X)\right) \\ &\text{Subject to} \quad X \in S \end{aligned} \tag{5}$$

Finalmente, a função de aptidão (DF) deve combinar o conceito de não-dominação com a importância relativa de cada objetivo. Ela é definida tendo em conta o *ranking* das soluções, conforme definido pelo conceito de não-dominância – *Rank (X)* e as preferências do DM, quantificadas pela equação 5 - T(X):

$$DF(X) = \text{Rank}(X) + \frac{T(X)}{T(X)+1} \tag{6}$$

Esta técnica pode ser aplicada a qualquer algoritmo multi-objetivo. No caso do RPSGA, o algoritmo é executado inicialmente sem modificações (ou seja, sem introduzir as preferências de DM) durante N1 gerações (gerações de procura), para se obter uma primeira aproximação à fronteira de Pareto. De seguida, aplica-se o algoritmo modificado durante N2 gerações (gerações de decisão) com a equação 6, para ter em conta o *input* do DM. Simultaneamente, usa-se um parâmetro de dispersão, □ que varia no intervalo] 0,1 [, para controlar a extensão do subconjunto da fronteira de Pareto a alcançar (quanto maior □□□ maior a região (Ferreira (2008))).

4. Robustez

Idealmente, o desempenho de uma solução resultante de um processo de otimização deveria ser indiferente às mudanças inevitáveis que ocorrem nas variáveis de decisão, ou nos parâmetros ambientais. Por outras palavras, as soluções não devem ter apenas bom desempenho, também devem ser robustas. Na prática, podem surgir diferentes tipos de dificuldades relacionadas com a robustez (Ray (2002), Jin (2005), Tsutsui (1997), Chen (1999): i) o desempenho é afetado pelo ruído originado pelas medições de sensores e/ou parâmetros ambientais; ii) as variáveis de decisão alteram-se após a solução ótima ter sido encontrada; iii) o desempenho é estimado por uma aproximação ao valor real da função objetivo; IV) o desempenho varia temporalmente, o que implica que o algoritmo de otimização deve ser atualizado continua ou periodicamente. Serão aqui considerados problemas da segunda categoria.

Face ao exposto, o algoritmo de otimização deve determinar simultaneamente a solução (ou o conjunto de soluções de uma otimização multi-objetivo) que maximizam o desempenho e a robustez. Deve realizar-se não só uma análise de robustez durante a procura, mas também ter em conta que a robustez pode estar em conflito com os objetivos definidos. Tal implica conhecer a sua interdependência para cada problema de otimização. No caso de problemas de otimização com uma única função objetivo, a análise de robustez foi aplicada em vários campos de engenharia, utilizando metodologias de otimização diferentes (Tsutsui (1997), Chen (1999)). Apenas recentemente se alargaram estes estudos a problemas de otimização Multi-Objetivo (Gaspar-Cunha (2008), Kouvelis (2006), Bagchi (2003), Deb (2006c)). Dependendo do tipo de fronteira de Pareto, a análise de robustez pode localizar a seção mais robusta dessa fronteira (Gaspar-Cunha (2008), Deb (2006c)) ou a fronteira mais robusta (caso de um problema multimodal) (Deb (2006c)).

A Figura 3a ilustra o conceito de robustez para problemas com uma função objetivo. No caso da solução S1, pequenas variações na variável de decisão x_1 produzem grandes variações na função objetivo, ao contrário do que acontece com a solução S2. Assim, a solução S2 é mais robusta, apesar de ter menor desempenho (função objetivo a maximizar). Como S1 tem mais desempenho e S2 é mais robusta, é essencial fazer um balanço entre desempenho e robustez. O conceito pode ser extrapolado para o caso multi-objetivo (Figura 3b): a mesma mudança na variável x_1 produz uma maior alteração na solução S2, ou seja, S1 é mais robusta.

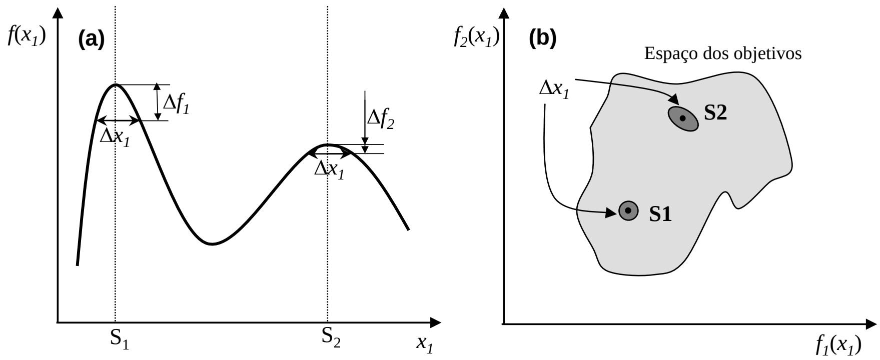

Figura 3 - Conceito de robustez para: a) otimização uni-objetivo; b) otimização multi-objetivo.

