COMPARAÇÃO DE TÉCNICAS DE PENALIZAÇÃO ADAPTATIVAS NO ALGORITMO COLÔNIA DE ABELHAS ARTIFICIAIS APLICADO A PROBLEMAS DE OTIMIZAÇÃO EM ENGENHARIA

GRASIELE REGINA DUARTE[1]

RESUMO

Este artigo avalia o desempenho da meta-heurística Algoritmo Colônia de Abelhas Artificiais (ABC), inspirada no comportamento social de forrageamento de colônias de abelhas reais, quando aplicada na resolução de problemas de otimização com restrições. Propõe-se neste trabalho a resolução de alguns problemas da Engenharia Mecânica encontrados na literatura. Para tratamento de violações às restrições dos problemas, combina-se ao ABC métodos de penalização adaptativos, que não dependem do usuário na definição de parâmetros de penalização. Realiza-se também um estudo comparativo entre os métodos de penalização aplicados, com o objetivo de avaliar a influência de cada um deles na convergência do algoritmo. Além disso, é proposto neste trabalho um método de penalização inspirado em um dos métodos estudados, objetivando melhoria de desempenho.

PALAVRAS-CHAVES: META-HEURÍSTICA. OTIMIZAÇÃO COM RESTRIÇÕES. MÉTODOS DE PENALIZAÇÃO ADAPTATIVOS.

COMPARISON OF TECHNIQUES FOR ADAPTIVE PENALTY IN ARTIFICIAL BEE COLONY ALGORITHM APPLIED TO PROBLEMS OF OPTIMIZATION IN ENGINEERING

ABSTRACT

This paper evaluates the performance of the meta-heuristic Artificial Bee Colony Algorithm (ABC), inspired by the social foraging behavior of colonies of real bees when applied in solving optimization problems with constraints. It is proposed in this work to solve some problems of Mechanical Engineering in the literature. For treatment of violations of the constraints of the problems, the ABC combines adaptive penalty methods, which do not depend on user definition of penalty parameters. Also carried out a comparative study of the methods of penalty applied, with the aim of evaluating the influence of each of them at the convergence of the algorithm. Furthermore, it is proposed in this paper a method of penalization inspired by one of the methods studied, aiming performance improvement.

KEYWORDS: META-HEURISTIC. OPTIMIZATION WITH CONSTRAINTS. ADAPTIVE PENALTY METHODS.

[1] Universidade Federal de Juiz de Fora; e-mail: grasird@yahoo.com.br

1. INTRODUÇÃO

Algoritmos inspirados na natureza têm sido propostos para resolução de problemas complexos de busca e otimização em diversas áreas das Ciências e Engenharias. Alguns exemplos desses algoritmos são os Algoritmos Genéticos, Algoritmos de Enxame de Partículas, Algoritmos de Formigas e os Algoritmos de Colônia de Abelhas Artificiais (ABC, do inglês *Artificial Bee Colony Algorithm*) [8].

O ABC, inspirado no comportamento social durante o forrageamento em colônias de abelhas reais, é um algoritmo baseado em Enxames Inteligentes, proposto por Karaboga [6] para a solução de problemas de otimização numérica multidimensional e multimodal. O ABC foi proposto inicialmente para resolução de problemas de otimização numérica sem restrições e tem-se mostrado competitivo com outras meta-heurísticas, por isso torna-se objeto de estudo neste trabalho.

Karaboga e Basturk [7] e Karaboga e Akay [10] propuseram, com excelentes resultados, uma modificação no ABC para resolução de problemas com restrições aplicando o critério de Deb [4]. Assim, promove-se seleção gulosa (*greedy*) por torneio entre soluções candidatas em que será selecionada a solução factível entre uma factível e outra infactível, a de melhor função objetivo entre soluções factíveis e a de menor violação entre soluções infactíveis. Tal estratégia também foi adotada por Sonmez [17] em problemas de otimização estrutural.

Brajevic; Tuba; Subotic [2] propôs uma estratégia em que na fase de exploração do ABC, qualquer solução infactível é substituída por uma nova solução produzida aleatoriamente, onde obtiveram bons resultados referente à convergência do algoritmo para soluções factíveis.

Propõe-se neste trabalho avaliação do ABC na resolução de problemas de otimização com restrições, tendo em vista que grande parte dos problemas de otimização numérica são desta classe. O tratamento a ser dado às violações de restrições será por meio da técnica de penalização. Será aplicada penalização adaptativa, que não depende do usuário na definição de parâmetros de penalização. Um método de penalização inspirado em um dos métodos estudados é proposto na tentativa de melhoria de desempenho após identificação de possíveis causas de limitações do método original. A performance do algoritmo será avaliada aplicando-o num conjunto de problemas de Engenharia Mecânica. Este trabalho também apresenta uma metodologia de comparação com outros resultados encontrados na literatura.

2. O ALGORITMO COLÔNIA DE ABELHAS ARTIFICIAIS (ABC)

O ABC, inspirado no forrageamento de colônias de abelhas reais, é um algoritmo proposto por Karaboga [6] para a solução de problemas de otimização numérica multidimensional e multimodal. No modelo do forrageamento de uma colônia de abelhas reais, estão presentes fontes de alimento, abelhas campeiras e abelhas não-campeiras, além de existir recrutamento de abelhas para exploração de fontes de alimento, bem como abandonos de fontes. Uma abelha campeira está relacionada a uma fonte de alimento com a finalidade de explorá-la [6]. Esta abelha, de posse de informações sobre a fonte, como localização, concentração de alimento, facilidade de extração, dentre outras, então as divide com outras abelhas da colônia. Sendo uma boa fonte de alimento, abelhas da colônia serão atraídas, resultando na exploração rápida do alimento. Fontes não atraentes acabam não sendo selecionadas pelas abelhas e tornam-se abandonadas [5,6]. Abelhas não-campeiras podem ser exploradoras, que buscam por novas fontes de alimento ao redor da colônia ou observadoras, que aguardam na colônia por informações das abelhas campeiras para então avaliar e selecionar com base na qualidade da fonte. Tanto abelhas campeiras quanto observadoras podem explorar ao redor da fonte conhecida, o que possibilita que sejam encontradas fontes ainda melhores. A forma de comunicação entre as abelhas no forrageamento é através de uma dança no ninho [5,6].

No ABC, as fontes de alimento representam possíveis soluções para o problema tratado, estando presentes em ciclos repetitivos, os tipos de abelhas mencionados. A cada fonte de alimento está associada uma abelha campeira artificial, sendo que metade da colônia é composta por este tipo de abelha e a outra metade por abelhas observadoras. Fontes esgotadas depois de um certo número de ciclos serão substituídas por outras, produzidas aleatoriamente por abelhas campeiras que passam a ser exploradoras [8]. Desta forma, o ABC simula uma colônia de abelhas artificiais praticando o forrageamento, na busca por uma solução ótima para o problema tratado. A avaliação da fonte artificial se dá pela qualidade da solução para atender o problema [6]. Os principais passos do ABC podem ser descritos no Algoritmo (1) [9] apresentado no QUADRO 1.

