

## ADAPTAÇÃO DO ALGORITMO OPTBEES PARA OTIMIZAÇÃO DE PROBLEMAS MULTI OBJETIVO

**Huggo Silva Ferreira**

Universidade Federal de Minas Gerais - UFMG  
Av. Pres. Antônio Carlos, 6627 - Pampulha, Belo Horizonte - MG, 31270-901  
huggosf@ufmg.br

**Renato Dourado Maia**

Universidade Estadual de Montes Claros - Unimontes  
Avenida Dr. Ruy Braga, S/N - Vila Mauriceia, Montes Claros - MG, 39401-089  
renato.dourado@unimontes.br

**Allysson Steve Mota Lacerda**

Universidade Estadual de Montes Claros - Unimontes  
Avenida Dr. Ruy Braga, S/N - Vila Mauriceia, Montes Claros - MG, 39401-089  
steve.lacerda@unimontes.br

### RESUMO

Este artigo propõe uma adaptação do algoritmo OptBees, originalmente proposto para tratar problemas de otimização mono-objetivo em espaços contínuos de forma que ele possa ser utilizado para otimização multiobjetivo. Para testar o desempenho do OptBees-MO, foram utilizados os 10 primeiros problemas irrestritos IEEE Congress on Evolutionary Computation de 2009.

**PALAVRAS CHAVE.** Otimização, OptBees, Abelhas.

**Tópicos:** Otimização Multiobjetivo, Inteligência de Enxame.

### ABSTRACT

This paper presents an adaptation of OptBees algorithm for multi-objective optimization. The OptBees is a multimodal optimization algorithm inspired in the behave of bees in foraging activity. Its adaptation for the multi objectivity uses the OptBees's capability of location of several local optima to the location of non-dominated solutions in multiobjective problems. To test this adaptation, the first ten problems proposed in the IEEE's Congress on Evolutionary Computation 2009 were used. The algorithm performed well in the resolution of most problems and ranks well when compared to the algorithms submitted in the same competition.

**KEYWORDS,** Optmization, OptBees, Bees.

**Topics:** Multi-objective Optmization, Swarm Intelligence

## 1. Introdução

O processo de otimização envolve minimizar ou maximizar os resultados de uma determinada função. Esse processo é vastamente aplicado no mundo real e é comum que tomadores de decisões necessitem realizar essa tarefa [Martínez, Santos 1995]. Para isso, é necessário que, primeiramente, o problema seja traduzido matematicamente, por intermédio da definição de variáveis, restrições e objetivos. A partir daí, a resolução do problema dependerá da escolha de um método apropriado, que pode ser uma heurística ou algum outro tipo de algoritmo.

Na prática, a maioria dos problemas envolve a otimização simultânea de vários critérios (objetivos) e, em algumas dessas situações, é possível que as funções relativas a esses critérios sejam otimizadas sem que ocorra algum conflito, mas isso acaba sendo algo raro. É mais comum que essas funções sejam conflitantes entre si e demandem um tratamento diferenciado. Por exemplo, quando procura-se adquirir um computador, espera-se que ele seja de bom desempenho e de baixo custo. Porém, de maneira conflitante, quanto mais desempenho uma máquina possui, mais cara ela é e, de forma análoga, quanto menor custo, menor desempenho. Por isso, nesse problema, deve-se buscar pelo computador que possua desempenho e preço de forma que não exista outro com melhor desempenho (ou menor preço) e com o preço igual ou menor (ou desempenho igual ou melhor). Tal situação de otimização é conhecida como otimização multiobjetivo [Pérez 2012].