Na literatura foram propostos dois tipos de medidas de robustez (Gaspar-Cunha (2008), Jin (2005), Ray (2002), Deb (2006c)):

- Medida de expectativa: a função objetivo original é substituída por uma medida que tem simultaneamente em conta o desempenho da solução e a expectativa de desempenho na vizinhança dessa solução;

- Medida de variância: consiste na adição de um objetivo ao problema de otimização, que mede o desvio do objetivo em torno do ponto considerado. Assim, no caso de uma função uni-objetivo, o problema de otimização transforma-se num problema multi-objetivo (com dois objetivos), um relativo ao desempenho da solução e outro à sua robustez.

Gaspar-Cunha e Covas (2008) avaliaram o desempenho de várias medidas de expectativa e de variância tendo em conta a: i) facilidade de aplicação a problemas onde a forma da função objetivo é desconhecida *a priori*, ii) capacidade de definir a robustez qualquer que seja a forma da função objetivo, iii) independência dos parâmetros do algoritmo, iv) definição do máximo da função na representação Aptidão *versus* Robustez de Pareto e v) eficiência. A seguinte medida de variância exibiu o melhor desempenho:

$$f_i^R = \frac{1}{N'}\sum_{j=0}^{N}\left|\frac{\tilde{f}(x_j)-\tilde{f}(x_i)}{x_j - x_i}\right|, \qquad d_{i,j} < d_{\max}. \tag{7}$$

Nesta equação, a robustez do indivíduo i é definida como o valor médio da razão entre a diferença da aptidão normalizada do indivíduo i, $\tilde{f}(x_i)$, e a dos seus vizinhos (j) e a distância que os separa, $\tilde{f}(x_j) = \frac{f(x_i) - f_{min}}{f_{max} - f_{min}}$ para a maximização e $\tilde{f}(x_j) = 1 - \frac{f(x_i) - f_{min}}{f_{max} - f_{min}}$ para a minimização de $f(x_i)$. f_{max} e f_{min} representam os limites do intervalo de variação dos objetivos, N' é o número de indivíduos da população cuja distância Euclidiana entre os pontos i e j ($d_{i,j}$) é inferior a d_{max} (isto é, $d_{i,j}<d_{max}$):

$$d_{i,j} = \sqrt{\sum_{m=1}^{M}\left(x_{m,j} - x_{m,i}\right)^2} \tag{8}$$

e M é o número de objetivos. Quanto menor f_i^R, mais robusta a solução.

Similarmente, no caso multi-objetivo a solução robusta deve ser pouco sensível a variações das variáveis de decisão. De um modo geral, cada uma das soluções ótimas de Pareto deve ser analisada em termos da sua robustez. Deve analisar-se o efeito combinado das mudanças que ocorrem em todos os objetivos, com a propósito de obter um conjunto de soluções de Pareto que simultaneamente são robustas e ótimas. Podem surgir situações distintas (Deb (2006c)): i) cada solução na fronteira ótima de Pareto é robusta; ii) apenas algumas das soluções que pertencem à fronteira ótima de Pareto são robustas; iii) as soluções da fronteira ótima de Pareto não são robustas, mas existe uma fronteira de Pareto robusta; iv) algumas das soluções robustas pertencem à fronteira ótima de Pareto, mas outras não. Por razões de simplicidade, no que se segue discutem-se os casos i) e ii). Para o efeito, modificou-se o RPSGA adicionando três novas etapas, Ferreira (2008):

i) cálculo das medidas de robustez tendo em conta um parâmetro de dispersão, ε', que quantifica a extensão da zona de robustez a obter na fronteira de Pareto. O valor é definido pelo DM e varia entre 0 (uma única solução) e 1 (totalidade da fronteira ótima de Pareto);

ii) cálculo da *contagem de nicho* e determinação da aptidão global, que é considerada usando uma *função de partilha*. A ideia provém dos conceitos de nicho e de espécies da teoria da evolução natural, onde se formam populações estáveis de organismos através da criação separada de nichos, forçados a partilhar os recursos disponíveis (Goldberg (1985)):

$$m(i) = \sum_{j=1}^{N} sh\left(d_{ij}\right) \tag{13}$$

onde $sh(d_{ij})$ está relacionado com o individuo i e tem em conta sua distância a todos os seu vizinhos j (d_{ij});

iii) determinação da aptidão global através de:

$$\tilde{F}(i) = Rank(i) + (1-\varepsilon')\frac{R(i)}{R(i)+1} + \varepsilon'\frac{m(i)}{m(i)+1} \tag{14}$$

Esta metodologia pode ser adaptada para diferentes implementações de MO.