Quadro 1: Algoritmo 1 – ABC

Início	Inicialize a população;
Repita	- Posicione as abelhas campeiras em suas fontes de alimento; - Posicione as abelhas observadoras nas fontes de alimento dependendo de seus valores; - Envie abelhas exploradoras para buscar novas fontes de alimento; - Memorize a melhor fonte de alimento encontrada até o momento;
Até	Número máximo de ciclos;
Fim	

No Algoritmo (1), a população inicial é composta por $SN/2$ soluções (fontes de alimento), onde SN é o tamanho da população de abelhas. Cada solução x_i ($i = 1, 2, \ldots, SN/2$) é um vetor $D-\text{dimensional}$, em que D é o número de parâmetros do problema. Iniciada a população, serão repetidos ciclos $C = 1, 2, \ldots, C_{max}$ do processo de busca das abelhas campeiras, observadoras e exploradoras. No modelo artificial a melhoria de soluções (fontes de alimento) por abelhas campeiras ou observadoras é baseada no processo de comparação, portanto as abelhas selecionam aleatoriamente uma fonte de alimento e produzem uma modificação de acordo com a Equação (1):

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}), \tag{1}$$

onde v_{ij} é o novo valor candidato a substituir x_{ij}, $k \in \{1, 2, \ldots, BN\}$ e $j \in \{1, 2, \ldots, D\}$ são índices escolhidos aleatoriamente, sendo que $k \neq i$, BN é o número de abelhas campeiras e $\emptyset_{ij}$ é um valor aleatório entre $[-1, 1]$. A Equação (1) mostra que como a busca aproxima da solução ótima, a diferença entre x_{ij} e x_{kj} e a perturbação na solução x_{ij} diminuem, então o tamanho do passo da modificação é adaptativamente reduzido. Se o valor produzido pela Equação (1) violar os limites pré-definidos para o parâmetro x_{ij}, este deverá ser ajustado para um valor aceitável. Tendo avaliado a qualidade da nova solução (v_i), e sendo esta melhor que a antiga (x_i), a solução será então substituída, caso contrário será mantida. Completado o processo de busca, todas as abelhas campeiras compartilharão informações sobre a fonte de alimento. Cada abelha observadora escolherá uma fonte (solução) de acordo com uma probabilidade p_i relacionada à sua qualidade, calculada por

$$p_i = \frac{fit_i}{\sum_{n=1}^{BN} fit_n}, \tag{2}$$

onde fit_i é o valor da aptidão da solução i [8], determinada pela equação

$$fit_i = \begin{cases} 1/(1+f_i), & se\ f_i \geq 0 \\ 1+|f_i|, & se\ f_i < 0 \end{cases}, \tag{3}$$

onde f_i é o valor da função objetivo para a solução x_i [10]. No ABC, se uma solução não pôde ser melhorada por um determinado número de ciclos chamado $limit$, então esta

será considerada como abandonada e será substituída por outra, descoberta aleatoriamente por uma abelha exploradora artificial [8].

3. MÉTODOS DE PENALIZAÇÃO

Têm por objetivo aproximar soluções de problemas com restrições por problemas irrestritos. São aplicados em problemas de otimização não-linear, contendo restrições de igualdade e/ou desigualdade. Denomina-se solução factível aquela que atende as restrições impostas ao problema, o contrário torna a solução infactível. A penalização se dá na forma de um acréscimo (Penalização Aditiva) ou amplificação (Penalização Multiplicativa) ao valor da função objetivo caso a solução seja infactível. As restrições são colocadas na função objetivo de forma a penalizar qualquer violação [1,12].

Dados problemas na forma

$$\begin{cases} \text{Minimize} & f(\mathbf{x}) \\ \text{Sujeito a} & g_i(\mathbf{x}) \leq 0; \quad h_j(\mathbf{x}) = 0; \quad \mathbf{x}_I \leq \mathbf{x} \leq \mathbf{x}_S \end{cases}, \tag{4}$$

onde $i = 1, 2, \ldots, m, j = 1, 2, \ldots, q, \boldsymbol{x}, \boldsymbol{x}_I$ e $\boldsymbol{x}_S$ são vetores n-dimensionais representando respectivamente as variáveis de projeto e os limites inferior e superior destas mesmas variáveis, aplicando o método de penalização (Penalização Aditiva), passa-se a considerar o problema Equação (4) reescrito tipicamente na forma da Equação (5).

$$\begin{cases} \text{Minimize} & f(\mathbf{x}) + k\left\{\sum_{i=1}^{m} max\,[0, g_i(\mathbf{x})]^p + \sum_{i=1}^{q} |h_i(\mathbf{x})|^p\right\} \\ \text{Sujeito a} & \mathbf{x} \in \mathbb{R}^n \end{cases}, \tag{5}$$

onde $k > 0$, é um número suficientemente grande e p é um inteiro positivo [1]. Solucionando o problema Equação (5), aproxima-se da solução do problema Equação (4).

Acompanhando o método, o parâmetro k define a severidade da penalização [1,12]. Caso o parâmetro k não sofra alterações durante o processo de busca, o método de penalização será dito estático, caso contrário será dito dinâmico [11].

3.1 Métodos de Penalização Adaptativos

Claramente o ajuste do parâmetro k presente na Equação (5), não consiste de uma tarefa trivial quando realizada pelo usuário, seja aplicando penalização estática ou dinâmica. Se k for pequeno, soluções infactíveis com aptidão sobre o valor penalizado, maior que a aptidão de uma solução ótima factível, pode provocar convergência para uma região não atrativa no processo de busca. Por outro lado, se k for grande, boas característica de soluções infactíveis não terão chance de serem aproveitadas, podendo provocar convergência prematura para solução não tão boa [14].

Os Métodos de Penalização Adaptativos possibilitam alteração do parâmetro k durante as iterações do processo de busca sem a necessidade de intervenção do usuário. Tal possibilidade, se deve ao fato de tais métodos atualizarem o valor de k com base em informações extraídas do próprio conjunto de soluções candidatas disponível no momento [11]. Desta forma, os parâmetros se adaptam às exigências de severidade impostas pelo próprio conjunto de soluções à medida que se busca a solução para o problema.

Foi proposto por Coit; Smith; Tate [3], um método de penalização, identificado como Coit neste trabalho, pelo qual a função objetivo ao converter o problema Equação (4) para um problema irrestrito será dada pela Equação (6):

$$F_p(\mathbf{x}) = f(\mathbf{x}) - (F_{all} - F_{feas}) \sum_{i=1}^{m+q} \left[\frac{d_i(\mathbf{x})}{NFT_i} \right]^{k_i}, \tag{6}$$

onde $f(\boldsymbol{x})$ equivale à função objetivo não penalizada, F_{all} é o valor não penalizado da função objetivo da melhor solução já encontrada e F_{feas} denota o valor não penalizado da função objetivo da melhor solução factível encontrada. Não havendo soluções factíveis, Coit; Smith; Tate [3] define que uma solução infactível próxima à região factível é equivalente a uma solução factível para determinação de F_{feas}. $d(\boldsymbol{x})$ é uma função que descreve a distância de $\boldsymbol{x}$ até a região factível. Desta forma $d_i(\boldsymbol{x})$ pode ser vista como a medida de violação a cada uma das restrições i impostas ao problema pela solução $\boldsymbol{x}$. Uma definição adequada para $d(\boldsymbol{x})$ pode ser dada por:

$$d(\mathbf{x}) = \sum_{i=1}^{m} max\,[0, g_i(\mathbf{x})] + \sum_{i=1}^{q} |h_i(\mathbf{x})|\,. \tag{7}$$

Ainda na Equação (6), NFT_i (do inglês *Near-Feasibility Threshold*) consiste de um limiar próximo à região factível referente à restrição i, cujo uso tornaria a busca pela solução atrativa para uma direção quando aplicado tal valor. Neste caso, pode-se

considerar como uma boa escolha para NFT_i o valor da menor violação para a restrição i.

