

Otimização Multiobjetivo com Estimação de Distribuição guiada por Tomada de Decisão Multicritério

Pedro M. Sousa Bezerra

Abstract—Considering the state-of-the-art meta-heuristics for multi-objective optimization, only one preference criterion at a time is considered to properly rank candidate solutions along the search process. Here, some of the preference criteria adopted by those algorithms are taken together as inputs to a multi-criteria decision making (MCDM) strategy, responsible for sorting all candidate solutions. New candidate solutions are then generated from this sorted list using a Gaussian mixture model. We used NSGA-II and NSGA-III methods as basis for our proposal, denoted by MOMCEDA (Multi-Objective Multi-Criteria Estimation of Distribution Algorithm). Distinctive aspects of the algorithm are the ranking of candidate solutions and the generation of new individuals. The results on ZDT benchmarks show that our method is competitive with other well-established meta-heuristics for multi-objective optimization.

Index terms: Multiobjective optimization, multicriteria decision making, estimation of distribution.

I. INTRODUÇÃO

Problemas envolvendo a otimização simultânea de múltiplos objetivos ganham cada vez mais evidência com o avanço tecnológico. Há um aumento de interesse por formulações matemáticas voltadas para problemas de otimização multiobjetivo e de tomada de decisão multicritério, os quais possuem frentes de pesquisa bem consolidadas e com as seguintes estratégias:

1. Otimização multiobjetivo (do inglês *Multi-Objective Optimization* – MOO) [1]: busca amostrar a fronteira de Pareto, composta por soluções eficientes do problema e que apresentam diferentes compromissos entre os objetivos, permitindo que a preferência do tomador de decisão seja definida e aplicada a posteriori.
2. Tomada de decisão multicritério (do inglês *Multicriteria Decision Making* – MCDM) [2]: busca explorar a preferência a priori do tomador de decisão, de modo a ordenar por mérito as múltiplas alternativas de soluções eficientes existentes.

Dentre as técnicas já propostas para resolver problemas MOO, destacam-se as metaheurísticas de otimização, em especial as que recorrem a estratégias de busca populacional [3]. Essas metaheurísticas centram seus esforços na descoberta de novas soluções candidatas que sejam diversas entre si e não-dominadas pelas demais soluções candidatas já descobertas, num processo iterativo. Nesta busca iterativa por se aproximar cada vez mais de soluções

pertencentes à fronteira de Pareto, e dadas limitações de memória e processamento que inviabilizam a manutenção de todas as soluções candidatas já descobertas, muitas propostas já foram feitas na literatura para se definir que soluções candidatas já descobertas devem ser descartadas e quais devem ser usadas como ponto de partida para a descoberta de novas soluções. Como as metaheurísticas para MOO recorrem a muitos processos de tomada de decisão durante a busca populacional, a ideia aqui proposta consiste em utilizar técnicas de MCDM para ordenar por mérito as soluções não-dominadas correspondentes no espaço dos objetivos, com o intuito de obter novas soluções a partir daquelas que se mostram mais promissoras. Soluções não-dominadas entre si geralmente são tomadas na literatura como tendo a mesma relevância na definição da próxima população de soluções candidatas [4]. Espera-se, com isso, chegar a técnicas de solução para MOO capazes de explorar ainda mais eficazmente os recursos computacionais disponíveis, além de potencialmente conduzir a soluções de melhor qualidade.

Visto que é razoável supor que as soluções candidatas melhor classificadas estão localizadas nas regiões mais promissoras do espaço de busca, a lista ordenada de soluções candidatas pode ser explorada para conceber um modelo de estimação de distribuição [5]. Neste trabalho, recorreremos a um modelo de mistura de gaussianas, onde o centro de cada gaussiana está posicionado em uma solução candidata e os coeficientes de ponderação das gaussianas são não-negativos, com soma igual a um. O coeficiente de cada gaussiana é proporcional à qualidade relativa da correspondente solução candidata, de modo que soluções de alta qualidade irão contribuir mais para a função densidade de probabilidade. O desvio padrão de cada gaussiana depende de uma distância calculada par-a-par, a ser detalhada posteriormente.