Um problema multiobjetivo pode ser caracterizado como [Coello e Aguirre 2002]:

$$\begin{aligned}
 &\text{Minimizar } z = \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}); f_2(\mathbf{x}); \dots; f_m(\mathbf{x})) \\
 &\text{sujeito a : } \quad \mathbf{g}(\mathbf{x}) = (g_1(\mathbf{x}); g_2(\mathbf{x}); \dots; g_r(\mathbf{x})) \leq \mathbf{b} \\
 &\quad \quad \quad \mathbf{h}(\mathbf{x}) = (h_1(\mathbf{x}); h_2(\mathbf{x}); \dots; h_s(\mathbf{x})) = \mathbf{c} \\
 &\quad \quad \quad \mathbf{x} = (x_1; x_2; \dots; x_n) \\
 &\quad \quad \quad \mathbf{z} = (z_1; z_2; \dots; z_m),
 \end{aligned} \tag{1}$$

em que  $X$  é o espaço de busca,  $Z$  o espaço de objetivos,  $\mathbf{g}(\mathbf{x})$  e  $\mathbf{h}(\mathbf{x})$ , respectivamente, os conjuntos de restrições de desigualdade e de igualdade;  $\mathbf{b}$  e  $\mathbf{c}$  constantes reais não negativas.

Diante da forma como a otimização multiobjetivo se caracteriza, como já comentado, é extremamente difícil que uma única solução seja capaz de atender a todos os objetivos de maneira ideal. Porém, dentre as soluções possíveis, há um conjunto de soluções em que cada solução pertencente a esse conjunto não pode ser considerada melhor que outra também pertencente a ele, mas pode ser considerada superior às demais. Esse conjunto é conhecido como conjunto Pareto-ótimo [Pareto 1896] e suas soluções como soluções não dominadas, pois, dentre elas, não há uma que atenda aos objetivos de forma que seja melhor em pelo menos um objetivo e seja, pelo menos, igual em todos os outros objetivos. Já a projeção do conjunto Pareto-ótimo no espaço de objetivos é conhecida como fronteira Pareto.

Dentre as várias abordagens para tratar de problemas multiobjetivo, este trabalho tem como objetivo propor uma adaptação, chamada de OptBees – Multiobjetivo (OptBees-MO), para o algoritmo OptBees, de forma a utilizar seu desempenho na localização de ótimos locais para a localização das soluções não dominadas. Como o OptBees é voltado para otimização mono-objetivo, foi utilizada a estratégia de atribuição de *fitness* proposta por Zitzler,

Zitzler, et al. [2001] no *Strength Pareto Evolutionary Algorithm II* (SPEA-II) para realizar a adaptação para multiobjetivo. Para melhor entendimento, a seção 2 introduz rapidamente o funcionamento do OptBees e a seção 3 a atribuição de *fitness* do SPEA-II. Na seção 4, é apresentado o OptBees-MO. Os resultados experimentais do OptBees - MO nos dez primeiros problemas irrestritos propostos na competição de algoritmos evolucionários multiobjetivos (*MOEA Competiton*) realizada no *Congress on Evolutionary Computation* (CEC) de 2009 [Zhang et al. 2008] são apresentados na seção 5. A seção 6 conclui o trabalho.

## 2. OptBees

Proposto por Maia *et al.* [2013], o OptBees é um algoritmo mono-objetivo, multimodal e inspirado no comportamento de forrageamento de abelhas, no qual abelhas se movem pelo ambiente (espaço de busca) à procura de fontes de alimento (soluções). Na execução do algoritmo, as abelhas podem ser divididas em dois grupos: o grupo de abelhas ativas, cujas abelhas realizam alguma atividade na iteração corrente, e o grupo de abelhas inativas, cujas abelhas estão à espera de alguma atividade para executar. As abelhas do grupo de abelhas ativa podem ser divididas em três tipos de agentes:

- **Abelhas recrutadoras:** aquelas que encontraram uma fonte de alimento promissora e, então, convocam outras para explorar a área em redor da fonte encontrada.
- **Abelhas recrutadas:** aquelas que respondem à convocação das recrutadoras.

**Abelhas exploradoras:** Abelhas que exploram de maneira contínua e aleatória o espaço de busca em busca de fontes de alimento promissoras.