Ainda é proposto em Coit; Smith; Tate [3], ao exemplificar aplicação de seu método proposto, uma alteração na Equação (6), identificada como Coit Min neste trabalho, dada por

$$F_p(\mathbf{x}) = f(\mathbf{x}) + (F_{feas} - F_{all}) \left(\frac{\eta}{NFT}\right)^k, \tag{8}$$

em que F_{all} e F_{feas} tiveram posições invertidas na equação, η corresponde ao número de restrições violadas por $\boldsymbol{x}$. Na Equação (8), escolhe para NFT a violação total às restrições pela solução candidata $\boldsymbol{x}$ [3].

Ao propor um método para manipulação de restrições já mencionado na Seção (1), que independe de parâmetros, Deb [4] argumentou que a partir do momento que se sabe que uma solução candidata trata-se de uma solução infactível (violando pelo menos uma restrição) não faz sentido calcular a função objetivo, visto que tal solução não poderá ser implementada na prática. Com base em tal argumento, Deb [4] propõe também uma função para avaliação de soluções candidatas para o problema Equação (4), titulada Deb neste trabalho. Por este método, a função objetivo ao converter o problema Equação (4) para um problema irrestrito será dada por

$$F_p(\mathbf{x}) = \begin{cases} f(\mathbf{x}), & \text{se } g_i(\mathbf{x}) \leq 0,\ i = 1,2,...,m \text{ e } h_j(\mathbf{x}) = 0,\ j = 1,2,...,q \\ f_{max} + \sum_{i=1}^{m} max\,[0, g_i(\mathbf{x})] + \sum_{j=1}^{q} |h_j(\mathbf{x})|, & \text{outro caso,} \end{cases} \tag{9}$$

onde f_{max} corresponde ao valor da função objetivo da pior solução factível presente no conjunto de soluções candidatas. Desta forma a avaliação de uma solução candidata infactível não depende apenas da quantidade de violações, mas também do conjunto de soluções. Deb [4]define ainda que não existindo soluções factíveis no conjunto, f_{max} será igual a 0.

Montemurro; Vincenti; Vannucci [13] propôs uma estratégia de penalização titulada ADP (do inglês *Automatic Dynamic Penalisation*), na qual estão presentes parâmetros de penalização relacionados a cada uma das restrições impostas. Aplicando o ADP, Montemurro; Vincenti; Vannucci [13] propõe que o problema Equação (4) ao ser convertido para um problema irrestrito tenha função objetivo dada por

$$F_p(\mathbf{x}) = \begin{cases} f(\mathbf{x}), & \text{se } g_i(\mathbf{x}) \leq 0,\ i = 1,2,...,m \text{ e } h_j(\mathbf{x}) = 0,\ j = 1,2,...,q \\ f(\mathbf{x}) + \sum_{i=1}^{m} c_i max[0, g_i(x)] + \sum_{j=1}^{q} r_j |h_j(\mathbf{x})|, & \text{outro caso} \end{cases}, \quad (10)$$

onde c_i e r_j são os parâmetros de penalização relacionados a cada uma das restrições, cujos ajustes a cada iteração do processo de busca são dados por

$$\begin{aligned} c_i &= |f_{best}^F - f_{best}^{NF}| \ / \ (G_i)_{best}^{NF},\ i = 1,2,...,m \\ r_j &= |f_{best}^F - f_{best}^{NF}| \ / \ (H_j)_{best}^{NF},\ j = 1,2,...,q \end{aligned}, \quad (11)$$

onde F e NF significam factível e não-factível respectivamente, f_{best}^F e f_{best}^{NF} são os valores da função objetivo da melhor solução factível e infactível respectivamente, enquanto $(G_i)_{best}^{NF}$ e $\left(H_j\right)_{best}^{NF}$ representam as violações da melhor solução infactível em relação a cada uma das restrições. No caso de não haver soluções factíveis, aquelas que tiverem pequenos valores de violação para uma determinada restrição serão agrupadas num sub-grupo de soluções virtualmente factíveis, de onde deve sair o valor f_{best}^F, enquanto o resto será considerado realmente infactível. Montemurro; Vincenti; Vannucci [13] define que o número de soluções virtualmente factíveis corresponde a 10% do conjunto de soluções candidatas.

4. EXPERIMENTOS NUMÉRICOS

Avalia-se neste trabalho a performance do algoritmo ABC combinado a métodos de penalização, aplicado a um grupo de problemas de otimização com restrições da Engenharia Mecânica amplamente tratado na literatura.

Destaca-se que quanto às restrições de igualdade, estas foram transformadas em restrições de desigualdade adotando a forma

$$|h_j(\mathbf{x})| - \epsilon \leq 0,\ j = 1,2,...,q, \quad (12)$$

que é numericamente aceitável para uma tolerância ϵ de valor suficientemente pequena, de forma a não afetar a qualidade da solução do problema. Neste trabalho foi considerado $\epsilon = 10^{-6}$.

Referente aos parâmetros do algoritmo ABC, neste trabalho foram definidos $SN = 100$, $C_{max} = 1000$ e $limit = 50$. Quanto aos métodos de penalização, aplicou-se os métodos propostos em Coit; Smith; Tate [3] (Coit Min), Deb [4] (Deb) e Montemurro; Vincenti; Vannucci [13] (ADP), descritos na Seção (3.1), e também o método Coit Mod, proposto neste trabalho, descrito na Seção (4.6). Para o método Coit Min, o expotente k, presente na Equação (8) foi fixado em 2. Para cada problema descrito a seguir, foram realizadas 25 execuções da combinação ABC + Método de Penalização.

4.1 P1 -- Redutor de Velocidade

Este problema consiste do projeto de um redutor de velocidade, com sete variáveis de projeto, cuja solução ótima conhecida é $\boldsymbol{x}^* = \{3{,}5;\ 0{,}7;\ 17{,}0;\ 7{,}3;\ 7{,}3;\ 3{,}35;\ 5{,}29\}^T$, $f^* = 2985{,}22$. O objetivo é minimizar o peso do redutor. As restrições são referentes a limitações de tensão e deflexão [16]. O problema é dado por

