

Controle de inércia não monotônico na otimização por enxame de partículas

Tiago Silveira, Humberto César Brandão de Oliveira,
Luiz Eduardo da Silva, Ricardo Menezes Salgado

Departamento de Ciências Exatas – Universidade Federal de Alfenas
Rua Gabriel Monteiro da Silva, 700, Alfenas, 37130-000, MG, Brasil

{tiago, humberto, luizedu, ricardo}@bcc.unifal-mg.edu.br

Resumo

Este trabalho apresenta um mecanismo que possibilita a redução das chances do processo de otimização de funções não lineares estacionarem em mínimos locais, ao utilizarem a meta-heurística Otimização por Enxame de Partículas (PSO). Tal mecanismo é uma forma não monotônica de controlar a inércia da partícula, que é um dos fatores responsáveis pela movimentação desta durante o processo de otimização. Os resultados experimentais foram comparados com o modelo original da PSO padrão, a fim de mostrar o potencial para encontrar uma melhor solução em funções de benchmark, em problemas complexos. Ao final, uma comparação de ambos os modelos aplicados a um problema real para a regulação de pesos sinápticos de uma rede neural do tipo Multi-Layer Perceptron foi feita, e as PSOs, por se tratarem de técnicas de propósito geral, mostraram resultados interessantes.

PALAVRAS-CHAVE: inteligência de enxame, Otimização por Enxame de Partículas, vida artificial, rede neural artificial MLP

Abstract

Particle Swarm Optimization with inertia non-monotonic control. This work presents a mechanism to reduce the chances of the optimization process of nonlinear functions stagnating in local minima, using the meta-heuristic “Particle Swarm Optimization”. This mechanism is a non-monotonic way to control the particle inertia, which is one of the factors responsible for movement during the optimization process. The experimental results were compared to the original PSO model aiming to show the potential to find a better solution related to the benchmark functions for complex problems. Finally, a comparison of both models was made in the adjustment of synaptic weights of a Multi-Layer Perceptron neural network obtaining interesting results.

KEY WORDS: swarm intelligence, Particle Swarm Optimization, artificial life, artificial neural network MLP

1 Introdução

A otimização numérica é a tarefa de determinar valores ótimos dentro de um universo de possibilidades ($x \in \mathfrak{R}^n$), no qual cada uma dessas possibilidades é avaliada por uma função ($f : \mathfrak{R}^n \rightarrow \mathfrak{R}$), chamada função de avaliação, que pode ser linear ou não linear. Nesse contexto, o grau de complexidade dos problemas de otimização pode ser alto, caso haja restrições a serem satisfeitas (Coello *et al.*, 2004).

As técnicas mais utilizadas para tratar problemas de otimização numérica podem ser divididas em duas classes:

os algoritmos específicos para problemas bem definidos, como, por exemplo, as técnicas de programação linear, e os meta-algoritmos de busca, como a própria otimização por enxame de partículas, que é o objeto de estudo deste trabalho.

O algoritmo de Otimização por Enxame de Partículas (PSO) é um meta-algoritmo de busca motivado no comportamento social. Mediante a competição e a cooperação entre indivíduos, assim como na natureza, a otimização com base no comportamento social pode trazer diversos benefícios e encontrar qualificadas soluções de forma eficiente, mantendo certa simplicidade no processo de otimização.

Apesar de constituírem abordagens diferentes, usualmente, os algoritmos com base na natureza trabalham de forma similar, com a atualização da população de acordo com informações obtidas sobre a adaptabilidade ao ambiente, implicando a busca de uma solução mais promissora (Shi e Eberhart, 1999). Porém, com a complexidade dos problemas não lineares, a otimização por tais algoritmos se torna deficiente para encontrar o melhor ponto do espaço, em vista do elevado número de soluções ótimas locais que o problema passa a ter. Um ponto ótimo local geralmente não é a solução desejada. Se, por influência de um mínimo local, a otimização converge para pontos distantes da solução ótima, o resultado desejado poderá não ser satisfatório. Logo, fazer com que a otimização desses problemas não convirja prematuramente para pontos que venham prejudicar o resultado vem sendo uma tarefa bastante explorada por muitos pesquisadores (Jiao *et al.*, 2008; Yang *et al.*, 2007; Lvbjerg *et al.*, 2001).