5. Algoritmos Miméticos

Uma das maiores dificuldades em aplicar MOEAs é o elevado número de avaliações da função objetivo necessárias para obter uma solução aceitável – tipicamente, da ordem dos vários milhares. Além disso, cada uma destas avaliações é na maior parte das vezes muito demorada, pois pode envolver o uso de códigos numéricos computacionalmente exigentes. Assim, como se referiu anteriormente, é interessante reduzir o número de avaliações aplicando métodos que aproximem os valores das funções objetivo, apesar das dificuldades associadas ao possível elevado número de objetivos individuais e às potenciais interações entre eles. As abordagens propostas podem ser agrupadas em termos da forma como são aplicadas nos MOEAs (Gaspar-Cunha (2004b), Naim (2002)):

a) Durante a avaliação, empregando uma função aproximada (por exemplo, métodos estatísticos, “herança” de aptidão, redes neurais artificiais, etc);

b) Durante a busca local após recombinação, gerando um número de novos indivíduos usando algoritmos de pesquisa local;

c) Durante a recombinação, gerando alguns indivíduos utilizando métodos mais eficientes.

As duas últimas estratégias criam uma aproximação mais rápida à fronteira ótima de Pareto e, consequentemente, reduzem o número de avaliações. Nain e Deb (2002) adotaram a abordagem *a)* combinado um processo iterativo com uma rede neuronal treinada com um conjunto de avaliações exatas da função de aptidão. Posteriormente, o algoritmo genético utiliza a rede neural para estimar a função objetivo para um número fixo de gerações. O processo é repetido até obtenção de uma solução aceitável. Esta estratégia híbrida foi ajustada para ser usada com MOEAs (Gaspar-Cunha 2005)).

Em alternativa, podem acoplar-se EAs a métodos de pesquisa local, dando origem a algoritmos Miméticos (Krasnogor (2005)). A ideia principal consiste em obter, em cada geração do MOEA, algumas (boas) soluções através do uso de um algoritmo de busca local eficiente ou, então, melhorar a velocidade de pesquisa através da introdução de alguma pressão de seleção. Em cada geração do MOEA, as novas soluções são determinadas utilizando não só os operadores de recombinação tradicionais, mas também um mapeamento inverso dos objetivos para as variáveis de decisão. Para o efeito, uma rede Neuronal Artificial, ANN, é ligada a um MOEA. Têm sido sugeridas algumas modificações a este método (Adra (2009)), por exemplo, conectando os MOEAs a uma rede Neural Artificial inversa, IANN, e incorporando as novas soluções melhoradas na população atual do MOEA. No caso do RPSGA tal requer uma etapa adicional, após a fase de seleção. Selecionam-se algumas soluções pertencentes à frente não-dominada, iniciando-se então um procedimento de busca local a partir de cada uma destas soluções; as novas soluções geradas são incorporadas na população principal (Gaspar-Cunha (2011)).

6. Exemplos de Aplicação

Por limitações de espaço, nesta seção ilustram-se os métodos propostos usando três exemplos, cada um com dois objetivos e uma variável de decisão:

TP1: x $\in$[-2;6]; minimizar; *L*=1; *M*=2.

$$\begin{aligned} f_1(x) &= x^2 \\ f_2(x) &= e^{|x|-5} + (6/5)\cos(2x) - 2{,}7x + 1 \end{aligned} \tag{15}$$

TP2: x $\in$[-1;1]; maximizar; *L*=1; *M*=2.

$$\begin{aligned} f_1(x) &= -x^2 + 1 \\ f_2(x) &= \arctan(4x) \end{aligned} \tag{16}$$

TP3: x $\in$[0;5]; maximizar; *L*=1; *M*=2.

$$\begin{aligned} f_1(x) &= -x^2 + 1 \\ f_2(x) &= \arctan(4x) \end{aligned} \tag{17}$$

A figura 4 mostra os resultados decorrentes da aplicação da estratégia de tomada de decisão tendo em conta o uso de 5 combinações diferentes de pesos, nos problemas TP1 e TP2. O método é sensível à importância relativa dos dois objetivos: no TP1, quando a importância relativa de f_1 decresce (deslocando-se de (a) para (c)), o algoritmo converge para valores maiores deste objetivo, enquanto o contrário ocorre para TP2 (o objetivo neste caso é maximizar f_1 e f_2).

