

Otimização Evolutiva Multiobjetivo: Implementação do Algoritmo SPEA2

Palavras-Chave: Computação Evolutiva, Otimização Multiobjetivo, Filtros Digitais IIR

Autores:

Huang Shiheng [FEEC/UNICAMP]

Prof. Dr. Romis Ribeiro de Faissol Attux (orientador) [DCA/FEEC/UNICAMP]

Prof. Dr. Guilherme Palermo Coelho (co-orientador) [FT/UNICAMP]

1 – Introdução

Diversos problemas práticos podem ser vistos como sendo tarefas de otimização dos seus objetivos com respeito aos parâmetros envolvidos. Em outros termos, a otimização pode ser vista como uma tentativa de maximizar as propriedades desejáveis do sistema e, simultaneamente, minimizar aquelas que são indesejáveis [7].

A otimização de dois ou mais objetivos, com algum grau de concorrência, é denominada otimização multiobjetivo. Nela, a qualidade das soluções candidatas pode ser avaliada através da relação de dominância entre elas: uma solução X domina a solução Y quando algum objetivo de X for melhor que o mesmo de Y , sendo melhor ou equivalente para os demais objetivos.

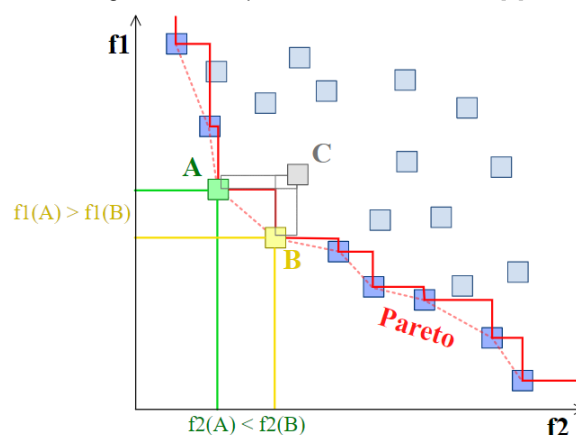
Assim, procuramos obter a fronteira de Pareto [4] no espaço dos objetivos, que, idealmente, recobre todas as variedades de soluções não-dominadas. A Figura 1 apresenta um exemplo de fronteira para um problema de minimização de dois custos, f_1 e f_2 . Podemos notar que as soluções A e B dominam a solução C , uma vez que os custos delas são menores em comparação com os custos de C .

Diante dessa situação, os algoritmos evolutivos são opções importantes para atuar como ferramentas de otimização [1]. Isso pode ser

explicado pela capacidade que esses métodos possuem de gerar diversidade e de incorporar a noção de dominância de maneira natural.

Neste projeto, realizamos um estudo sobre os fundamentos de otimização multiobjetivo e da operação focada no algoritmo SPEA2 [10], com a realização de testes de benchmark [1]. Além disso, usamos essa técnica de otimização para o projeto de filtros digitais [3], circuito muito importante na área de processamento de sinais.

Figura 1 – Exemplo de Fronteira de Pareto [6].



2 – Metodologia

O algoritmo SPEA2 (Seção 2.1) foi implementado em MATLAB e foi analisado diante de alguns problemas de benchmark (Seção 2.2). O desempenho do algoritmo foi avaliado a partir de

um critério baseado em hipervolume (Seção 2.3). Além disso, o algoritmo foi utilizado para projetar filtros digitais do tipo IIR (Seção 2.4).

2.1 – Algoritmo SPEA2

O algoritmo SPEA2 [10] (*Strength Pareto Evolutionary Algorithm*) é um algoritmo evolutivo clássico usado para solucionar problemas de otimização multiobjetivo, cujo pseudocódigo é mostrado na Figura 2. Usa um arquivo externo para guardar as melhores soluções de cada geração, conferindo um caráter elitista para o algoritmo. O arquivo final corresponde à fronteira de Pareto computada pelo algoritmo.

