

ABORDAGEM DE ENXAME DE PARTÍCULAS INSPIRADA EM ALGORITMO CULTURAL APLICADA EM UM PROBLEMA DE OTIMIZAÇÃO NÃO-LINEAR COM RESTRIÇÕES

LEANDRO DOS SANTOS COELHO E LUCAS WEIHMANN

*Laboratório de Automação e Sistemas, Grupo Produtônica
Pós-Graduação em Engenharia de Produção e Sistemas, PUCPR / PPGEPS
Imaculada Conceição, 1155, CEP 80215-901 Curitiba, PR
E-mails: leandro.coelho@pucpr.br; lucas.weihmann@pucpr.br*

Abstract Optimization problems are widely encountered in various fields of mechanical engineering. Sometimes such problems can be very complex due to the nature of the objective function or the model constraints. In this context, the Particle Swarm Optimization (PSO) is a useful bio-inspired optimization method based on the metaphor of social behavior of birds flocking and fish schooling in search for food. Inspired by the potentialities of PSO, this work presents a comparative study of classical PSO algorithm and a new cultural PSO approach based on normative knowledge applied to a pressure vessel design, a constrained and nonlinear optimization problem.

Keywords Particle swarm optimization, nonlinear optimization, pressure vessel design, cultural algorithms.

Resumo Os problemas de otimização são vastamente encontrados em várias áreas de engenharia mecânica. Algumas vezes, tais problemas podem ser muitos complexos devido à natureza da função objetivo ou restrições do modelo. Neste contexto, a otimização por enxame (nuvem) de partículas (*Particle Swarm Optimization*, PSO) é um método de otimização bio-inspirado que pode ser útil. O PSO é baseado na metáfora do comportamento social de bandos de pássaros e cardumes de peixes na busca por alimentos. Inspirado pelas potencialidades do PSO, este trabalho apresenta um estudo comparativo da abordagem clássica de PSO e uma nova abordagem de PSO cultural baseada em conhecimento normativo aplicados ao projeto de um vaso de pressão, este um problema de otimização não-linear restrita.

Palavras-chave Enxame de partículas, otimização não-linear, projeto de vaso de pressão, algoritmo cultural.

1 Introdução

Os métodos de otimização têm duas formas de configuração: os métodos determinísticos e os métodos estocásticos. Os métodos determinísticos tendem a buscar um ponto de mínimo (quando o problema é de minimização) no espaço de busca baseadas na informação dada pelo gradiente da função objetivo (função custo). A eficiência destas técnicas depende de diversos fatores, tais como: o ponto inicial, a precisão da avaliação da direção descendente e o método utilizado para executar a busca em linha, além do critério de parada. A solução obtida é geralmente um ponto de mínimo local, que pode ser mínimo global se a função apresentar apenas uma moda. As duas desvantagens principais são: (i) a necessidade de avaliações do gradiente da função objetivo e (ii) a falta da garantia quanto à obtenção do mínimo global (Vasconcelos *et al.*, 1999).

Os métodos estocásticos, em contra partida, não necessitam de avaliações do gradiente da função objetivo, mas, no entanto, necessitam de muitas avaliações da função objetivo para obtenção de resultados aceitáveis. Entre os representantes dos métodos estocásticos estão os métodos *simulated annealing* e algoritmos evolutivos, onde o algoritmo genético é a abordagem mais difundida na literatura (Goldberg, 1989).

Neste trabalho é apresentado um estudo comparativo da abordagem do algoritmo evolutivo

denominado enxame (nuvem) de partículas (*Particle Swarm Optimization*, PSO) e uma nova abordagem cultural de PSO (CPSO) usando conhecimento normativo (Coelho e Mariani, 2006), um método de otimização estocástico, para resolução de um problema de otimização (*benchmark*) de um vaso de pressão cilíndrico, este um estudo de caso não-linear com restrições (Cao e Wu, 1997), com o propósito de minimizar o custo total de material e modelagem no projeto do vaso de pressão.