Na Figura 5 apresentam-se os resultados da aplicação da metodologia de robustez proposta para os problemas TP1 e TP3. Como se pode observar, a metodologia é capaz de selecionar a região mais robusta, independentemente dela ser contínua ou discreta.

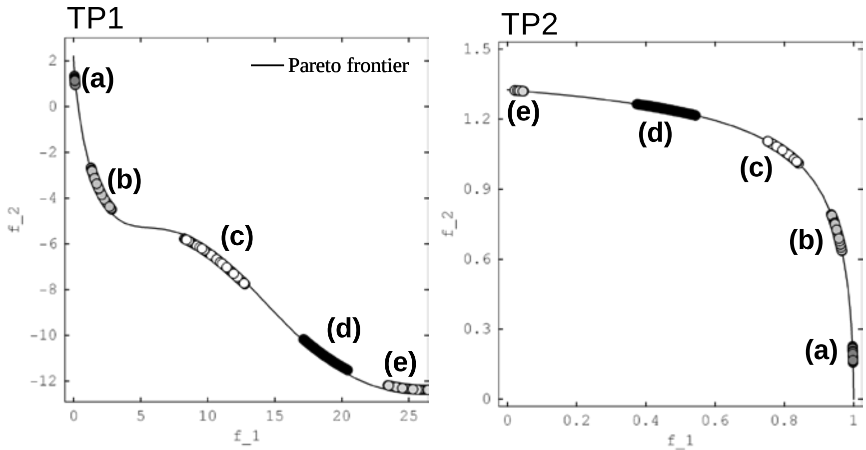

Figura 4- Aplicação da metodologia de DM proposta (□=0.1). Vetores peso: (a) (0.98,0.02), (b) (0.8,0.2), (c) (0.5,0.5), (d) (0.2,0.8) e (e) (0.02,0.98).

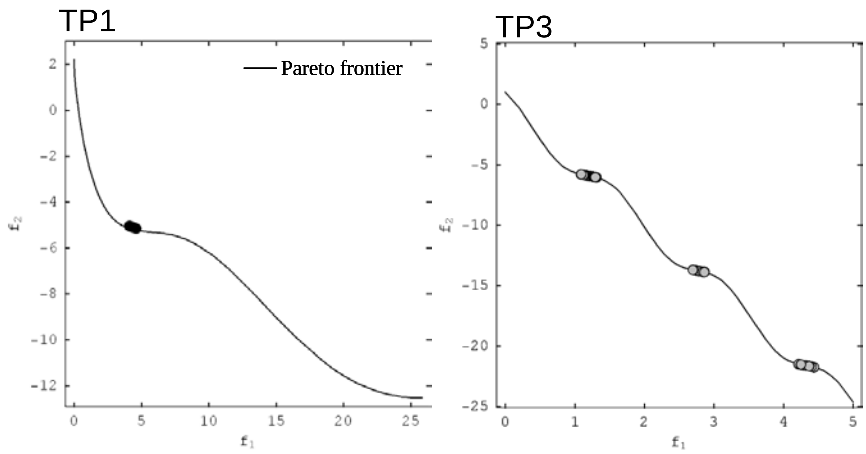

Figura 5- Aplicação da metodologia de robustez proposta (□=0.1).

7. Conclusões

Com o fim de resolver eficazmente problemas complexos de engenharia, neste trabalho propôs-se uma abordagem de otimização Multi-Objetivo de Projeto e Otimização (MO-MDO). A metodologia proposta liga um algoritmo MOEA a estratégias de tomada de decisão e de robustez das soluções, que podem auxiliar o tomador de decisão na escolha das melhores soluções que satisfaçam as suas preferências. Foram também sugeridos dois tipos de algoritmos Miméticos, que podem ser adotados com o objetivo de reduzir o tempo de computação.

Referências

Adra, S.F., Griffin, I., Fleming, P.J., (2009), Multiobjective Memetic Algorithms, 183-208.

Avriel, M.; Dembo, R.S. (eds.), (1979) Mathematical Programming Studies on Engineering Optimization, North-Holland: New York, NY.

Bentley, P.J. (1999), Evolutionary Design by Computers, Morgan Kaufmann: San Francisco, CA.

Bagchi, T.P., (2003), Materials and Manufacturing Processes, 18, 341-354.