$$
\mathbf{P1}\left\{\begin{array}{ll}
\text{Minimize} & f(\mathbf{x}) = 0,7854x_1x_2^2(3,3333x_3^2 + 14,9334x_3 - 43,0934) - 1,508x_1(x_6^2 + x_7^2) + \\
 & \qquad 7,477(x_6^3 + x_7^3) + 0,7854(x_4x_6^2 + x_5x_7^2) \\
\text{Sujeito a} & g_1(\mathbf{x}) = 27x_1^{-1}x_2^{-2}x_3^{-2} - 1 \leq 0 \\
 & g_2(\mathbf{x}) = 397,5x_1^{-1}x_2^{-2}x_3^{-2} - 1 \leq 0 \\
 & g_3(\mathbf{x}) = 1,93x_2^{-1}x_3^{-1}x_4^3x_6^{-4} - 1 \leq 0 \\
 & g_4(\mathbf{x}) = 1,93x_2^{-1}x_3^{-1}x_5^3x_7^{-4} - 1 \leq 0 \\
 & g_5(\mathbf{x}) = \left[\left(\frac{745x_4}{x_2x_3}\right)^2 + (16,9)10^6\right]^{0,5} /0,1x_6^3 - 1100 \leq 0 \\
 & g_6(\mathbf{x}) = \left[\left(\frac{745x_5}{x_2x_3}\right)^2 + (157,5)10^6\right]^{0,5} /0,1x_7^3 - 850 \leq 0 \\
 & g_7(\mathbf{x}) = x_2x_3 - 40 \leq 0 \\
 & g_8(\mathbf{x}) = 5 - \frac{x_1}{x_2} \leq 0 \\
 & g_9(\mathbf{x}) = \frac{x_1}{x_2} - 12 \leq 0 \\
 & g_{10}(\mathbf{x}) = (1,5x_6 + 1,9)x_4^{-1} - 1 \leq 0 \\
 & g_{11}(\mathbf{x}) = (1,1x_7 + 1,9)x_5^{-1} - 1 \leq 0 \\
 & 2,6 \leq x_1 \leq 3,6, \quad 0,7 \leq x_2 \leq 0,8, \quad 17 \leq x_3 \leq 28, \\
 & 7,3 \leq x_4, x_5 \leq 8,3, \quad 2,9 \leq x_6 \leq 3,9, \quad 5,0 \leq x_7 \leq 5,5
\end{array}\right. \tag{13}
$$

3.2 P2 -- Treliça de Três Barras

Este experimento corresponde ao projeto ótimo de uma treliça plana de três barras, cuja solução ótima conhecida é $\boldsymbol{x}^* = \{0{,}78706;\ 0{,}40735\}^T$, $f^* = 263{,}35$ [16]. O objetivo é minimizar as áreas das seções transversais das barras, sujeitas a restrições de tensão (σ) [16,18]. O problema é formulado por

$$\mathbf{P2}\begin{cases} \text{Minimize} & f(\mathbf{x}) = (2\sqrt{2x_1} + x_2) \times l \\ \text{Sujeito a} & g_1(\mathbf{x}) = \dfrac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \leq 0 \\ & g_2(\mathbf{x}) = \dfrac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \leq 0 \\ & g_3(\mathbf{x}) = \dfrac{1}{x_1 + \sqrt{2}x_2}P - \sigma \leq 0 \\ & 0 \leq x_1, x_2 \leq 1 \end{cases} \tag{14}$$

onde $\mathrm{l} = 100\ \mathrm{cm}$, $\mathrm{P} = 2\mathrm{kN/cm^2}$ e $\sigma = 2\mathrm{kN/cm^2}$.

3.3 P3 -- Trocador de calor

Rao [16] exemplifica o projeto de um equipamento trocador de calor, com 8 variáveis de projeto, cuja solução ótima conhecida é $\boldsymbol{x}^* = \{567;\ 1357;\ 5125;\ 181;\ 295;\ 219;\ 286;\ 395\}^T$, $f^* = 7049$. O problema é dado por

$$\mathbf{P3}\begin{cases} \text{Minimize} & f(\mathbf{x}) = x_1 + x_2 + x_3 \\ \text{Sujeito a} & g_1(\mathbf{x}) = 0{,}0025(x_4 + x_6) - 1 \leq 0 \\ & g_2(\mathbf{x}) = 0{,}0025(-x_4 + x_5 + x_7) - 1 \leq 0 \\ & g_3(\mathbf{x}) = 0{,}01(-x_5 + x_8) - 1 \leq 0 \\ & g_4(\mathbf{x}) = 100x_1 - x_1x_6 + 833{,}33252x_4 - 83333{,}333 \leq 0 \\ & g_5(\mathbf{x}) = x_2x_4 - x_2x_7 - 1250x_4 + 1250x_5 \leq 0 \\ & g_6(\mathbf{x}) = x_3x_5 - x_3x_8 - 2500x_5 + 1250000 \leq 0 \\ & 100 \leq x_1 \leq 10000, \quad 1000 \leq x_i \leq 10000\ \ (i = 2, 3) \\ & 10 \leq x_i \leq 1000\ \ (i = 4, \ldots, 8) \end{cases} \tag{15}$$

3.4 P4 -- Vaso de Pressão

Este experimento corresponde à minimização do peso de um vaso de pressão cilíndrico. O problema envolve quatro variáveis de projeto, sendo duas discretas e duas contínuas [11]. A formulação do problema se dá por

$$\mathbf{P4}\begin{cases} \text{Minimize} & f(\mathbf{x}) = 0,6224x_1x_2x_3 + 1,7781x_2x_3^2 + 3,1661x_1^2x_4 + 19,84x_1^2x_3 \\ \text{Sujeito a} & g_1(\mathbf{x}) = 0,0193x_3 - x_1 \leq 0 \\ & g_2(\mathbf{x}) = 0,00954x_3 - x_2 \leq 0 \\ & g_3(\mathbf{x}) = 1296000 - \pi x_3^2 x_4 - 4/3\pi x_3^3 \leq 0 \\ & g_4(\mathbf{x}) = x_4 - 240 \leq 0 \\ & 0,0625 \leq x_1, x_2 \leq 5 \ \text{(em passos constantes de } 0,0625) \\ & 10 \leq x_3, x_4 \leq 200 \end{cases} \quad (16)$$

3.5 P5 -- Viga Engastada

Este problema corresponde ao projeto de uma viga engastada, cuja solução ótima conhecida é $\boldsymbol{x}^* = \{0{,}2444;\ 6{,}2177;\ 8{,}2915;\ 0{,}2444\}^T$, $f^* = 2{,}3810$. O objetivo é minimizar o custo da viga, sujeita a restrições envolvendo limites de tensão e deslocamento [11,16]. O problema envolve quatro variáveis de projeto e é dado por

$$\mathbf{P5}\begin{cases} \text{Minimize} & f(\mathbf{x}) = 1,10471x_1^2x_2 + 0,04811x_3x_4(14,0 + x_2) \\ \text{Sujeito a} & g_1(\mathbf{x}) = \tau - 13600 \leq 0 \\ & g_2(\mathbf{x}) = \sigma - 30000 \leq 0 \\ & g_3(\mathbf{x}) = x_1 - x_4 \leq 0 \\ & g_4(\mathbf{x}) = 6000 - P_c \leq 0 \\ & g_5(\mathbf{x}) = \delta - 0,25 \leq 0 \\ & 0,125 \leq x_1 \leq 10 \\ & 0,1 \leq x_2, x_3, x_4 \leq 10 \end{cases} \quad (17)$$

onde

$$\tau' = \frac{6000}{\sqrt{2}x_1x_2}, \quad \tau'' = \frac{6000(14+0,5x_2)\alpha}{2\left\{0,707x_1x_2\left[x_2^2/12+0,25(x_1+x_3)^2\right]\right\}}$$
$$\alpha = \sqrt{0,25\left[x_2^2+(x_1+x_3)^2\right]}, \quad \tau = \sqrt{(\tau')^2+(\tau'')^2+x_2\tau'\tau''/\alpha} \tag{18}$$
$$\sigma = \frac{504000}{x_3^2x_4}, \quad P_c = 64746,022(1-0,0282346x_3)x_3x_4^3, \quad \delta = \frac{2,1952}{x_3^3x_4}.$$