II. METODOLOGIA

A estratégia da técnica inspira-se em algoritmos estado-da-arte em MOO, NSGA-II [4] e NSGA-III [6] - do inglês *Non-dominated Sorting Genetic Algorithm*. Estas metaheurísticas populacionais realizam, a cada geração, uma ordenação por não-dominância dos indivíduos. Os indivíduos são separados em classes de dominância, e

aqueles pertencentes às classes superiores são selecionados para integrar a população da próxima geração. Para comparar soluções candidatas pertencentes à mesma classe, utiliza-se uma medida que indica qual solução deve ser escolhida para melhorar a diversidade das soluções. No entanto, apenas um critério por vez é utilizado para classificar soluções candidatas. Neste trabalho, propomos adotar uma conhecida técnica de tomada de decisão multicritério, o algoritmo TOPSIS (do inglês *Technique for Order of Preference by Similarity to Ideal Solution*) [7], para classificar soluções candidatas dentro de uma mesma classe de dominância segundo múltiplos critérios estabelecidos por um tomador de decisão, como proximidade a valores de referência e a própria promoção de diversidade da população. As preferências do usuário são inspiradas em políticas de decisão já propostas por algoritmos estado-da-arte. A amostragem de novas soluções candidatas é implementada por uma mistura de gaussianas [8], de modo que regiões do espaço de busca que contenham soluções de alta qualidade sejam exploradas de maneira mais intensa.

O algoritmo aqui proposto é denominado MOMCEDA (*Multi-Objective Multi-Criteria Estimation of Distribution Algorithm*) [9]. Sua estrutura principal, apresentada no Algoritmo 1, é inspirada no algoritmo genético proposto no NSGA-II e NSGA-III. A cada geração t , a população de soluções candidatas P_t e seus descendentes Q_t , ambas de tamanho N e inicializadas aleatoriamente, são combinadas em uma população R_t e ordenados por dominância, obtendo uma lista \mathcal{F} de classes de dominância. Os critérios são então avaliados e uma lista ordenada \mathcal{L} dos indivíduos é obtida pelo algoritmo TOPSIS. Os melhores indivíduos de R_t são extraídos e adicionados um a um à população da geração seguinte P_{t+1} . Após uma nova ordenação pelo TOPSIS, o modelo de distribuição \mathcal{M} é construído e utilizado para gerar a nova população de descendentes Q_{t+1} . O processo se repete até atingir um critério de parada.

Os critérios que adotamos para classificação de indivíduos foram:

- Classe de dominância, obtida pela ordenação por dominância como feita no NSGA-II/NSGA-III;
- Tamanho da vizinhança da direção de referência associada a um indivíduo, como no NSGA-III;
- Distância entre o indivíduo e a direção de referência associada a ele, como no NSGA-III;
- Contribuição exclusiva do indivíduo para a métrica de hipervolume.

A métrica de hipervolume é comumente utilizada para comparar o desempenho de diferentes algoritmos de otimização multiobjetivo [10]. Esta métrica mede o “tamanho” do espaço dominado pela população final do algoritmo com relação a um ponto de referência; em duas dimensões, como é o caso dos problemas-teste aqui utilizados, esta métrica corresponde a uma área. Assim,

quanto maior o valor do hipervolume, melhor a solução obtida pelo algoritmo. A contribuição exclusiva que um indivíduo traz para este indicador representa a porção do espaço que é dominada apenas por ele.

Algoritmo 1: MOMCEDA

Resultado: Indivíduos não-dominados de P_t

$t = 1$

$P_t, Q_t = \text{inicializa-populacao}(N)$

Enquanto o critério de parada não for atendido, faça:

$R_t = P_t \cup Q_t$

$\mathcal{F} = \text{fast-non-dominated-sort}(R_t)$

$P_{t+1} = \emptyset$

Enquanto $|P_{t+1}| \leq N$, faça:

avalia-critérios (R_t)