O algoritmo PSO foi desenvolvido inicialmente por Kennedy e Eberhart (1995) baseada nos estudos do sócio-biologista Edward Osborne Wilson. O algoritmo PSO constitui uma técnica evolutiva baseada em uma população de soluções e transições aleatórias. O algoritmo PSO apresenta características similares a técnicas da computação evolutiva, que são baseadas em uma população de soluções. Entretanto, no algoritmo PSO é motivada pela simulação de comportamento social e cooperação entre agentes em vez da sobrevivência do indivíduo mais apto. No algoritmo PSO, cada solução candidata (denominada partícula) possui associada uma velocidade. A velocidade é ajustada através de uma equação de atualização que considera a experiência da partícula correspondente e a experiência das outras partículas da população.

Além disso, o algoritmo PSO pode fazer uso de um algoritmo cultural para melhorar seu desempenho na otimização. O algoritmo cultural é uma classe de modelo computacional derivado a partir da observação do processo evolutivo cultural na

natureza humana. É um sistema de herança dupla que caracteriza a evolução da cultura humana tanto ao nível macro-evolutivo, que toma lugar no *espaço de crenças*, e ao nível micro-evolutivo, que ocorre no espaço populacional (Saleem e Reynolds, 2000). A principal idéia por trás dos algoritmos culturais é adquirir conhecimento sobre a solução do problema (crença) a partir da população em evolução e aplicar este conhecimento para guiar a busca (Reynolds e Chung, 1996). O conhecimento produzido no espaço populacional, dentro do nível micro-evolutivo, é seletivamente aceito ou passado ao espaço de crenças e usado para ajustar as estruturas simbólicas lá existentes. Este conhecimento pode então ser usado para influenciar as mudanças feitas na próxima geração da população (Almeida, 2005).

O restante do artigo é organizado da seguinte forma. Na seção 2 é apresentada uma descrição do problema de otimização do projeto do vaso de pressão e o tratamento de restrições. Os algoritmos PSO clássico e PSO cultural são apresentados na seção 3. Na seção 4 são discutidos os resultados de otimização pelo PSO e CPSO, e é apresentada uma comparação com os resultados apresentados na literatura recente. Na seção 5, a conclusão e perspectivas de futuros trabalhos são apresentadas.

2 Estudo de Caso de Otimização com Restrições

Uma das partes complexas encontradas na prática de otimização de projetos de engenharia envolve a manipulação de restrições. As limitações do mundo real freqüentemente introduzem múltiplas restrições não-lineares e não-triviais. As restrições geralmente limitam as soluções possíveis à um subconjunto do espaço de trabalho. Em geral, um problema de otimização restrita pode ser definido da seguinte forma:

$$\text{Minimização } f(X), X = \{x_1, x_2, \dots, x_n\} \in \mathfrak{R} \quad (1)$$

sujeita à $g_i(X) \leq 0, i = 1, 2, \dots, p$ e / ou

$$h_i(X) = 0, i = 1, 2, \dots, m$$

tal que $x_i^{(L)} \leq x_i \leq x_i^{(U)}, i = 1, 2, \dots, n$.

onde $x_i, i=1, 2, \dots, n$ são as variáveis de projeto de um problema de dimensão n ; $x_i^{(L)}$ e $x_i^{(U)}$ são os limites inferior (*Lower bound*) e superior (*Upper bound*) permitidos para cada variável i de projeto; $f(X)$ é a função objetivo a ser minimizada, $g_i(X)$ e $h_i(X)$ são, respectivamente, as p e m restrições de desigualdade e igualdade em um problema de otimização restrita.

No estudo de caso abordado neste trabalho de projeto de um vaso de pressão, o objetivo da implementação da otimização é minimizar o custo total de material, modelagem e soldagem de um vaso de pressão cilíndrico. Neste caso existem quatro

variáveis que são consideradas: a espessura do corpo (T_s), a espessura da cabeça (T_h), o raio interno (R) e o comprimento da seção cilíndrica (L).