3.6 Alteração Proposta

Tendo em vista que todos os problemas tratados neste trabalho são de minimização, deve-se considerar a possibilidade dos métodos propostos por Coit; Smith; Tate [3] (Coit e Coit Min) gerarem valores negativos nos termos $(F_{all} - F_{feas})$ e $(F_{feas} - F_{all})$ em Equação (6) e Equação (8), se $F_{all} < F_{feas}$ ou $F_{feas} < F_{all}$ respectivamente. Caso isso aconteça, o valor da função objetivo de uma solução candidata infactível, será reduzido e não acrescido como é esperado. Sendo aplicada uma penalização negativa, cresce a chance de uma solução infactível ser vista como uma boa solução, além de fazer com que possíveis novas soluções geradas pelo ABC sejam mantidas na região infactível. A possibilidade dos termos $(F_{all} - F_{feas})$ e $(F_{feas} - F_{all})$ em Equação (6) e Equação (8) resultarem em 0, quando $F_{all} = F_{feas}$, também deve ser considerada. Este resultado faz com que soluções infactíveis não sejam penalizadas, ainda que tenham valores altíssimos de violação. Considerando essas possibilidades, propõe-se neste trabalho uma alteração nos métodos propostos por Coit; Smith; Tate [3], identificada como Coit Mod, dada por

$$F_p(\mathbf{x}) = \begin{cases} f(\mathbf{x}) + |F_{all} - F_{feas}| \sum_{i=1}^{m+q} \left[\frac{d_i(\mathbf{x})}{NFT_i}\right]^{k_i}, & se\ F_{all} \neq F_{feas} \\ f(\mathbf{x}) + |F_{all} \sum_{i=1}^{m+q} NFT_i - F_{feas}| \sum_{i=1}^{m+q} \left[\frac{d_i(\mathbf{x})}{NFT_i}\right]^{k_i}, & se\ F_{all} = F_{feas} \end{cases}, \tag{19}$$

onde as características da Equação (6), propostas por Coit; Smith; Tate [3] são mantidas, com exceção dos termos na Equação (19) que substituem $F_{all} - F_{feas}$ da Equação (6), garantindo assim que os parâmetros de penalização se mantenham positivos ao serem ajustados durante o processo de busca.

4. RESULTADOS

Esta seção apresenta resultados referentes a cada um dos métodos de penalização, bem como resultados comparativos entre estes, quando combinados com o ABC na resolução de cada um dos problemas mencionados na Seção (4).

As FIGURAS 1-5 apresentam (em escala logarítmica) os gráficos de convergência para cada um dos problemas tratados, distinguindo entre os métodos de penalização. Cada gráfico apresenta a média dos valores da função objetivo da última iteração do ABC na totalidade de execuções de cada experimento. O eixo secundário (parte superior do gráfico) identifica quantas das soluções finais são factíveis em relação ao número de execuções. Observando os gráficos, constata-se que os métodos ADP, Coit Min, Coit Mod e Deb proporcionam os melhores resultados quanto à convergência do algoritmo ABC para soluções factíveis ao fim do processo.

Os métodos Coit Min e Coit Mod não contabilizam soluções factíveis em apenas um dos problemas tratados.

A condição para que uma solução fosse considerada factível dentre as soluções encontradas, foi a penalização nula. Tendo por base esta condição, destaca-se que foi observado que algumas das soluções factíveis, identificadas nas totalizações da FIGURA 1 à FIGURA 5, foram assim consideradas de forma incorreta.

As incorreções quanto à condição de factibilidade de soluções ocorreram ao aplicar o método ADP, onde em alguma iteração ocorria $f_{best}^{F} = f_{best}^{NF}$, e também ao aplicar o método Coit Min, onde ocorria $F_{all} = F_{feas}$ para alguma solução durante as iterações do algoritmo. Nestas condições, estes métodos deixam de penalizar as soluções infactíveis, fazendo-as parecer factíveis. Assim, dentre os métodos que contribuíram com o ABC na convergência para soluções factíveis, os métodos Coit Mod proposto neste trabalho e Deb, são os únicos que contabilizam soluções factíveis de forma correta.

Ainda que os resultados apresentados nas *boxplots* demonstrem que algumas das melhores soluções nas últimas iterações do ABC não sejam factíveis, não significa que durante execução do algoritmo este tipo de solução não tenha sido avaliada.