O processo do algoritmo, para cada iteração, ocorre da seguinte forma: as abelhas se movem pelo espaço de busca à procura de fontes de alimento em potencial. Conforme as qualidades do alimento que são encontrados pelas abelhas, cada uma delas será classificada como recrutadora ou não-recrutadora. Com base no número de abelhas recrutadoras encontrado, é definido quantas das abelhas não-recrutadoras serão recrutadas e quantas continuarão com o processo de exploração do espaço de busca. Já com base no número de abelhas recrutadas e na qualidade da fonte de alimento encontrada por uma abelha, são definidas quantas abelhas responderão ao recrutamento dessa abelha em questão. Isso é feito de forma a simular o fenômeno natural de recrutamento [Gadau e Fewell 2009], em que os movimentos de uma abelha quando encontra uma fonte de alimento depende da qualidade dessa e leva a um maior, ou menor, número de abelhas atraídas. Caso seja encontrado um o número grande de fontes de alimentos promissoras de forma que sejam necessárias mais abelhas para explorá-las do que as ativas no momento, parte das abelhas inativas se torna ativa para suprir essa demanda. Caso o número de fontes de alimento promissoras seja muito pequeno a ponto de que existam mais abelhas ativas do que o necessário, as abelhas ativas excedentes se tornam inativas e deixam o processo de busca.

O algoritmo procede, então, para o processo de recrutamento, no qual as abelhas recrutadas são atraídas por uma recrutadora para explorar a área em redor dessa. Em seguida, o processo de exploração inicia-se e as abelhas não-recrutadas restantes se movem aleatoriamente pelo espaço de busca. Após esses processos serem realizados, inicia-se uma nova iteração.

É interessante destacar que sobre as soluções encontradas pelas abelhas, antes da determinação de recrutadoras e não recrutadoras, é aplicado o algoritmo de busca local *LocalSearch1*, que foi proposto como parte do algoritmo MTS (*Multiple Trajectory Search*) [Tseng e Chen 2008]. Detalhes da implementação do OptBees e como os processos descritos são realizados podem ser lidos em Maia [2012] e Maia *et al.* [2013]. Com base nesse processo, são definidas as seguintes variáveis do algoritmo:

- **Tamanhos mínimo e máximo do enxame:** essas variáveis são responsáveis por limitar os números mínimo e máximo de abelhas ativas durante uma iteração.
- **Raio de inibição social:** define uma área em redor de uma abelha caso ela tenha se tornado recrutadora. Se houver outra abelha recrutadora dentro dessa área, a que estiver explorando a fonte de pior qualidade será inibida e deixará de ser recrutadora.
- **Esforço médio de forrageamento:** número utilizado para calcular a quantidade ideal de abelhas ativas conforme a quantidade fontes de alimentos promissoras exploradas na iteração.
- **Probabilidade mínima para uma abelha ser recrutadora.**
- **Percentual de abelhas não recrutadoras que serão recrutadas.**

### 3. Atribuição de *Fitness* do SPEA-II

Proposto por Zitzler, et al. [2001], o SPEA-II é um algoritmo genético para aproximação de conjuntos Pareto-ótimos para problemas multiobjetivos. Como um aperfeiçoamento ao SPEA [Zitzler et al. 1999], a atribuição de *fitness* do SPEA-II busca concentrar em um único valor as qualidades de solução de cada objetivo encontrado por um indivíduo, bem como a sua diversidade no espaço de objetivos. Isso pode ser observado na Eq. (2), em que  $f_i^{SPEA2}(t)$  é o *fitness* do indivíduo  $i$  na iteração  $t$ ,  $R_i(t)$  indica o *fitness* bruto do indivíduo  $i$  na iteração  $t$  e  $D_i(t)$  indica a densidade do indivíduo  $i$  na iteração  $t$ .

$$f_i^{SPEA2}(t) = R_i(t) + D_i(t), \quad (2)$$

Neste caso, o *fitness* bruto de um determinado indivíduo é o indicador de qualidade com base num critério de não dominância e é dado pelo somatório das forças  $S_j(t)$  dos indivíduos que dominam o indivíduo  $i$ , como indicado na Eq. (3). Força, nesse caso, representa o número de indivíduos  $k$  que o indivíduo  $j$  domina, como indicado na Eq. (4).

$$R_i(t) = \sum_{j \in P(t) e j \leq i} S_j(t), \quad (3)$$

$$dS_j(t) = |\{k | k \in P(t) e j \leq k\}|, \quad (4)$$

Nas Eq 3 e 4,  $P(t)$  é o conjunto de soluções candidatas na iteração  $t$ ,  $|\cdot|$  indica a cardinalidade do conjunto e  $\leq$  a dominância.