Os valores das variáveis T_s e T_h são inteiros múltiplos de 0,0625 polegadas. Neste caso é a espessura disponível na indústria para chapas de aço prensado. Entretanto, os valores de R e L são contínuos. Resumindo, o problema de otimização é dado pelas seguintes equações:

$$\text{Minimizar } f(X) = 0,6224x_1x_3x_4 + 1,7781x_2x_3^2 + 3,1661x_1^2x_4 + 19,84x_1^2x_3 \quad (2)$$

sujeito às desigualdades

$$g_1(X) = -x_1 + 0,0193x_3 \leq 0$$

$$g_2(X) = -x_2 + 0,00954x_3 \leq 0$$

$$g_3(X) = -px_3^2x_4 - \frac{4}{3}px_3^3 + 1296000 \leq 0$$

$$g_4(X) = x_4 - 240 \leq 0$$

onde, x_1, x_2, x_3 e x_4 representam T_s, T_h, R e L , respectivamente, conforme representado na figura 1.

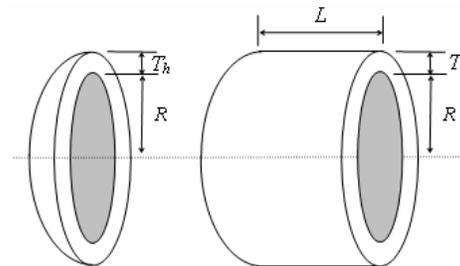


Figura 1. Diagrama do vaso de pressão.

Os seguintes limites para as variáveis de projeto foram considerados (Coello Coello, 2000): $1 \leq x_1 \leq 99$, $1 \leq x_2 \leq 99$, $10,0 \leq x_3 \leq 200,0$ e $10,0 \leq x_4 \leq 200,0$.

Note que os valores para x_1 e x_2 são considerados como múltiplos inteiros de 0,0625 polegadas. Assim, ao lidar com inteiros usando o algoritmo PSO adota-se que os valores de x_1 e x_2 são truncados para os valores inteiros mais próximos, isto quando da avaliação da função objetivo dada pela equação (2).

2.1 Tratamento de restrições

Devida à complexidade e imprevisibilidade das restrições, é difícil encontrar uma solução próxima da ótima. Neste contexto, recentemente, alguns algoritmos evolutivos têm sido propostos para otimização de problemas de engenharia. Diferentes tipos de métodos foram propostos para manipulação destas restrições na literatura (Michalewicz e Schonauer, 1996) são aspectos relevantes para eficiência do procedimento de otimização.

Com o propósito de tornar viável a implementação computacional de um algoritmo de otimização para este sistema de vaso de pressão, utilizaram-se algumas regras que restringem o espaço de busca dos

possíveis valores para cada variável e funções de penalidade permitindo lidar com soluções inactíveis.

Neste trabalho, a metodologia de tratamento de restrições é dividida em duas etapas. A primeira etapa visa à obtenção de soluções para as variáveis de decisão dentro dos limites inferior (lim_{inf}) e superior (lim_{sup}) impostos pelo projetista, isto é, $x \in [x^{(L)}, x^{(U)}]$. Neste caso, quando uma restrição de limite inferior não é satisfeita aplica-se a regra de “reparo”,

$$x_i = x_i + w \cdot r \cdot \{x^{(U)} - x^{(L)}\} \quad (3)$$

De forma análoga, no caso de que uma restrição de limite superior não seja satisfeita utiliza-se a seguinte regra de reparo,

$$x_i = x_i - w \cdot r \cdot \{x^{(U)} - x^{(L)}\} \quad (4)$$

onde $w \in [0,1]$ é um parâmetro de projeto e r é um número aleatório gerado com distribuição uniforme no intervalo $[0,1]$. Na segunda etapa as variáveis de decisão são consideradas nas desigualdades, g ; neste artigo adota-se um procedimento para desigualdades do tipo $g_i(X) \leq 0$, onde deseja-se minimizar uma função $f(X)$. Neste caso adota-se o seguinte cálculo para a equação de *fitness*,

$$aval(X) = \begin{cases} f(X), & \text{para } g_i(x) \leq 0 \\ f(X) + na \cdot q \cdot \sum_{i=1}^n g_i(x), & \text{para } g_i(x) > 0 \end{cases} \quad (5)$$

onde na é o número de restrições não atendidas, q é uma constante de penalidade (adotou-se $q=1$). A penalidade é nula se nenhuma violação de restrições ocorre e possui valor positivo em outros casos, ou seja, se $\forall g_i(x) \leq 0, i=1, \dots, p$. Usualmente, a função de penalidade é baseada na distância de uma solução da região factível ou no esforço de “reparar” um conjunto de soluções (ou seja, forçá-lo a ficar em uma região factível).