Com este intuito, neste trabalho é apresentado um mecanismo para reduzir as chances de o processo de otimização de funções não lineares permanecer estacionado em mínimos locais, ao utilizar a meta-heurística Otimização por Enxame de Partículas. Tal mecanismo está relacionado ao controle de forma não monotônica da inércia da partícula. Esta é um dos fatores de atualização da velocidade que cada partícula terá durante o processo de otimização. Para isso, em vez de se manter a inércia da partícula com o comportamento monotonicamente decrescente do modelo original da PSO, aplica-se uma modificação no termo de inércia, modulando-o pela função cosseno, gerando, então, um comportamento não monotônico deste termo. Assim, estando estabilizadas em um mínimo local, as partículas, por meio do controle da inércia não monotônico proposto, ganham energia, aumentando suas velocidades e, consequentemente, saem do mínimo local encontrado, podendo, assim, buscar soluções mais favoráveis e trazer certos benefícios ao desempenho do algoritmo.

No decorrer deste trabalho, mostra-se o potencial do mecanismo de controle de inércia da partícula em problemas de benchmark, comparando seus resultados experimentais com o modelo original da PSO padrão. Outro importante aspecto abordado é a aplicação das técnicas focalizadas neste estudo em um problema de reconhecimento de padrões. Nesse sentido, também foram realizados experimentos para proceder com o treinamento de redes neurais artificiais do tipo *Multi-Layer Perceptron* (MLP), no qual as PSOs tiveram a tarefa de ajustes dos pesos sinápticos, representando o conhecimento adquirido pela rede neural e usando como treinamento uma base de dados reais relacionados ao diagnóstico de câncer de mama, oriunda dos hospitais da Universidade de Wisconsin, EUA (Wolberg e Mangasarian, 1990) e obtidas, para o presente experimento, do repositório Proben1 (Prechelt,

1994). O restante do trabalho está dividido da seguinte forma: a Seção 2 apresenta uma visão geral do modelo da PSO padrão. Na Seção 3, são citados alguns trabalhos que buscam um melhor desempenho da PSO. A Seção 4 apresenta a forma como este trabalho trata a inércia da partícula na PSO. A Seção 5 descreve as configurações experimentais usadas para encontrar os resultados descritos e discutidos na Seção 6. A Seção 7 apresenta uma breve introdução ao problema em que ambas as PSOs foram aplicadas, mostrando, também, os resultados obtidos e as conclusões acerca dos experimentos. Finalmente, a Seção 8 faz uma conclusão do estudo, indicando também as futuras diretrizes do trabalho.

2 Otimização por enxame de partículas

O algoritmo de Otimização por Exame de Partículas (PSO, do inglês *Particle Swarm Optimization*) foi introduzido em meados da década de 1990 por Kennedy e Eberhart (1995), como uma alternativa ao Algoritmo Genético padrão. Conceitualmente, a PSO é uma técnica de busca estocástica que visa otimizar uma função de objetivo, que é desenvolvida por meio da tentativa de simulação gráfica da coreografia realizada por pássaros em busca de alimentos. Mais tarde, na busca de fundamentos teóricos, foram realizados estudos sobre a maneira como indivíduos interagem, de uma forma geral, em sociedades, trocando informações e revendo seus conceitos em busca de melhores soluções para seus problemas (Kennedy *et al.*, 2001).

A PSO tem raízes em 2 principais metodologias componentes. Talvez o mais evidente sejam seus laços com a vida artificial em geral. Ela é também relacionada, entretanto, à computação evolucionária e tem ligação com algoritmos genéticos e com a programação evolucionária (Kennedy e Eberhart, 1995). Entretanto, a PSO não utiliza os operadores evolucionários para manipular seus indivíduos, mas uma velocidade que é atribuída a cada indivíduo para a movimentação pelo espaço de busca, havendo um ajuste de velocidade a cada iteração, de acordo com a sua própria experiência (experiência cognitiva), a experiência das outras partículas (experiência social) do enxame e sua velocidade atual. A não execução do operador de seleção é uma característica que distingue a PSO dos algoritmos genéticos, da programação evolucionária e das estratégias evolucionárias (Eberhart e Shi, 1998; Angeline, 1998).