Já a densidade de um indivíduo é dada pela Eq. (5), em que  $\lambda_i^k(t)$  indica a distância euclidiana do indivíduo  $i$  para o  $k$ -ésimo indivíduo mais próximo. No caso,  $k$  é definido por  $k = \sqrt{|P(t)|}$ .

$$D_i(t) = \frac{1}{\lambda_i^k(t) + 2}, \quad (5)$$

### 4. OptBees-Mo

Devido ao comportamento de seus indivíduos, isso é, o uso de várias abelhas recrutadoras, o OptBees é capaz de manter a diversidade de sua população e, dessa forma, de localizar vários ótimos locais além de convergir para o ótimo global. Tal característica sugere que ele possa ter um bom desempenho em problemas de otimização multiobjetivo que sejam apropriadamente tratados como mono-objetivo. Assim, ao invés de localizar ótimos locais, a manutenção da diversidade seria utilizada para localizar o melhor conjunto possível de soluções não dominadas.

Em todo o seu processo, o OptBees trabalha com base em apenas uma variável, que é a qualidade da fonte de alimento encontrada e, portanto, foi utilizada a atribuição de *fitness* proposta no SPEA-II. Porém, como a atribuição de *fitness* trata valores menores como melhores e o OptBees trata valores maiores como melhores, os resultados de *fitness* são corrigidos conforme a Eq. (6), em que  $f_i^{Ag}(t)$  é o *fitness* corrigido da abelha  $i$  na iteração  $t$  e está contido entre 0 e 1.

$$f_i^{Ag}(t) = 1 - \frac{f_i^{SPEA2}(t) - \min_j(f_j^{SPEA2}(t))}{\max_j(f_j^{SPEA2}(t)) - \min_j(f_j^{SPEA2}(t))} \quad (6)$$

O algoritmo do OptBees-MO está apresentado no Algoritmo 1.

---

Algoritmo 1. OptBees-MO

---

**Parâmetros de Entrada:**

- $n_i$ : número inicial de abelhas ativas (corresponde também ao número mínimo de abelhas ativas).
- $n_{max}$ : número máximo de abelhas ativas.
- $n_{mean}$ : esforço médio de forrageamento.
- $\rho_i$ : raio de inibição social.
- $p_{min}$ : probabilidade mínima de uma abelha ser recrutadora.
- $prec$ : percentual de abelhas não-recrutadoras que serão de fato recrutadas.

**Parâmetros de Saída:**

- Arquivo externo e os valores correspondentes da função objetivo (as abelhas recrutadoras correspondem às soluções factíveis).

1. Gerar aleatoriamente um enxame de  $n_i$  abelhas (ativas).

**enquanto** (critério de parada não estiver satisfeito) **faça**

2. Avalie as qualidades das fontes de alimento sendo exploradas pelas abelhas ativas.
3. Avalie o fitness das abelhas.
4. Determine as abelhas recrutadoras.
5. Atualize o número de abelhas ativas.
6. Determine as abelhas a serem recrutadas e as exploradoras.
7. Execute o processo de recrutamento.
8. Execute o processo de exploração.
9. Aplique o operador de busca local.
10. Inserção das não-dominadas num arquivo externo.
11. Redução do raio  $\rho_i$ .

**fim enquanto**

---

Além de inserir a atribuição de *fitness* do SPEA-II, foi feita a inserção de um processo de afastamento de abelhas no processo de recrutamento, foi reposicionado o passo de aplicação do operador de busca local e foi inserida a redução do raio de inibição social.

A inserção do processo de afastamento no processo de recrutamento foi feita de forma a aumentar a diversidade no espaço das variáveis. Foi observado que, durante o processo de recrutamento, as recrutadas tendiam a explorar uma área cada vez menor em torno da recrutadora, até que as soluções se tornassem praticamente iguais. Para evitar que isso ocorra, caso a distância entre a recrutada e a recrutadora seja que 0,5% da maior distância possível no espaço de buscas, a recrutada pode ser movida para algum ponto próximo do limite do raio de inibição da recrutadora, como pode ser observado nas Eq. (7) e (8) (cada uma possui probabilidade de 50% de ser utilizada), em que  $u$  é um número aleatório com distribuição uniforme no intervalo  $[-1,1]$ ,  $\mathbf{U}$  um vetor de dimensão igual à de  $x_i$  e composto por números aleatórios com distribuição uniforme no intervalo  $[-1,1]$ .