3 Fundamentos do Algoritmo PSO

3.1 Algoritmo PSO clássico

O algoritmo PSO é uma abordagem da inteligência coletiva baseada em população de soluções. De forma similar aos algoritmos evolutivos, o PSO inicia com uma população de soluções gerada aleatoriamente com distribuição uniforme.

De forma diferente dos algoritmos evolutivos (Goldberg, 1989), cada solução potencial (indivíduo) no PSO é também atribuída uma velocidade aleatória. As soluções potenciais denominadas *partículas* são então “movimentadas” pelo espaço de busca do problema. Cada partícula conserva o conhecimento do seu melhor valor da função de aptidão (*fitness*) denotada por *pbest* (versão *local*). Um outro melhor valor é “seguido” pela versão *global*, *gbest*, da PSO e sua localização obtida de alguma partícula que compõe a população.

O conceito do PSO consiste de, a cada passo iterativo, mudar a velocidade de cada partícula em direção as localizações do *pbest* e do *gbest*. A rapidez do procedimento de busca é ponderada através de um termo gerado de forma aleatória, sendo vinculado este, de forma separada, as localizações do *pbest* e do *gbest*. O procedimento para implementação da PSO é regido pelas seguintes etapas:

(i) iniciar uma população (matriz) de TP partículas, com posições e velocidades em um espaço de problema n dimensional, aleatoriamente com distribuição uniforme.

(ii) para cada partícula, avaliar a função de aptidão (função objetivo);

(iii) comparar a avaliação da função de aptidão da partícula com o *pbest* da partícula. Se o valor corrente é melhor que *pbest*, então o valor de *pbest* passa a ser igual ao valor da função de aptidão da partícula, e a localização do *pbest* passa a ser igual a localização atual no espaço n dimensional;

(iv) comparar a avaliação da função de aptidão com o prévio melhor valor de aptidão da população. Se o valor atual é melhor que o *gbest*, atualizar o valor de *gbest* para o índice e valor da partícula atual;

(v) modificar a velocidade e a posição da partícula de acordo com as equações (6) e (7), respectivamente:

$$v_i(t+1) = w \cdot v_i(t) + c_1 \cdot ud \cdot [p_i(t) - x_i(t)] + c_2 \cdot Ud \cdot [p_g(t) - x_i(t)] \quad (6)$$

$$x_i(t+1) = x_i(t) + \Delta t \cdot v_i(t+1) \quad (7)$$

onde Δt é igual a 1, t é a geração atual e $i=1, \dots, TP$, tal que o tamanho de população é TP .

(vi) ir para a etapa (ii) até que um critério de parada seja encontrado, usualmente uma função de aptidão suficientemente boa ou um número máximo de iterações (gerações).

As notações usadas são: $x_i = [x_{i1}, x_{i2}, \dots, x_{in}]^T$ armazena a posição da i -ésima partícula, $v_i = [v_{i1}, v_{i2}, \dots, v_{in}]^T$ armazena a velocidade da i -ésima partícula e $p_i = [p_{i1}, p_{i2}, \dots, p_{in}]^T$ representa a posição do melhor valor de aptidão da i -ésima partícula. O índice g representa o índice da melhor partícula entre todas as partículas do grupo. A variável w é a ponderação de inércia, c_1 e c_2 são constantes positivas; ud e Ud são duas funções para geração de números aleatórios com distribuição uniforme no intervalo $[0,1]$, respectivamente.

As velocidades das partículas em cada dimensão são limitadas a um valor máximo de velocidade, V_{max} . O V_{max} é importante, pois determina a resolução na qual a região próxima a solução atual é procurada. Se V_{max} é alto, o algoritmo PSO facilita a busca global,

enquanto um valor V_{max} pequeno enfatiza as buscas locais.