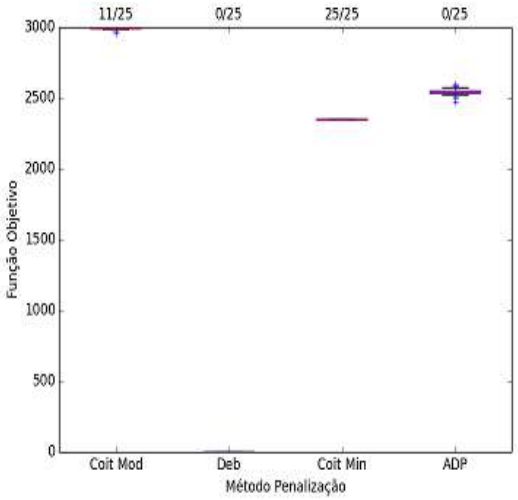

Figura 1: Resultados finais - P1

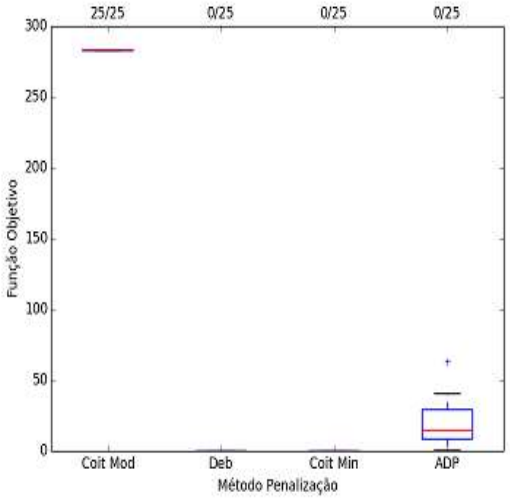

Figura 2: Resultados finais – P2

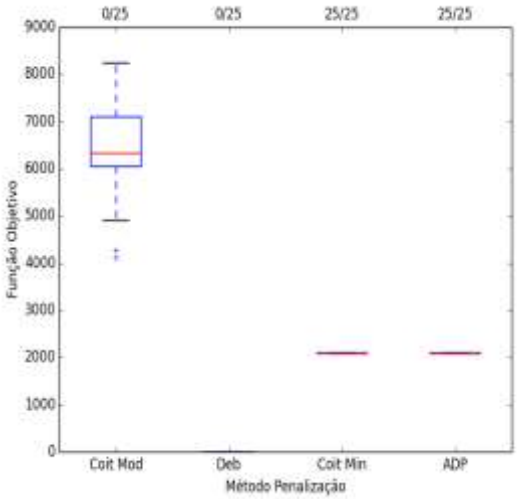

Figura 3: Resultados finais – P3

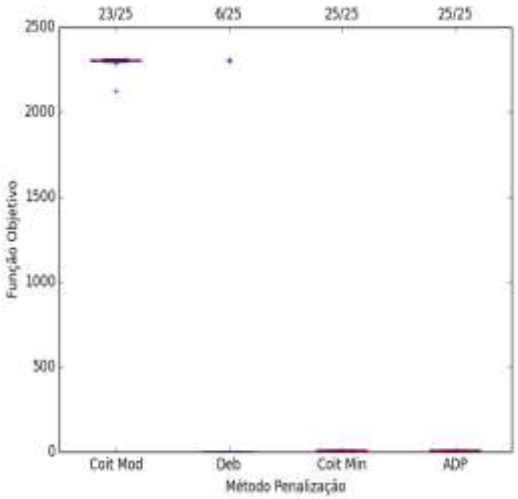

Figura 4: Resultados finais – P4

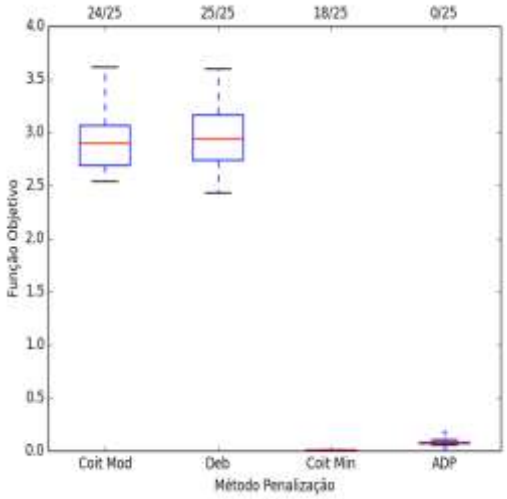

Figura 5: Resultados finais – P5

A TABELA 1 apresenta valores de função objetivo das soluções factíveis identificadas entre as melhores obtidas ao final de cada ciclo do ABC para cada problema. Casos em que não foram obtidas soluções factíveis, identifica-se por SFNE (Solução Factível Não Encontrada). Novamente soluções finais geradas por pelos métodos ADO e Coit foram identificadas como factíveis de forma incorreta. Soluções infactíveis identificadas como factíveis encontram-se destacadas em negrito na TABELA 1. Visto que os resultados apresentados nesta tabela são identificados considerando todo o processo de busca, nesse conjunto de soluções candidatas poderiam haver valores não tão atraentes para a função objetivo, mas que seriam realmente factíveis. Este tipo de resultado incorreto é uma consequência do método de penalização aplicado. Observando a TABELA 1, considerando apenas resultados realmente factíveis, o melhor resultado obtido para o problema P1 foi aplicando o método de

penalização Coit Mod, adotando como melhor solução $\boldsymbol{x} = \{3{,}5;\ 0{,}7;\ 17{,}0;\ 7{,}3;\ 7{,}7;\ 3{,}4;\ 5{,}3\}$. Para P2, o melhor resultado foi obtido aplicando os métodos de penalização Coit Mod e Deb, adotando como melhor solução $\boldsymbol{x} = \{1;\ 10^{-10}\}$ em ambos os casos. Para o problema P3 o método Coit Mod foi o único que contribuiu com o ABC na produção de solução realmente factível, sendo considerada como a melhor solução:

$$\boldsymbol{x} = \{879{,}6;\ 1339{,}7;\ 5757{,}5;\ 171{,}2;\ 271{,}9;\ 216{,}1;\ 279{,}9;\ 371{,}7\}.$$

Para o problema P4, o método que melhor contribuiu com o ABC foi o método Coit Mod, produzindo a melhor solução factível $\boldsymbol{x} = \{0{,}8125;\ 0{,}4375;\ 41{,}4;\ 200{,}0\}$. Finalmente, o melhor resultado obtido para P5 foi aplicando o método de penalização Deb, tendo como melhor solução $\boldsymbol{x} = \{0{,}231;\ 6{,}648;\ 8{,}424;\ 0{,}244\}$.

Tabela 1: Melhores Soluções Factíveis do Processo de Busca

Método Penalização	P1	P2	P3	P4	P5
Coit Min	**2352.3430**	SFNE	**2100.0000**	**0012.0361**	**0.0085**
Coit Mod	2994.3410	282.8427	7976.8430	2298.8930	2.5495
Deb	5019.0410	282.8427	SFNE	2299.2320	2.4321
ADP	SFNE	SFNE	**2100.0000**	**0012.0361**	4.2386

Os resultados apresentados da FIGURA 6 à FIGURA 10 ilustram a evolução entre as iterações do ABC, das médias entre as 25 execuções, dos valores da função objetivo de cada um dos problemas tratados. De maneira geral, os gráficos demonstram resultado satisfatório quanto à aplicação do ABC + Método de Penalização Adaptativo, visto que pode ser observado na maior parte dos casos, o comportamento esperado de um algoritmo de busca quando aplicado a um problema de minimização, onde os valores da função objetivo devem diminuir com o andamento do processo.