Figura 2 – Pseudocódigo SPEA2.

```

1-Gere uma população inicial e um arquivo externo vazio
enquanto critério de parada não é alcançado faça    ▷ repetição
  2-Calcule o fitness de todos os indivíduos
  3-Selecionos os indivíduos para o arquivo
  se critério de parada é alcançado então          ▷ condição
    4-Pare o algoritmo
  fim se
  5-Execute seleção binária para preencher o conjunto de pares
  6-Aplique operadores de recombinação e mutação
fim enquanto

```

Para calcular os fitness $F(i)$, com $1 \leq i \leq N + \bar{N}$, dos indivíduos da população e do arquivo da mesma geração, é necessário determinar seus valores das funções objetivos; seus valores de força $S(i)$, igual ao número de indivíduos em que cada um deles domina; seus valores de fitness bruto $R(i)$, iguais à soma dos valores de força dos indivíduos que os dominam; e a densidade de cada um, obtida por $D(i) = \frac{1}{\sqrt{\sigma_i^k + 2}}$, com $k = \sqrt{N + \bar{N}}$,

onde σ_i^k significa a distância no espaço dos objetivos do indivíduo i até o k -ésimo indivíduo mais próximo. Finalmente, o valor de fitness é dado por $F(i) = R(i) + D(i)$.

Os indivíduos não dominados irão compor o arquivo da próxima geração. Caso o número deles seja maior que \bar{N} , então os indivíduos com menores distâncias para outros indivíduos serão descartados. Caso contrário, o arquivo será

preenchido com os indivíduos dominados com os menores valores de fitness.

No torneio binário, dois indivíduos do arquivo serão selecionados aleatoriamente para comparar seus valores de fitness, aquele que tiver o melhor fitness será selecionado para a reprodução. Esse processo é repetido até selecionar N pais.

Na etapa da reprodução, cada par de pais gera dois filhos, cada variável das soluções possui uma probabilidade de recombinação e mutação. Usaremos a recombinação aritmética [5]: dados os valores p_1 e p_2 da mesma variável dos dois pais, os valores p'_1 e p'_2 da variável dos filhos serão:

$$\begin{cases} p'_1 = ap_1 + (1-a)p_2 \\ p'_2 = (1-a)p_1 + ap_2 \end{cases}$$

onde a é um valor escolhido aleatoriamente entre 0 e 1. Usaremos a mutação gaussiana [8]: dado o valor da variável x do pai, o valor y da variável do filho é:

$$y = x + N(0, \sigma),$$

onde $N(0, \sigma)$ é um valor gerado pela função gaussiana de média 0 e desvio padrão σ .

2.2 – Problemas e parâmetros

A Tabela 1 apresenta os problemas biobjetivo de minimização (minimizar f_1 e f_2), o número de variáveis n envolvidas em cada problema e o domínio delas. A Tabela 2 apresenta os parâmetros definidos para cada problema: tamanhos da população e do arquivo fixados em 100; número de iterações que aumenta de acordo com o valor de n ; probabilidade de recombinação, alta para explorar o espaço de busca; probabilidade de mutação, baixa para que poucas variáveis sofram ajustes pequenos em cada iteração. Esses problemas e as informações das suas fronteiras ideais são fornecidos no livro [1].

Tabela 1 – Informações sobre os problemas de benchmarks.

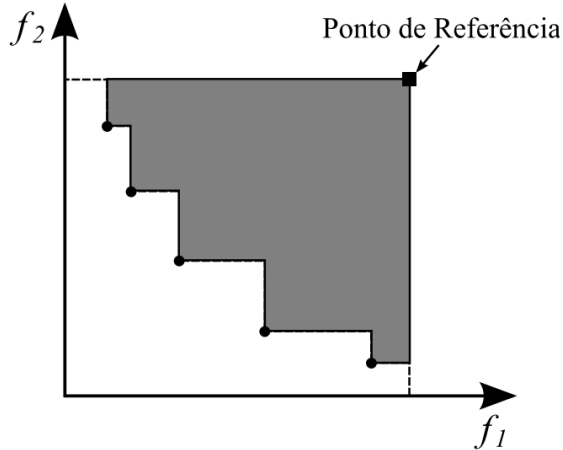
DEB2	$\begin{aligned} f_1 &= x_1 \\ f_2 &= g * h \\ g &= 1 + 10x_2 \\ h &= 1 - \left(\frac{f_1}{g}\right)^2 - \frac{f_1}{g} \text{sen}(12\pi f_1) \end{aligned}$
$\begin{aligned} n &= 2 \\ 0 \leq x_i &\leq 1 \end{aligned}$	