A primeira parte na equação (6) é um termo de momento da partícula. A ponderação de inércia w representa o grau de momento da partícula. A segunda parte consiste da parte “cognitiva”, que representa o “conhecimento” independente da partícula. A terceira parte é a “social”, que representa a colaboração entre as partículas.

As constantes c_1 e c_2 representam ponderações às partes cognitivas e social, respectivamente, que influenciam cada partícula em direção ao $pbest$ e $gbest$. Estes parâmetros são usualmente ajustados por heurísticas de tentativa e erro. O tamanho da população também é selecionado dependendo do problema. Neste trabalho, uma modificação variante durante as gerações t de c_1 e c_2 foi utilizada, que pode ser representada por (Ratnaweera *et al.*, 2004):

$$c_1(t) = (c_{1f} - c_{1i}) \cdot \frac{t}{t_{max}} + c_{1i} \quad (8)$$

$$c_2(t) = (c_{2f} - c_{2i}) \cdot \frac{t}{t_{max}} + c_{2i} \quad (9)$$

onde c_{1i} , c_{1f} , c_{2i} e c_{2f} são constantes.

A abordagem de algoritmo PSO proposta, neste artigo, é baseada em conceitos de conhecimento normativo de algoritmos culturais. Esta abordagem é descrita na sub-seção a seguir.

3.2 Algoritmo PSO usando conhecimento normativo

A incorporação de mecanismos baseados em conhecimento pode melhorar consideravelmente o desempenho de um algoritmo evolutivo quando o conhecimento adquirido durante o processo evolutivo de solução do problema é usado para guiar o processo de solução do problema. Os padrões de identificados ou conhecimento adquirido durante a busca da solução, pode ser usado para guiar a geração de soluções candidatas: promover mais instâncias de candidatos desejados, ou reduzir o número de candidatos indesejados na população, de forma a guiar apropriadamente a otimização (Almeida, 2005). Saleem e Reynolds (2000) identificam cinco categorias básicas de conhecimento que podem ser úteis na tomada de decisão. As categorias são as seguintes: conhecimento normativo (intervalos de comportamentos aceitáveis), conhecimento situacional (exemplares de soluções de sucesso ou insucesso), conhecimento de domínio (conhecimento do domínio dos objetos, seus relacionamentos, e interações), conhecimento histórico (padrões temporais de comportamento) e conhecimento topográfico (padrões espaciais de comportamento). Este conjunto de categorias é visto como completo para um dado domínio no sentido de que todo o conhecimento disponível pode ser expresso em termos de uma destas classificações.

Um modelo de algoritmo cultural possui dois níveis de evolução: (i) nível de população e (ii) nível de espaço de crenças. Ou seja, adicionalmente ao

espaço de população, um algoritmo cultural tem um espaço de crenças em que as crenças (conhecimento sobre a resolução do problema) adquiridas durante a evolução da população pode ser armazenadas e integradas. Neste caso, uma *função de aceitação* é utilizada para gerar as crenças baseadas na experiências das partículas (no caso do PSO) para o espaço de população (Coelho e Mariani, 2006). Uma representação do PSO baseado em algoritmo cultural (C-PSO) é apresentada na figura 2.

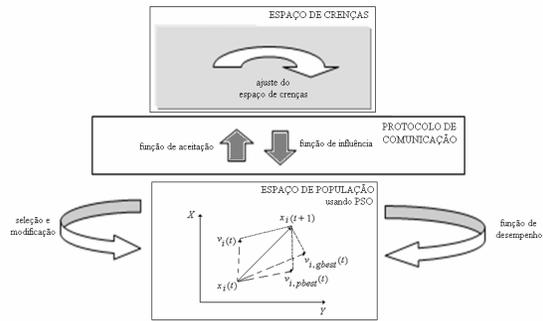


Figura 2. Representação do algoritmo PSO no espaço de população de um algoritmo cultural.