$\mathcal{L} = \text{TOPSIS}(R_t, \mathcal{F})$

$P_{t+1} = P_{t+1} \cup \mathcal{L}[1]$

$R_t = R_t \setminus (\mathcal{L}[1])$

$\mathcal{L}' = \text{TOPSIS}(P_{t+1})$

$\mathcal{M} = \text{gera-modelo}(P_{t+1}, \mathcal{L}')$

$Q_{t+1} = \text{gera-descendentes}(P_{t+1}, \mathcal{M})$

$t = t + 1$

III. EXPERIMENTOS COMPUTACIONAIS

Instruções para a reprodução dos experimentos realizados neste artigo estão disponíveis em <https://github.com/pedro-mariano/IA369>. O conjunto de problemas-teste que serão utilizados para testar o método de otimização multiobjetivo são os problemas ZDT [11]. O conjunto apresenta 5 problemas (de ZDT1 a ZDT4 e ZDT6) com duas funções objetivo a serem otimizadas. As fronteiras de Pareto dos problemas desta classe apresentam diferentes características: convexas, não-convexas, desconexas e não-uniformemente distribuídas. Os problemas possuem a seguinte formulação:

$$\min \begin{cases} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) = g(\mathbf{x}) \cdot h(f_1(\mathbf{x}), g(\mathbf{x})) \end{cases}$$

onde \mathbf{x} é o vetor de variáveis de decisão de uma solução candidata, $f_1(\mathbf{x}), f_2(\mathbf{x})$ são as funções objetivo do problema a serem otimizadas, e $g(\mathbf{x}), h(\mathbf{x})$ são funções auxiliares. O número de variáveis de decisão e as funções variam entre os 5 problemas da classe. A fronteira de Pareto é representada por uma curva no espaço dos objetivos cujos pontos correspondem aos valores de $f_1(\mathbf{x})$ e $f_2(\mathbf{x})$ que constituem as soluções não dominadas do problema de otimização multiobjetivo. No caso dos problemas ZDT, esta curva é dada pelos valores de \mathbf{x} tais que $g(\mathbf{x}) = 1$.

Duas métricas são computadas para avaliar o desempenho do algoritmo: hipervolume e convergência. Aqui, adotamos o ponto de referência com as coordenadas (1,1;1,1) para o cálculo do hipervolume. A medida de

convergência mede a distância média, no espaço dos objetivos, entre as soluções obtidas pelo algoritmo e um conjunto de 1000 amostras equi-espçadas da fronteira de Pareto. Logo, quanto menor o valor desta métrica, melhor a solução obtida.

O conjunto de parâmetros adotado é apresentado na Tabela I. O vetor w apresenta a importância relativa dos critérios utilizados para classificação dos indivíduos, na mesma ordem em que foram definidos na Seção II. Quanto maior o valor atribuído para um dado critério, maior será a importância atribuída a ela na classificação dos indivíduos. Aqui, escolhemos como critério de maior importância a classe de dominância. Estes valores foram ajustados ao longo de diversos testes de forma a maximizar a métrica de hipervolume.

Realizamos 10 execuções para cada problema da classe ZDT. O algoritmo NSGA-II também foi executado para comparação de resultados. Os experimentos foram realizados em uma máquina com as seguintes configurações: Sistema Operacional: Ubuntu 16.04 LTS, Processador: Intel® Core™ i7-4720HQ CPU @ 2.60GHz \times 8, Memória RAM: 16GB. O tempo médio para cada execução do MOMCEDA foi cerca de 100 segundos, e o tempo total médio para cada problema-teste foi aproximadamente 17 minutos. Para o NSGA-II, o tempo médio para cada execução foi cerca de 7 segundos, e o tempo total médio para cada problema-teste foi aproximadamente 1 minuto e 10 segundos.

TABELA I: Parâmetros utilizados para o MOMCEDA

Parâmetro	Valor	Descrição
$seed$	1	Semente aleatória
N_{Pop}	100	Tamanho da população
w	$[10,0; 5,0; 3,0; 1,0]^T$	Importância relativa das preferências do usuário
N_{ref}	100	Número de pontos de referência
N_{eval}	20000	Número de avaliações das funções-objetivo

IV. RESULTADOS

Primeiramente, vamos analisar a evolução da métrica de hipervolume ao longo das gerações para os dois algoritmos, MOMCEDA e NSGA-II, nos problemas ZDT1, ZDT2 e ZDT6. As Figuras 1, 2 e 3 apresentam um diagrama de barras representando o valor médio da métrica e os respectivos desvios padrões em função do número de avaliações das funções objetivo realizadas para cada um dos problemas.