BINH1	$f_1 = x_1^2 + x_2^2$ $f_2 = (x_1 - 5)^2 + (x_2 - 5)^2$
$n = 2$ $-5 \leq x_i \leq 10$	
KURSAWE	$f_1 = \sum_{i=1}^{n-1} \left(-10 \exp \left(-0.2 \sqrt{x_i^2 + x_{i+1}^2} \right) \right)$ $f_2 = \sum_{i=1}^n \left(x_i ^{0.8} + 5 \sin(x_i^3) \right)$
$n = 3$ $-5 \leq x_i \leq 5$	
ZDT1	$f_1 = x_1$ $f_2 = g \left(1 - \sqrt{\frac{f_1}{g}} \right)$ $g = 1 + \frac{9}{n-1} \sum_{i=2}^n x_i$
$n = 30$ $0 \leq x_i \leq 1$	

Tabela 2 – Parâmetros de cada problema (em ordem): tamanho da população, número de iterações, probabilidade de recombinação, probabilidade de mutação e desvio padrão.

	Tam. Pop.	Iter.	P.R.	P.M.	D.P.
DEB2	100	50	1	1/n	0.1
BINH1	100	50	1	1/n	0.1
KURSAWE	100	100	1	1/n	0.1
ZDT1	100	500	0.9	1/n	0.1

Figura 3 – A área em destaque corresponde ao hipervolume da fronteira [2].



2.3 – Razão de Hipervolume

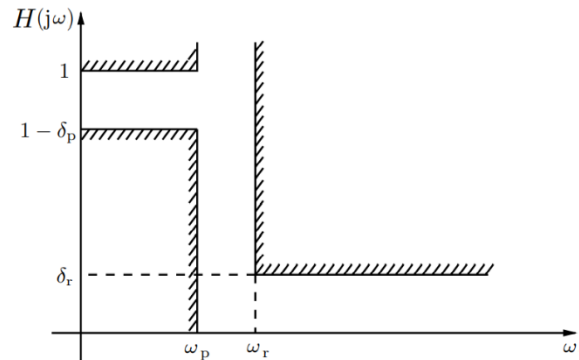
Para avaliar o desempenho, utilizamos a métrica da razão de hipervolume [1]. O hipervolume é formado pelo “volume” limitado pelas soluções não dominadas e pelo ponto de referência e que, para um problema biobjetivo, pode ser interpretado como a área conforme mostrado na Figura 3. Escolhemos cada coordenada deste ponto como sendo o máximo

valor assumido por cada objetivo das soluções ideais. Já a Razão de Hipervolume consiste em dividir o volume formado pelas soluções encontradas e do volume formado pelas soluções ideais. Quanto mais próximo de 1, mais perto da fronteira ideal as soluções estão.

2.4 – Filtros Digitais

Um filtro digital é aquele capaz de realizar uma dada função de transferência, associada a uma equação de diferenças específica pela transformada z [3]. Analisando a resposta em frequência de módulo (Figura 4), a faixa de passagem e de rejeição são aquelas cujos módulos ideais são iguais à 1 e à 0, respectivamente, sendo δ_p e δ_r as ondulações máximas permitidas nessas faixas. A faixa de transição é a região delimitadas por ω_p e ω_r .

Figura 4 – Formato de um filtro passa-baixas. Adaptado de [3].



A função de transferência de um filtro digital do tipo IIR pode ser escrita da seguinte forma [9]:

$$H(z) = K \prod_{i=1}^m \frac{1 + a_i z^{-1}}{1 + b_i z^{-1}} \prod_{j=1}^n \frac{1 + c_j z^{-1} + c_j z^{-2}}{1 + d_j z^{-1} + d_j z^{-2}}$$

Ela é composta por dois produtórios, um de termos de primeira ordem e outro de termos de segunda ordem. Neste projeto, utilizaremos $m = 3$ e $n = 2$. Além disso, escolheremos o ganho K para que o módulo máximo do filtro seja 1.

A codificação de cada solução será um vetor de 19 variáveis (Figura 5). As 5 primeiras são variáveis binárias de controle, sendo p_i

responsável por ativar o i -ésimo termo de primeira ordem quando é 1, e desativá-lo quando é 0 (o termo não entra nos cálculos). O mesmo vale para q_j , responsáveis pelos termos de segunda ordem. Já as outras variáveis são os coeficientes de $H(z)$ e podem adquirir valores entre -5 e 5 .