Branke, J.; Deb, K., (2004), In Knowledge Incorporation in Evolutionary Computation, pp.461-477.

Branke, J.; Deb, K.; Miettinen, K.; Slowinski, R.; Eds., (2008), Multiobjective Optimization:

Interactive and Evolutionary Approaches. Lecture Notes in Computer Science, Springer-Verlag, Berlin, Heidelberg.

Chen, W.; Sahai, A.; Messac, A.; Sundararaj, G., (1999), In Structural Dynamics and Materials Conference, St. Louis, USA.

Coran, A.Y.; Patel, R., (1980), Rubber Chemistry and Technology, 53,141-150.

Costa, L.; Oliveira, P., (2007), PAMM, Vol. 7, pp. 2060047 – 2060048, Wiley.

Deb, K., (2001), Multi-Objective Optimization using Evolutionary Algorithms, Wiley: Chichester, UK.

Deb, K.; Saxena, D., (2006a), In IEEE Congress on Evolutionary Computation. IEEE Computer Society Press, Los Alamitos, pp. 3353-3360.

Deb, K.; Sundar, J.; Bhaskara, U.; Chaudhuri, S., (2006b), ISSN International Journal of Computational Intelligence Research, 2, 273-286.

Deb, K.; Gupta, H., (2006c), Evolutionary Computation, 14,463-494.

Ferreira, J.; Fonseca, C.; Covas, J.A.; Gaspar-Cunha, A., (2008), In Advances in Evolutionary Algorithms, Kosińsk, W.; Ed.; I-Tech Education and Publishing, Vienna, Austria, pp. 261-278.

Fonseca, C.M.; Fleming, P.J., (1998), IEEE Transactions on Systems, Man and Cybernetics, 28, pp.26-37.

García-Martínez, C.; Cordón, O.; Herrera, F., (2007), European J. of Operational Research, 180,116-148.

Gaspar-Cunha, A.; Covas, J.A., (2004a) In Metaheuristics for Multiobjective Optimisation. Gandibleux, X.; Sevaux, M.; Sörensen, K.; T'kindt, V.; Ed.; Springer: Berlin, Germany; pp.221-249.

Gaspar-Cunha, A.; Vieira, A.S.; Fonseca, C.M., (2004b), In Workshop on Design and Evaluation of Advanced Hybrid Meta-Heuristics, November, Nottingham, UK.

Gaspar-Cunha, A.; Vieira, A., (2005), International Journal of Computers, Systems, and Signals, 6, 18-36.

Gaspar-Cunha, A.; Covas, J., (2008), Computational Optimization and Applications, 39, 75-96.

Gaspar-Cunha, A.; Mendes, F.; Costa, M.F.P., (2011), International Transactions in Operational Research, 18, 2, pp. 183-203.

Goldberg, D.; Richardson, J., (1985), In Proceedings of Second Int. Conf. on Genetic Algorithms, pp. 41-49, 0-8058-0158-8, Cambridge.

Jin, Y.; Branke, J., (2005), IEEE Transactions on Evolutionary Computation, 9, 303-317.

Keeney, R.L.; Raiffa, H., (1976) Decisions with Multiple Objectives: Preferences and Value Tradeoffs, Wiley: New York.

Kouvelis, P.; Sayin, S., (2006), Annals of Operational Research, 147, 71–85.

Krasnogor, N., Smith, J.E., (2005), IEEE Transactions on Evolutionary Computation, 9, 474-488.

Miettinen, K., (1999), Nonlinear Multiobjective Optimization, Kluwer: Boston.

Nain, P.K.S.; Deb, K. (2002) Kangal Report No. 2002005, http://www.iitk.ac.in/kangal/deb.htm.

Oztemela, E.; Tekezb, E.K., (2009), Engineering Applications of Artificial Intelligence, 22, 855-864.

Paquete, L.; Stützle, T., (2007), In Handbook of Approximation Algorithms and Metaheuristics. Gonzalez, T. F.; Ed.; Computer and Information Science Series, Chapman & Hall, Boca Raton, FL, USA.

Parmee, I.C.; Cevtkovic, D.; Watson,A.W.; Bonham, C.R., (2000), Evolutionary Computation, 8, 197-222.

Ray, T., (2002), In Proceedings of the 2002 Congress on Evolutionary Computation, Honolulu, pp. 419-424.

Siddall, J.N., (1982), Optimal Engineering Design, CRC: London.
Tsutsui, S.; Ghosh, A., (1997) IEEE Transactions on Evolutionary Computation, 1, 201-208.