$$x_i = x_i + \rho_i \otimes \mathbf{U} \cdot \frac{x_i}{|x_i|} \quad (7)$$

$$x_i = x_i + \rho_i \cdot u \cdot \frac{x_i}{|x_i|} \quad (8)$$

O reposicionamento do operador de busca local foi feito com o objetivo de diminuir o custo computacional do algoritmo, pois a utilização logo após a atribuição de *fitness* exigiria que

o *fitness* fosse calculado novamente. Além disso, não foi utilizado o *fitness* como indicador de melhora da solução na busca local, pois a cada alteração em um indivíduo, o *fitness* de cada abelha precisaria ser recalculado. Portanto, para indicar a melhora de solução, foi utilizado o critério de dominância restrita, sendo que a dominância de uma solução  $x$  em uma solução  $y$  se dá nos casos em que:

- A solução  $x$  é factível e  $y$  não;
- $x$  e  $y$  são infactíveis, mas  $x$  viola menos as restrições que  $y$ ;
- $x$  e  $y$  são factíveis, mas  $x$  domina  $y$ .

A utilização da redução do raio de inibição social foi uma estratégia para reduzir o custo computacional e manter o funcionamento normal do algoritmo mesmo com um enxame reduzido. Com o algoritmo iniciando com um raio de inibição maior, mais abelhas são inibidas de virarem recrutadoras e, logo, menos abelhas precisarão se tornar ativas. A cada iteração, o raio será reduzido de forma que abelhas que antes seriam inibidas e atraídas por uma recrutadora possam explorar sua região e atrair outras não-recrutadoras. Após diversos testes, recomenda-se que o raio de inibição inicie com 30% da maior distância possível no espaço de buscas e reduza linearmente até 1% dessa distância com o decorrer das iterações. Cabe destacar que a utilização dessa estratégia não é obrigatória, ficando a cargo do usuário utilizá-la com valores diferentes para o raio de inibição inicial e final ou manter o processo padrão do OptBees.

No arquivo externo são inseridas as soluções não-dominadas da iteração e, em seguida, são retiradas as soluções do arquivo externo que foram dominadas pelas soluções recém inseridas.

## 5. Resultados Experimentais

O OptBees-MO foi testado com os dez primeiros problemas irrestritos do CEC 2009 e comparado com os resultados dos algoritmos que participaram da competição. Foram utilizados apenas os dez primeiros problemas porque os seguintes são de quatro ou mais problemas, o que caracteriza otimização com muitos objetivos atualmente. Para tanto, uma série de restrições foram seguidas. São elas:

- Utilizar a distância geracional invertida (*Inverted Generational Distance* - IGD) como indicador de qualidade.
- Número máximo de soluções para cálculo do IGD:
  - 100 para dois objetivos.
  - 150 para três objetivos.
- Critério de parada: 300000 avaliações da função objetivo.
- Trinta execuções independentes de cada problema.
- Devem ser reportados as médias e os desvios-padrão dos IGDs das trinta execuções de cada problema.

Além disso, todos os testes foram feitos em condições iguais de processamento. Foi configuração do algoritmo:

- $nmin$ : 20 abelhas
- $nmax$ : 350 abelhas
- $nmean$ : 10
- $pmin$ : 10%
- $prec$ : 20%
- $\rho_i$  iniciando em 30% da maior distância possível no espaço de buscas e finalizando em 1% dessa distância

Para comparação dos resultados de IGD, foram selecionados três algoritmos da competição de forma que possa ser vista a diferença dos resultados do OptBees-MO para cada um deles. Dessa forma, os algoritmos selecionados foram o MOEA/D [Zhang et al. 2009], que foi o de melhor desempenho, AMGA [Zhang et al. 2009], que obteve desempenho mediano, e OMOEA-II [Gao 2009], que foi o de pior desempenho.