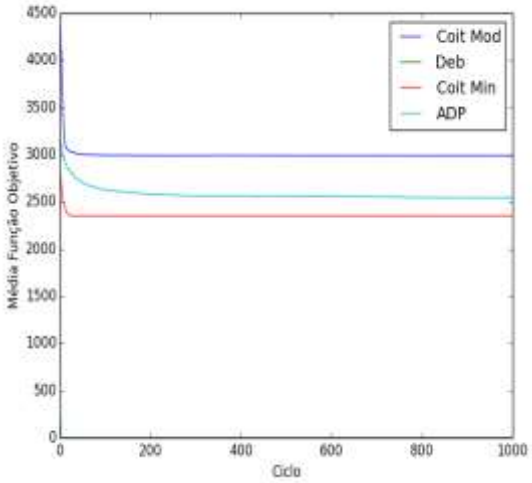

Figura 6: Evolução da Função Objetivo - P1

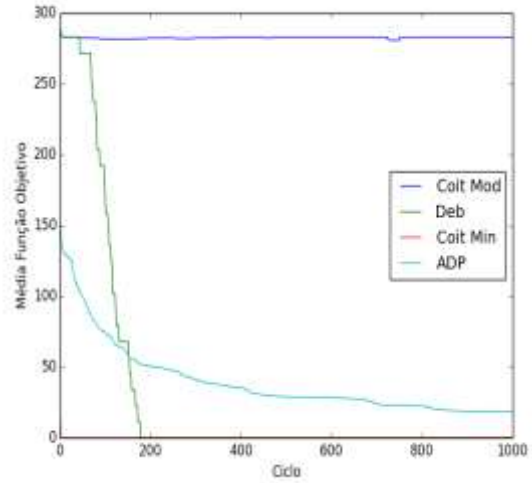

Figura 7: Evolução da Função Objetivo – P2

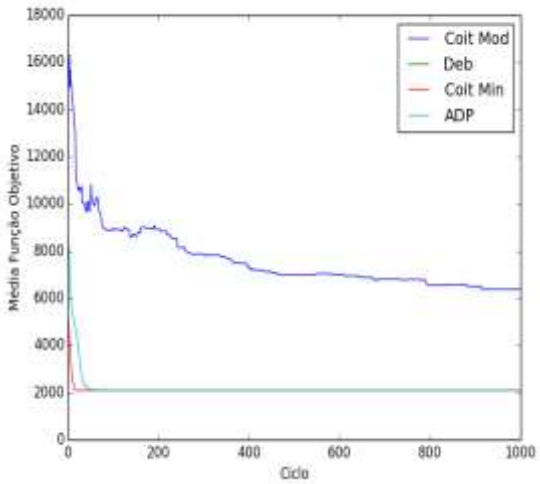

Figura 8: Evolução da Função Objetivo – P3

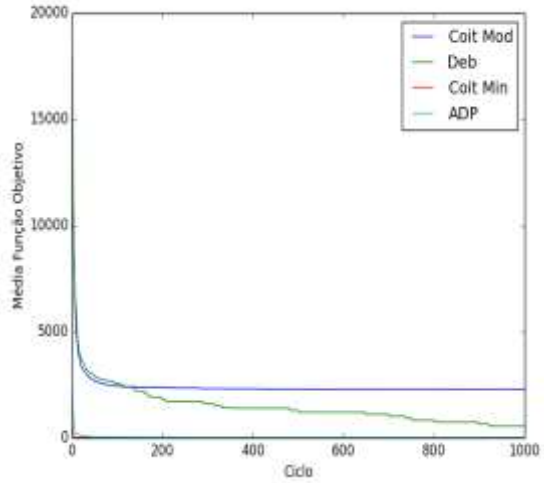

Figura 9: Evolução da Função Objetivo – P4

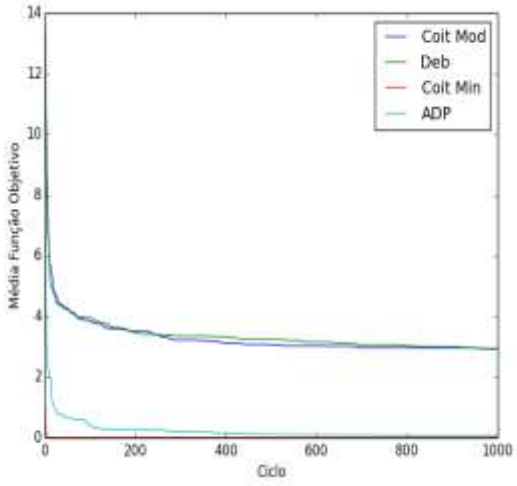

Figura 10: Evolução da Função Objetivo – P5

Para melhor análise comparativa entre os métodos de penalização tratados neste trabalho, optou-se pela técnica de visualização denominada Escalonamento Multidimensional (MDS - do inglês *Multidimensional Scaling*). O MDS consiste de uma técnica para medir a similaridade entre objetos de um conjunto, produzindo uma distribuição visual em um espaço multidimensional, em que objetos semelhantes encontram-se próximos, tornando-se distantes aqueles dissimilares Nevo, S.; Nevo, D.; Ein-Dor [15]. Da FIGURA (11) à FIGURA (15) são apresentados resultados obtidos por MDS para cada problema. Para aplicação da técnica, entre as 25 execuções foram considerados da última iteração do ABC, os menores valores de função objetivo (penalizada), quantidade de soluções factíveis, média, desvio padrão e mediana, todos normalizados entre 0 e 1. As similaridades entre os métodos tiveram redução de 5 para 2 dimensões. Como pode ser observado, os métodos Coit Mod e Deb, que identificam

corretamente a condição de factibilidade das soluções encontradas, tendem a se posicionar consideravelmente distante se comparado com os métodos. O posicionamento dos métodos Coit Min e ADP para constatação de similaridade entre os métodos não é confiável, visto que estes identificam incorretamente a condição de factibilidade de soluções discutida anteriormente. Assim, ainda que os métodos Coit Min e ADP se posicionem próximos ao Coit Mod, considerado o melhor método de penalização neste trabalho, não significa que estes sejam realmente melhores que o método Deb. Neste trabalho o objetivo com a técnica MDS é sintetizar informações para maiores conclusões. Pode ser percebida a divisão entre os métodos em grupos distintos, onde encontram-se próximos aqueles com resultados similares quanto à convergência dos métodos, principalmente quanto à convergência para soluções factíveis, ainda que estas sejam assim consideradas de forma incorreta pelos métodos.