Neste trabalho são empregados conceitos de conhecimento normativo no algoritmo PSO. O conhecimento normativo é um conjunto de intervalos de variáveis considerados promissores, que provêm padrões para comportamentos dos indivíduos e diretivas para os ajustes das partículas. O conhecimento normativo instrui as partículas para se mover no intervalo certo se eles ainda não estiverem lá. Para cada componente do conhecimento normativo, existe um limite superior e inferior, e o valor de desempenho para as partículas dos limites superiores e inferiores.

Neste trabalho, o número de partículas aceita, $h_{aceitas}$, para atualização do espaço de crenças é dado por

$$h_{aceitas} = p \cdot TP + \frac{P}{t_c} TP \quad (10)$$

onde p é um parâmetro definido pelo usuário (neste trabalho adotou-se $p=0,4$) e t_c é um contador de gerações que é reiniciado para 1 quando a melhor solução ($gbest$) da população é melhorado. Na figura 3 é apresentado um pseudo-código do algoritmo CPSO proposto neste artigo. No CPSO proposto só é modificada a etapa (v) do PSO clássico, este descrito na seção 3.1.

Neste trabalho, adotou-se também um escalonamento variante no tempo da variável de velocidade máxima V_{max} , tal que:

$$V_{max}(t) = (V_{final} - V_{inicial}) \cdot \frac{t}{t_{max}} + V_{inicial} \quad (11)$$

onde V_{final} e $V_{inicial}$ representam, respectivamente, os valores final e inicial da velocidade. Neste trabalho adotou-se $V_{final} = 0,10$ e $V_{inicial} = 0,30$, ou

seja, 10% e 30% dos limites de busca de cada partícula.

4 Análise dos Resultados de Otimização

Em termos de configuração dos parâmetros de controle, tanto para o PSO clássico quanto para o CPSO foi adotado: tamanho de população de 20 partículas e critério de parada de 50 gerações, totalizando um total de 1000 avaliações da função objetivo f por experimento.

Adota-se, neste artigo, algoritmo PSO clássico e CPSO com fator de inércia com decaimento linear, iniciando com valor 0,9 e terminado com 0,4. O V_{max} no PSO clássico é configurado para 20% do intervalo da variável de cada dimensão.

Foram realizados 50 experimentos com cada abordagem de algoritmo PSO. Uma análise estatística dos resultados dos experimentos é apresentada na tabela 1. Um detalhe é que para as abordagens PSO(2) e CPSO(2) apresentadas na tabela 1, adotou-se as equações (8) e (9) com $[c_{1i}; c_{1j}] = [2,05; 0,40]$ e $[c_{2i}; c_{2j}] = [0,40; 2,05]$.

Nota-se pelos resultados de simulação apresentados na tabela 1, que as abordagens de CPSO testadas obtiveram desempenho superior em termos de média, mínimo e mediana da função objetivo, $f(X)$, aos obtidos pelas abordagens de PSO clássico testadas. O melhor resultado foi obtido usando CPSO(1) com $f(X) = 6145,2816$. Entretanto, o melhor desempenho, em termos da média de $f(X)$, após 50 gerações foi obtido pelo CPSO(2).

O melhor resultado obtido pelo CPSO(1), quando comparado a outras abordagens de literatura, só é inferior ao obtido por Mezura-Montes *et al.* (2006). Entretanto, neste artigo a CPSO(1) avaliou a $f(X)$ somente 1000 vezes, enquanto na abordagem de Mezura-Montes *et al.* (2006) foram realizados experimentos com 24000 avaliações da $f(X)$.

5 Conclusão e Futura Pesquisa

Neste artigo é proposta a implementação de um algoritmo de otimização combinando as teorias de PSO e conhecimento normativo dos algoritmos culturais.

Um estudo de caso de otimização não-linear com restrições foi apresentado neste artigo. Os resultados obtidos mostraram que a metodologia proposta apresentou desempenho apropriado e eficiência quanto à precisão, flexibilidade e convergência na otimização de problema *benchmark* de um vaso de pressão. Notou-se pelos resultados obtidos (ver tabela 1) que as abordagens CPSO(1) (melhor f mínimo) e CPSO(2) (melhor f médio) validadas nos testes de simulação são competitivas frente a outras da literatura (ver tabela 2).