Sobre os problemas irrestritos, os sete primeiros são problemas de dois objetivos e os três seguintes são de três objetivos. Na Tabela 1 são apresentados a média, o desvio padrão, o mínimo e o máximo do IGD conforme as soluções encontradas nas trinta execuções do OptBees-MO.

TABELA 1. ESTATÍSTICAS DE IGD BASEADAS EM 30 EXECUÇÕES

Problemas	Média	Desvio P.	Mínimo	Máximo
UF1	0,01101	0,00267	0,00775	0,01775
UF2	0,00778	0,00176	0,00598	0,01389
UF3	0,08997	0,00961	0,06623	0,11065
UF4	0,04122	0,00220	0,03817	0,04872
UF5	0,51744	0,16357	0,29819	0,85095
UF6	0,07902	0,03713	0,03592	0,20502
UF7	0,01344	0,00373	0,00835	0,02563
UF8	0,11901	0,02751	0,08156	0,22317
UF9	0,09057	0,03445	0,04903	0,18094
UF10	0,27661	0,06588	0,18651	0,46435

No geral, o pior desempenho do OptBees-MO foi para o problema UF5, ao passo que seu desempenho para os demais problemas é aceitável. A Tabela 2 demonstra o resultado de IGD do OptBees - MO e dos algoritmos selecionados e na Tabela 3 a colocação dos algoritmos conforme seus desempenhos por IGD.

TABELA 2. COMPARAÇÃO DOS RESULTADOS DO IGD PARA OS PROBLEMAS DO CEC 2009

Problemas	OptBees - MO	MOEA/D	AMGA	OMOE - II
UF1	0,01101	<b>0,00435</b>	0,03588	0,08564
UF2	0,00777	<b>0,00679</b>	0,01623	0,03057
UF3	0,08997	<b>0,00742</b>	0,06998	0,27141
UF4	0,04122	0,06385	<b>0,04062</b>	0,04624
UF5	0,51744	0,18071	<b>0,09405</b>	0,1692
UF6	0,07902	<b>0,00587</b>	0,12942	0,07338
UF7	0,01344	<b>0,00444</b>	0,05707	0,03354
UF8	0,11901	<b>0,05840</b>	0,17125	0,192
UF9	0,09056	<b>0,07896</b>	0,18861	0,23179
UF10	<b>0,27661</b>	0,47415	0,32418	0,62754

TABELA 3. COLOCAÇÃO DOS ALGORITMOS CONFORME SEUS DESEMPENHOS

Algoritmos	UF 1	UF 2	UF 3	UF 4	UF 5	UF 6	UF 7	UF 8	UF 9	UF 10	Média
MOEAD	1	4	1	14	7	1	1	1	3	10	4
DMOEADD	6	3	3	6	10	3	4	2	1	3	4
MTS	4	1	5	1	1	2	12	6	8	1	4

<b>LiuLi Algorithm</b>	5	8	2	8	4	10	2	3	6	9	6
<b>OptBees MO</b>	7	5	8	5	12	5	5	7	5	2	6
<b>GDE3</b>	2	7	13	2	2	12	10	13	4	8	7
<b>AMGA</b>	11	10	7	4	3	9	13	8	11	4	8
<b>MOEADGM</b>	3	2	4	10	14	14	3	12	10	11	8
<b>DECMOSA-SQP</b>	13	13	9	3	5	8	9	10	9	6	9
<b>MOEP</b>	12	11	10	7	8	7	6	14	14	5	9
<b>NSGAILS</b>	8	9	12	12	13	13	7	4	2	14	9
<b>Clustering MOEA</b>	10	12	6	13	9	6	8	11	13	7	10
<b>OWMOSaDE</b>	9	6	11	11	11	11	14	5	7	13	10
<b>OMOEAI</b>	14	14	14	9	6	4	11	9	12	12	11

Como pode ser observado, quando comparado aos algoritmos que participaram do CEC 2009, o OptBees-MO obteve um bom desempenho. Embora tenha sido superado pelo MOEA/D, seu desempenho na maioria dos problemas foi superior ao AMGA, que obteve o sétimo lugar na competição. Com isso, o OptBees-MO se mostrou superior à maioria dos algoritmos apresentados no evento.