A Tabela II os valores finais das duas métricas avaliadas: hipervolume e convergência, para os dois algoritmos executados. Os valores em negrito destacam o melhor resultado obtido para cada métrica em cada problema-teste.

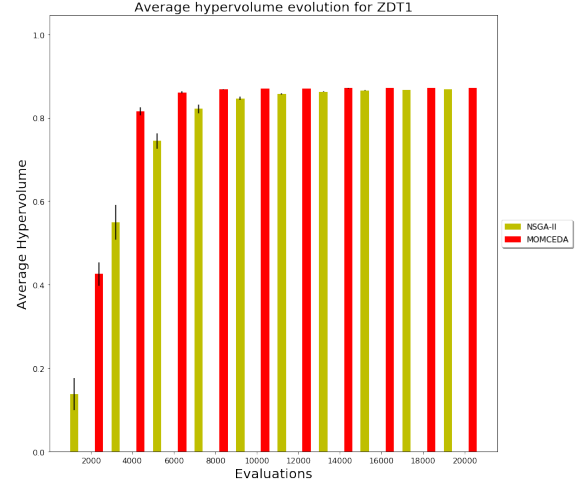


Fig. 1: Evolução do hipervolume médio em função do número de avaliações das funções-objetivo para o problema ZDT1

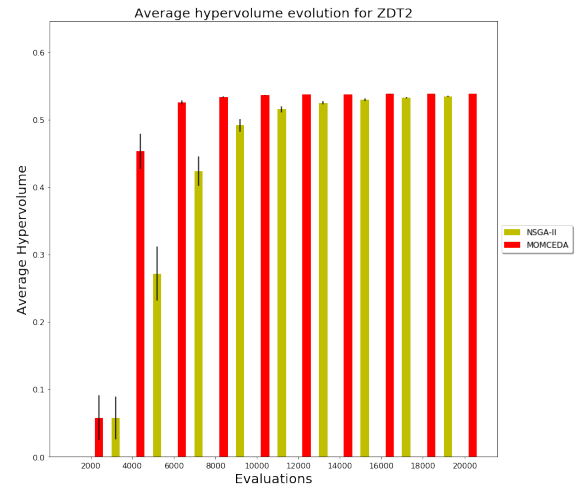


Fig. 2: Evolução do hipervolume médio em função do número de avaliações das funções-objetivo para o problema ZDT2

As Figuras 1, 2 e 3 mostram que o método proposto atinge valores maiores da métrica de hipervolume utilizando menos avaliações de funções do que o NSGA-II nos três problemas considerados. Os resultados finais para as métricas de hipervolume e convergência da Tabela II também mostram um desempenho superior para o método na maioria dos problemas, com exceção para o problema ZDT4. O MOMCEDA também apresentou maior robustez do que o NSGA-II, uma vez que os valores obtidos para desvio padrão são menores na maioria dos casos, o que mostra que os resultados do MOMCEDA são mais consistentes.

TABELA II: Resultados para as métricas de Hipervolume e Convergência nos problemas ZDT

Problema	Métrica	Algoritmo	Média	Desvio padrão
ZDT1	Hipervolume	MOMCEDA	0.871316	7.84E-05
		NSGA-II	0.868347	4.91E-04
	Convergência	MOMCEDA	0.000490	4.63E-05
		NSGA-II	0.002068	2.15E-04
ZDT2	Hipervolume	MOMCEDA	0.538105	8.41E-05
		NSGA-II	0.534382	6.82E-04
	Convergência	MOMCEDA	0.000472	4.38E-05
		NSGA-II	0.002178	4.05E-04
ZDT3	Hipervolume	MOMCEDA	1.328007	2.96E-04
		NSGA-II	1.326033	5.26E-04
	Convergência	MOMCEDA	0.001181	9.86E-05
		NSGA-II	0.001678	2.04E-04
ZDT4	Hipervolume	MOMCEDA	0.853892	2.85E-02
		NSGA-II	0.859106	5.13E-03
	Convergência	MOMCEDA	0.016243	3.17E-02
		NSGA-II	0.006694	2.36E-03
ZDT6	Hipervolume	MOMCEDA	0.504190	1.17E-04
		NSGA-II	0.483644	2.32E-03
	Convergência	MOMCEDA	0.008420	1.77E-02
		NSGA-II	0.015131	1.76E-03