A futura pesquisa vinculada a este artigo objetivará um estudo comparativo de abordagens de CPSO em outros problemas *benchmark* da literatura.

<p>Notação</p> <p>% Índice g representa o índice da melhor partícula da população.</p> <p>% j é a dimensão do problema, tal que $j=1, \dots, n$</p> <p>% Gd é um número gerado aleatoriamente com distribuição normal com média zero e variância unitária.</p> <p>% l_j e u_j são os limites inferiores e superiores do espaço de crenças para a j-ésima dimensão</p> <p>Pseudo-código da etapa (v) do CPSO</p> <p>Calcula as $h_{aceitas}$ partículas da população TP</p> <p>Escolhe as $h_{aceitas}$ de $a = \{1 \vee 2 \vee \dots \vee TP\}$</p> <p>Calcula l_j e u_j das $h_{aceitas}$ partículas</p> <p>Se $x_{ij}(t) < l_j$ e $x_{ij}(t) < p_{gj}(t)$ então</p> $v_{ij}(t+1) = w \cdot v_{ij}(t) + c_1 \cdot ud \cdot [p_{ij}(t) - x_{ij}(t)] + c_2 \cdot Gd \cdot [p_{gj}(t) - x_{ij}(t)]$ <p>Senão</p> <p>Se $x_{ij}(t) > u_j$ e $x_{ij}(t) > p_{gj}(t)$ então</p> $v_{ij}(t+1) = w \cdot v_{ij}(t) - c_1 \cdot ud \cdot [p_{ij}(t) - x_{ij}(t)] - c_2 \cdot Gd \cdot [p_{gj}(t) - x_{ij}(t)]$ <p>Senão</p> <p>Se $x_{ij}(t) < u_j$ e $x_{ij}(t) > p_{gj}(t)$ então</p> $v_{ij}(t+1) = w \cdot v_{ij}(t) + c_1 \cdot ud \cdot [p_{ij}(t) - x_{ij}(t)] - c_2 \cdot Gd \cdot [p_{gj}(t) - x_{ij}(t)]$ <p>Senão</p> <p>Se $x_{ij}(t) > u_j$ e $x_{ij}(t) < p_{gj}(t)$ então</p> $v_{ij}(t+1) = w \cdot v_{ij}(t) - c_1 \cdot ud \cdot [p_{ij}(t) - x_{ij}(t)] + c_2 \cdot Gd \cdot [p_{gj}(t) - x_{ij}(t)]$ <p>Fim Se</p>
--

Figura 3. Pseudo-código do CPSO.

Referências Bibliográficas

- Almeida, A. D. V. (2005). B-spline:CAEP – Algoritmos culturais para a geração de trajetórias b-spline de robôs móveis, Dissertação de mestrado, PPGEPS, Pontifícia Universidade Católica do Paraná, Curitiba, PR.
- Cao, Y. J. e Wu, Q. H. (1997). Mechanical design optimization, *IEEE Congress on Evolutionary Computation*, Indianapolis, USA, pp. 443-446.
- Coelho, L. S. e Mariani, V. C. (2006). An efficient particle swarm optimization approach based on cultural algorithm applied to mechanical design, *IEEE Congress on Evolutionary Computation*, Vancouver, Canada, pp. 3844-3848.