Projetaremos os filtros baseados nas especificações de [9]: frequência normalizada entre 0 e 1; $w_p = 0.2$; $w_r = 0.3$; $\delta_p = 0.1088$; $\delta_r = 0.17783$. Utilizaremos 500 iterações; tamanho da população e do arquivo de 100; probabilidade de mutação (flip dos bits) e de recombinação de $1/5$ e 1 para as variáveis de controle, respectivamente; probabilidade de mutação e de recombinação de $1/14$ e 0.9 para os coeficientes, respectivamente; desvio padrão de 0.2 e utilizaremos 101 amostras retiradas da resposta do filtro nos cálculos.

Seja P_n , T_n e R_n o número de amostras na faixa de passagem, de transição e de rejeição, respectivamente. Primeiramente, definiremos o erro de módulo na faixa de passagem como sendo:

$$H_p(w) = \begin{cases} 1 - \delta_p - |H(e^{jw})|, & |H(e^{jw})| < 1 - \delta_p \\ 0, & |H(e^{jw})| \geq 1 - \delta_p \end{cases}$$

Definiremos o erro de módulo na faixa de rejeição do seguinte modo:

$$H_r(w) = \begin{cases} |H(e^{jw})| - \delta_r, & |H(e^{jw})| \geq \delta_r \\ 0, & |H(e^{jw})| \leq \delta_r \end{cases}$$

O primeiro objetivo a ser minimizado será:

$$f_1 = \frac{1}{P_n} \sum_{i=1}^{P_n} H_p(w_i) + \frac{1}{R_n} \sum_{j=1}^{R_n} H_r(w_j)$$

Caso f_1 de algum filtro seja atinja 0, isso indica que o filtro atendeu aos requisitos de módulo. Recalcularemos f_1 dele para que o filtro continue evoluindo, adquirindo valores negativos. Para este caso, alteraremos as fórmulas de H_p e de H_r :

$$H_p(w) = 1 - \delta_p - |H(e^{jw})|$$

$$H_r(w) = |H(e^{jw})| - \delta_r$$

Queremos que a fase do filtro na faixa de passagem e transição seja a mais linear possível, para evitar uma grande distorção do sinal filtrado.

Assim, dado a sequência de fases das amostras $\{\theta_1, \theta_2, \dots, \theta_{P_n+T_n}\}$ e definindo $\Delta\theta_i = \theta_{i+1} - \theta_i$, o segundo objetivo (a ser minimizado) será a variância da sequência de diferença das fases. Se a fase for linear, esse objetivo terá valor nulo.

$$f_2 = var\{\Delta\theta_1, \Delta\theta_2, \dots, \Delta\theta_{P_n+T_n-1}\}$$

Por último, queremos que a ordem do filtro seja a menor possível, para reduzir o custo e a complexidade da sua estrutura:

$$f_3 = \sum_{i=1}^m p_i + 2 \sum_{j=1}^n q_j$$

Figura 5 – Codificação dos indivíduos dos filtros

1	2	3	4	5	6	7	8	9	10
p_1	p_2	p_3	q_1	q_2	a_1	b_1	a_2	b_2	a_3
11	12	13	14	15	16	17	18	19	
b_3	c_{11}	c_{12}	d_{11}	d_{12}	c_{21}	c_{22}	d_{21}	d_{22}	

3 – Resultados e discussão

Executando 30 vezes o programa para cada problema, encontramos a média e o desvio padrão do desempenho, mostrados na Tabela 3. Podemos notar que as soluções finais encontradas estão muito próximas das soluções ideais de acordo com o critério da Razão de Hipervolume, uma vez que as médias são muito próximas de 1, além disso, o baixo valor de desvio padrão indica que há pouca diferença dos resultados em diversas execuções, mostrando a consistência do algoritmo.

Tabela 3 – Média e desvio padrão da Razão de Hipervolume avaliada nas 30 execuções.

DEB2	$9.9685e - 01 \pm 1.2582e - 04$
BINH1	$9.9584e - 01 \pm 2.1274e - 04$
KURSAWE	$9.9015e - 01 \pm 9.1615e - 04$
ZDT1	$9.8494e - 01 \pm 8.9391e - 04$

A Tabela 4 mostra um dos filtros do arquivo final de SPEA2 em uma execução (Figura 6 e Figura 7), além dos filtros projetados por diversos métodos clássicos produzidos pelo MATLAB (butterworth, chebyshev tipo 1, chebyshev tipo 2,

elíptico). Podemos observar que o filtro projetado por SPEA2 apresenta uma das menores ordens e domina os filtros elíptico e chebyshev tipo 2. O objetivo f_1 de SPEA2 é superado pelos filtros butterworth e chebyshev tipo 1, entretanto, é melhor comparando os demais requisitos.