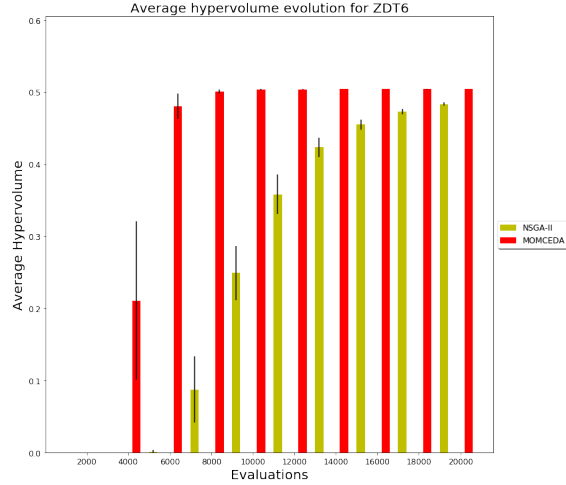


Fig. 3: Evolução do hipervolume médio em função do número de avaliações das funções-objeto para o problema ZDT6

V. CONCLUSÃO

Neste trabalho, propomos um algoritmo, denominado MOMCEDA, para solucionar problemas MOO. O método proposto consiste de um algoritmo genético que utiliza como base os algoritmos NSGA-II e NSGA-III, combinando uma técnica MCDM, o algoritmo TOPSIS, para guiar a busca por novas soluções candidatas com o auxílio de um modelo de distribuição. O método foi avaliado no conjunto de problemas-teste ZDT. Os resultados mostraram que, utilizando o mesmo número de recursos, o método obteve um melhor desempenho que o NSGA-II, além de apresentar resultados mais consistentes, ressaltando os aspectos positivos de inserir mecanismos de tomada de decisão em técnicas de otimização multiobjetivo.

REFERÊNCIAS

- [1] J. Branke, K. Deb, K. Miettinen, and R. Slowinski. *Multiobjective Optimization: Interactive and Evolutionary Approaches*. Springer, 2008.
- [2] M. Köksalan, J. Wallenius, and S. Zionts. *Multiple Criteria Decision Making: From Early History to the 21st Century*. World Scientific, 2011.
- [3] Carlos Coello Coello, Gary B. Lamont, and David A. Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, 2007.
- [4] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.
- [5] M. Hauschild and M. Pelikan. An introduction and survey of estimation of distribution algorithms. *Swarm and Evolutionary Computation*, 1:111–128, 2011.
- [6] Kalyanmoy Deb and Himanshu Jain. An Evolutionary Many-Objective Optimization Algorithm Using Reference-point Based Non-dominated Sorting Approach, Part I: Solving Problems with Box Constraints. *IEEE Transactions on Evolutionary Computation*, 18(4):577–601, 2013.
- [7] C.L. Hwang and K.P. Yoon. *Multiple Attributes Decision Making Methods and Applications*. Springer-Verlag, Berlin, 1981.
- [8] P. Larrañaga and J.A. Lozano. *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Kluwer Academic Publishers, 2001.
- [9] Pedro M. Sousa Bezerra. Multi-objective optimization based on a multi-criteria estimation of distribution. Master's thesis, Faculdade de Engenharia Elétrica e Computação da Universidade Estadual de Campinas - FEEC/ Unicamp, 2018.
- [10] Eckart Zitzler and Lothar Thiele. Multiobjective optimization using evolutionary algorithms — A comparative case study. *Parallel Problem Solving from Nature - PPSN V*, pages 292–301, 1998.
- [11] E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8:173–195, 2000.