- Coello, C. A. C. (2000). Use of self-adaptive penalty approach for engineering optimization problems, *Computers in Industry*, **41**(2): 113-127.
- Deb, K. (1997). GeneAS: a robust optimal design technique for mechanical component design, Dasrputa, D. e Michalewicz (eds.), *Evolutionary algorithms in Engineering Applications*, Berlin, Springer-Verlag, pp. 497-514.
- Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. Reading: Addison-Wesley, USA.
- Kennedy, J. e Eberhart, R. C. (1995). Particle swarm optimization, *IEEE Int. Conference on Neural Networks*, Perth, Australia, pp. 1942-1948.
- Mezura-Montes, E., Coello, C. A. C. e Velásquez-Reyes, J. (2006). Increasing successful offspring and diversity in differential evolution for engineering design, *Proceedings of the 7th International Conference on Adaptive Computing in Design and Manufacture (ACDM'2006)*, The Institute for People-centred Computation (IP-CC), Bristol, UK, pp. 1310-139.
- Michalewicz, Z. e Schoenauer, M. (1996). Evolutionary algorithms for constrained parameter optimization problems, *Evolutionary Computation*, **4**(1): 1-32.
- Ratnaweera, A., Halgamuge, S. K. e Watson, H. C. (2004). Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients, *IEEE Transactions on Evolutionary Computation*, **8**(3): 240-255.
- Reynolds, R. G. e Chung, C. (1996). A self-adaptive approach to representation shifts in cultural algorithms, *IEEE International Conference on Evolutionary Computation*, Nagoya, Japan.
- Saleem, S. e Reynolds, R. G. (2000). The impact of environmental dynamics on cultural emergence, *Festschrift, in Honor of John Holland*, Oxford University Press.
- Sandgren, E. (1990). Nonlinear integer and discrete programming in mechanical design optimization, *Journal of Mechanical Design*, **112**: 223-229.
- Vasconcelos, J. A., Saldanha, R. R., Krähenbühl, L. e Nicolas, A. (1999). Genetic algorithm coupled with a deterministic method for optimization in electromagnetics, *IEEE Transactions on Magnetics*, **33**(3): 1860-1863.
- Zhang, C. e Wang, H. P. (1993). Mixed-discrete nonlinear optimization with simulated annealing, *Engineering Optimization*, **17**(3): 263-280.

Tabela 1. Resultados de simulação para a função objetivo f (50 experimentos) dos algoritmos PSO e CPSO.

método	c_1	c_2	Máximo	Mínimo	Média	Mediana	Desvio padrão
PSO(1)	2,05	2,05	30083,2684	11152,2320	18334,2895	17570,2579	3814,2064
PSO(2)	eq. (8)	eq. (9)	44952,5739	8329,4909	19132,7897	18877,9877	6672,1899
PSO(3)	0,50	0,50	39161,7822	9092,3680	21116,9249	20485,1172	7402,9950
PSO(4)	1,00	1,00	37206,3491	9635,4847	20327,7222	19073,1670	5601,0156
CPSO(1)	2,05	2,05	80614,2822	6145,2816	6986,4229	6952,2456	4063,6665
CPSO(2)	eq. (8)	eq. (9)	14885,9004	6157,5340	6788,8838	6577,1691	1247,6607
CPSO(3)	0,50	0,50	19086,6359	6213,1045	8897,8453	7193,6987	3369,8639
CPSO(4)	1,00	1,00	10491,8070	6206,3422	6840,0610	6732,9620	642,4890

Tabela 2. Comparação de resultados para diferentes algoritmos para otimização de um vaso de pressão.

referência	Sandgren (1990)	Zhang e Wang (1993)	Cao e Wu (1997)	Coello (1990)	Mezura-Montez et al. (2006)	Deb (1997)	Sistema proposto
método / variáveis	otimização mista	<i>simulated annealing</i>	programação evolutiva	algoritmo genético	evolução diferencial	algoritmo genético	PSO cultural
x_1	48,3807	58,2900	51,1958	40,3239	42,098446	48,3290	44,8872
x_2	11,7449	43,6930	90,7821	200,0000	176,636047	112,6790	144,9984
x_3	1,1250	1,1250	1,0000	0,8125	0,8125	0,9375	0,8750
x_4	0,6250	0,6250	0,6250	0,4375	0,4375	0,5000	0,4375
g_1	-0,1913	-0,0250	-0,0119	-0,0034324	0	-0,004750	-0,008677
g_2	-0,1634	-0,0689	-0,1366	-0,052847	-0,035881	-0,038941	-0,009276
g_3	-75,8750	6,5496	-13584,5631	-27,105845	-0,000002	-3652,876838	-663,1437
g_4	-128,2551	-196,3070	-149,2179	-40,0000	-63,363949	-127,321000	-95,0016
$f(X)$	8048,6190	7197,7	7108,6160	6288,7445	6059,70166	6370,7035	6145,2816