## 6. Conclusão

Neste trabalho foi descrito a ideia e funcionamento básico do algoritmo OptBees – MO. O OptBees – MO foi proposto para a otimização multiobjetivo partindo-se da ideia que a manutenção de diversidade do OptBees poderia ser utilizada para a localização de soluções não dominadas enquanto converge para a fronteira Pareto. Para tanto, foi utilizada a função de *fitness* proposta no *SPEA-II* que procura concentrar a diversidade de uma solução e a sua dominância quanto aos objetivos num único valor. Outras alterações foram feitas no algoritmo de forma que seu bom desempenho fosse mantido.

Para testar o desempenho do algoritmo, foram utilizados os problemas propostos no CEC-2009 e o algoritmo foi comparado aos algoritmos que participaram da competição. Os testes preliminares apresentaram bons resultados quando comparados a outros algoritmos, o que significa que seus mecanismos de diversidade do OptBees permitem que esse encontre soluções não dominadas eficientemente. Em trabalhos futuros, visa-se realizar o teste com outras atribuições de *fitness* menos custosas e a adaptação do OptBees – MO para o tratamento de problemas com muitos objetivos.

## Agradecimentos

Os autores agradecem o apoio financeiro da Fundação de Amparo a Pesquisa do Estado de Minas Gerais – FAPEMIG.



## Referências

COELLO COELLO, Carlos A.; AGUIRRE, Arturo Hernández. Design of combinational logic circuits through an evolutionary multiobjective optimization approach. **AI EDAM**, v. 16, n. 01, p. 39-53, 2002.

GADAU, Jürgen; FEWELL, Jennifer; WILSON, Edward O. **Organization of insect societies: from genome to sociocomplexity**. Harvard University Press, 2009.

GAO, Song *et al.* An orthogonal multi-objective evolutionary algorithm with lower-dimensional crossover. In: **Evolutionary Computation, 2009. CEC'09. IEEE Congress on**. IEEE, 2009. p. 1959-1964.

MAIA, Renato Dourado. **Colônias de Abelhas como Modelo para Otimização Multimodal em Espaços Contínuos: uma Abordagem Baseada em Alocação de Tarefas**. 2012. Tese de Doutorado. doctoral dissertation, Universidade Federal de Minas Gerais, Engenharia Elétrica, Belo Horizonte.

MAIA, Renato Dourado *et al.* Collective decision-making by bee colonies as model for optimization-the OptBees algorithm. **Applied Mathematical Sciences**, v. 7, n. 87, p. 4327-4351, 2013.

MARTINEZ, José Mario; SANTOS, Sandra Augusta. Métodos computacionais de otimização. **Colóquio Brasileiro de Matemática, Apostilas**, v. 20, 1995.

PARETO, Vilfredo. **Cours d'économie politique**. Librairie Droz, 1896.

PÉREZ, Miguel Angel Fernández. **Um método heurístico para o problema de escalonamento multiobjetivo em vários ambientes de máquinas**. Diss. PUC-Rio, 2012.

TIWARI, Santosh *et al.* Performance assessment of the hybrid archive-based micro genetic algorithm (AMGA) on the CEC09 test problems. In: **Evolutionary Computation, 2009. CEC'09. IEEE Congress on**. IEEE, 2009. p. 1935-1942.

TSENG, Lin-Yu; CHEN, Chun. Multiple trajectory search for large scale global optimization. In: **Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on**. IEEE, 2008. p. 3052-3059.

ZHANG, Qingfu *et al.* Multiobjective optimization test instances for the CEC 2009 special session and competition. **University of Essex, Colchester, UK and Nanyang technological University, Singapore, special session on performance assessment of multi-objective optimization algorithms, technical report**, p. 1-30, 2008.

ZHANG, Qingfu; LIU, Wudong; LI, Hui. The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances. In: **IEEE Congress on Evolutionary Computation**. 2009. p. 203-208.

ZITZLER, E., LAUMANN, M., & THIELE, L. **SPEA2: Improving the strength Pareto evolutionary algorithm**. 2001.

ZITZLER, Eckart; THIELE, Lothar. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. **Evolutionary computation, IEEE transactions on**, v. 3, n. 4, p. 257-271, 1999.