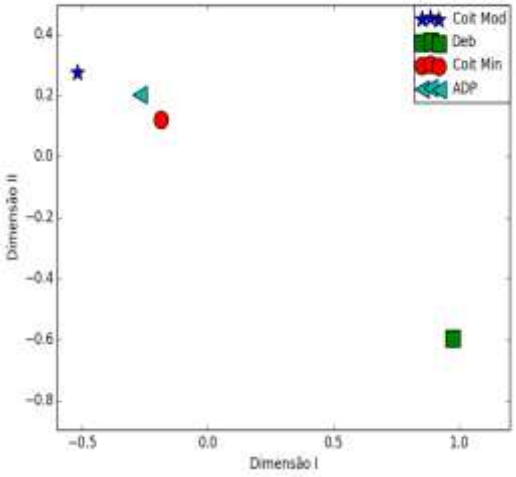

Figura 11: MDS - P1

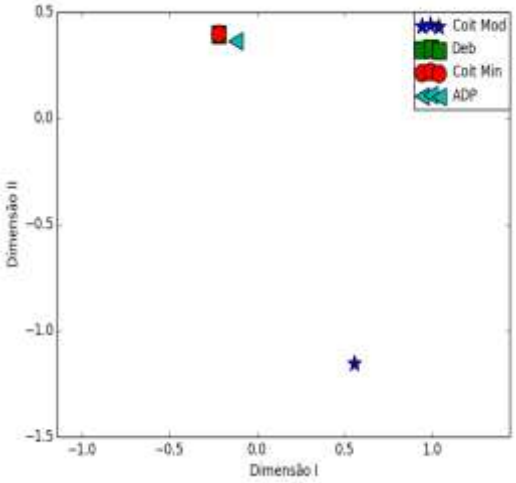

Figura 12: MDS - P2

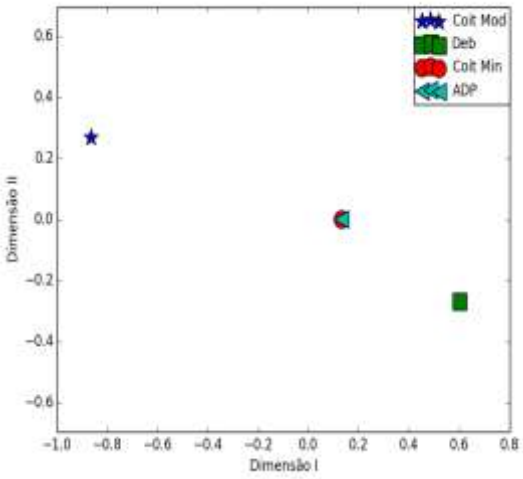

Figura 13: MDS – P3

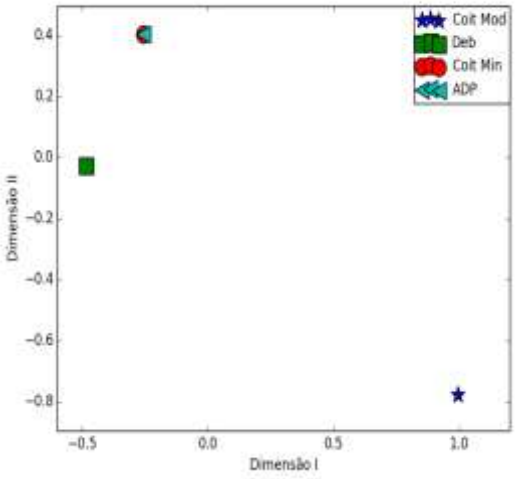

Figura 14: MDS – P4

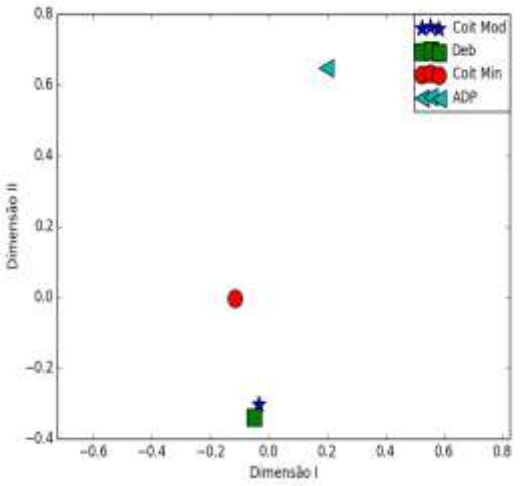

Figura 15: MDS – P5

4. CONCLUSÕES

Com base nos resultados obtidos, conclui-se neste trabalho que a combinação ABC e Método de Penalização Adaptativo é uma alternativa interessante para a resolução de problemas de otimização com restrições. Deve-se atentar para a escolha de um bom método de penalização.

Quanto aos métodos de penalização estudados, considera-se aplicação do método Coit Mod, proposto neste trabalho como a melhor escolha. O método Deb também é uma boa alternativa, visto que identifica corretamente factibilidade de soluções, porém este método contribuiu menos com o ABC na resolução dos problemas tratados. Quanto aos métodos Coit Min e ADP, não recomenda-se aplicação destes junto ao ABC, devido a possíveis incorreções na identificação da condição de factibilidade de soluções, causando resultados incorretos. O método de penalização Coit Mod não convergiu para soluções factíveis em algum momento da execução do ABC em apenas um dos problemas tratados.

Métodos como Coit Min e ADP que podem identificar factibilidade de soluções incorretamente, podem deixar de resultar em soluções realmente factíveis avaliadas, ainda que a função objetivo não seja tão atraente quanto outra de uma solução que na prática é infactível.

De maneira geral, conclui-se que combinar ao ABC métodos de penalização adaptativos que ao ajustar parâmetros de penalização realizam operação de subtração entre valores de função objetivo de soluções candidatas não é uma boa escolha. Durante a convergência do algoritmo, abelhas artificiais distintas podem estar memorizando a mesma fonte de alimento e tal operação pode fazer com que o parâmetro tenha valor 0, fazendo com que soluções infactíveis não sejam penalizadas.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] BAZARAA, M. S.; SHERALI, H. D.; SHETTY, C. M. **Nonlinear Programming - Theory and Algorithms**. 3.Ed. USA. John Wiley & Sons Inc, 2006.

[2] BRAJEVIC, I.; TUBA, M.; SUBOTIC, M. Performance of the improved artificial bee colony algorithm on standard engineering constrained problems. **International Journal of Mathematics and Computers in Simulation**, v. 5, n. 2, p. 135-143, 2011.

[3] COIT, D. W.; SMITH, A. E.; TATE, D. M. Adaptive penalty methods for genetic optimization of constrained combinatorial problems. **INFORMS Journal on Computing**, v. 8, n. 2, p. 173-182, 1996.

[4] DEB, K. An efficient constraint handling method for genetic algorithms. **Computer Methods in Applied Mechanics and Engineering**. v. 186, n. 2-4, p. 311-338, 2000.

[5] FREE, J. B. **A organização social das abelhas (Apis)**, São Paulo, Brasil. EPU, Ed. Univesidade de São Paulo, 1980.

[6] KARABOGA, D. **An idea based on honey bee swarm for numerical optimization**. Kayseri, Turkiye. Technical Report TR06, Erciyes University, Engineering Faculty, 2005.

[7] KARABOGA, D.; BASTURK, B. Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems. In MELIN, P., CASTILLO, O., AGUILAR, L., KACPRZYK, J., PEDRYCZ, W. (Eds), **Foundations of fuzzy logic and soft computing**, v. 4529 of Lecture Notes in Computer Science. Berlin Heidelberg: Springer, 2007a. p. 789-798.

[8] KARABOGA, D.; BASTURK, B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. **Journal of Global Optimization**, v. 39, n. 3, p. 459-471, 2007b.

[9] KARABOGA, D.; AKAY, B. A comparative study of artificial bee colony algorithm. **Applied Mathematics and Computation**. v. 214, n. 1, p. 108-132, 2009.

[10] KARABOGA, D.; AKAY, B. A modified artificial bee colony (ABC) algorithm for constrained optimization problems. **Applied Soft Computing**, v. 11, n. 3, p. 3021-3031, 2011.

[11] LEMONGE, A. C. C.; BARBOSA, H. J. C. An adaptive penalty scheme for genetic algorithms in structural optimization. **International Journal for Numerical Methods in Engineering**, v. 59, p. 703-736, 2004.

[12] LUENBERGER, D. G.; YE, Y. **Linear and nonlinear programming**, 3.Ed. New York, USA. Springer Science+Business Media, 2008.

[13] MONTEMURRO, M.; VINCENTI, A.; VANNUCCI, P. The automatic dynamic penalization method (ADP) for handling constraints with genetic algorithms. **Computer Methods in Applied Mechanics and Engineering**, v. 256, n. 0, p. 70-87, 2013.

[14] NANAKORN, P.; MEESOMKLIN, K. An adaptive penalty function in genetic algorithms for structural design optimization. **Computers & Structures**, v. 79, n. 29-30, p. 2527-2539, 2001.

[15] NEVO, S.; NEVO, D.; EIN-DOR, P.; Classifying information technologies: A multidimensional scaling approach. **Communications of the Association for Information Systems**, v. 27, n. 45, p. 831-842, 2010.

[16] RAO, S. S. **Engineering optimization: theory and practice**, 4.Ed. Hoboken, USA. John Wiley & Sons, Inc, 2009.

[17] SONMEZ, M. Discrete optimum design of truss structures using artificial bee colony algorithm. **Structural and Multidisciplinary Optimization**, v. 43, n. 1, p. 85-97, 2011.

[18] YANG, X. S.; GANDOMI, A. H. Bat algorithm: a novel approach for global engineering optimization. **International Journal for Computer-Aided Engineering and Software**, v. 29, n. 5, p. 464-483, 2012.

AGRADECIMENTOS

Os autores agradecem às agências de fomento pelos seguintes apoios CNPq (306815/2011-7) e FAPEMIG (TEC PPM 528/11, TEC APQ 03900/10 e TEC 388/14).