Tabela 4 – Valores das funções objetivos encontrados pelos métodos clássicos e por SPEA2.

	f_1	f_2	f_3
Butterworth	-0.14169	12.623	7
Chebyshev 1	-0.13193	25.614	4
Chebyshev 2	-0.10055	10.891	4
Elíptico	-0.09725	1082.666	3
SPEA2	-0.11590	5.030	3

Figura 6 – Módulo do filtro projetado por SPEA2.

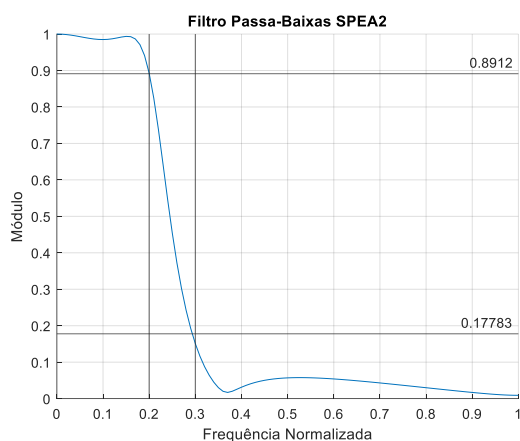
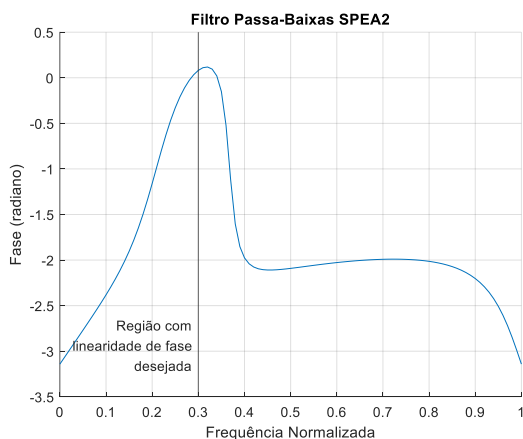


Figura 7 – Fase do filtro projetado por SPEA2.



4 – Conclusão

O algoritmo SPEA2 cumpre seu propósito de ferramenta de otimização multiobjetivo,

alcançando resultados muito próximos das soluções conhecidas dos problemas de benchmark. Além disso, o uso dele para o projeto de filtros digitais mostra a sua capacidade de gerar soluções competitivas em relação aos métodos clássicos.

5 – Referências

- [1] COELLO COELLO, C. A.; LAMONT, G. B.; VAN VELDHIJZEN, D. A. **Evolutionary Algorithms for Solving Multi-Objective Problems**. 2. ed. Urbana: Springer, 2007.
- [2] Coelho, G. P.; Von Zuben, F. J. A Concentration-Based Artificial Immune Network for Multi-Objective Optimization, **International Conference on Multi-Criterion Optimization**, Lecture Notes in Computer Science, v. 6576, p. 343–357, 2011.
- [3] Diniz, P. S. R.; Da Silva, E. A. B.; Netto, S. L. **Processamento digital de sinais: Projeto e análise de sistemas**. 2. ed. Porto Alegre: Bookman, 2014.
- [4] Ehrgott, M. **Multicriteria Optimization**. 2. ed. Springer: Berlin, 2005.
- [5] Mitchell, M. **An Introduction to Genetic Algorithms**. Cambridge: MIT Press, 1998.
- [6] Pareto efficiency. **Wikipedia**, 2021. Disponível em: https://en.wikipedia.org/wiki/Pareto_efficiency#Pareto_frontier. Acesso em: 22 ago. 2021.
- [7] Price, K. V.; Storn, R. M.; Lampinen, J. A. **Differential Evolution: A Practical Approach to Global Optimization**. Berlin: Springer, 2005.
- [8] Schwefel, H. P. **Evolution and Optimum Seeking**. New York: Wiley, 1995.
- [9] Wang, Y; Li, B; Chen, Y. Digital IIR filter design using multi-objective optimization evolutionary algorithm. **Applied Soft Computing**, v. 11, nº 2, p. 1851–1857, 2011.
- [10] Zitzler, E.; Laumanns, M.; Thiele, L. **SPEA2: Improving the Strength Pareto Evolutionary Algorithm**. TIK Report, 2001.