Na PSO, cada partícula é considerada uma possível solução para o problema de otimização. Aqui, todas as partículas são mantidas como membros permanentes da população, e o termo atualizado é a velocidade da partícula. Essa velocidade é responsável por fazer com que a partícula tente ir para uma região mais promissora, pois se trata de um vetor em permanente busca de uma melhor

solução momentaneamente. Por essa razão, a PSO é o único algoritmo evolucionário que não considera a sobrevivência do indivíduo mais forte (Eberhart e Shi, 1998).

A implementação do algoritmo da PSO é dada da seguinte forma: seja s o tamanho do enxame; n , a dimensão do problema; t , o instante atual, cada partícula i possui uma posição $x_i(t) \in \mathbb{R}^n$ no espaço de soluções e uma velocidade $v_i(t) \in \mathbb{R}^n$, que indica a direção e a magnitude de seu deslocamento. Adicionalmente, cada partícula possui a lembrança $p_i^* \in \mathbb{R}^n$ da melhor posição individual visitada, e o enxame possui a lembrança da melhor posição visitada por alguma partícula até então ($p_b^* \in \mathbb{R}^n$). No decorrer do algoritmo, a velocidade de cada partícula é calculada segundo a melhor posição visitada individual p_i^* , a melhor posição visitada do enxame p_b^* e a componente que agrupa sua velocidade anterior, servindo com um termo de momentum (inércia). Assim, a atualização da velocidade de cada partícula permanece de acordo com a equação (1):

$$v_i(t+1) = wv_i(t) + c_1r_1(p_i^*(t) - x_i(t)) + c_2r_2(p_b^*(t) - x_i(t)) \quad (1)$$

em que r_1 e r_2 são componentes aleatórias retiradas de uma distribuição uniforme entre 0 e 1, responsáveis por uma busca mais natural, como na natureza, durante o processo de otimização (Kennedy e Eberhart, 1995). Já c_1 e c_2 são os coeficientes de aceleração, que, geralmente, possuem valores fixos e iguais, responsáveis por controlar a distância do movimento de uma partícula em apenas uma iteração. O item w é o peso de inércia (termo de *momentum*) que multiplica a velocidade no instante t anterior e faz com que a busca seja mais explorativa no início e mais explorativa no final, para um valor inércia linearmente decrescente, como sugerido por Kennedy e Eberhart (1995).

Após a atualização da velocidade da partícula, sua posição atual sofre a atualização segundo a equação (2):

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (2)$$

Assim sendo, como apresentado em Kennedy e Eberhart (1995), nota-se que a PSO compreende um conceito extremamente simples, pois necessita apenas de operadores matemáticos primitivos para sua implementação. Isso é vantajoso, pois acaba gerando um processo computacionalmente barato, tanto em termos de memória quanto de velocidade.

3 Trabalhos relacionados

A partir da concepção original da PSO, têm sido pesquisadas novas adaptações no algoritmo a fim de melhorar seu desempenho. Tal tarefa não se mostra tão

simples de ser feita. Como exemplo, para fazer a seleção de parâmetros, no trabalho apresentado em Shi e Eberhart (1998), evidencia-se a complexidade da elaboração de tal processo na PSO.

Além das configurações dos parâmetros, vários pesquisadores têm analisado o desempenho da PSO com diferentes configurações, utilizando várias funções para o espaço de busca, muito conhecidas na literatura. Por exemplo, nos trabalhos de Suganthan (1999) e Kennedy (1999), são discutidas as configurações de vizinhança das partículas. Esta é uma característica que afeta o termo social das partículas durante o processo de otimização. O trabalho de Kennedy (1999) conclui que a vizinhança do tipo estrela (em que a partícula tem o conhecimento da melhor posição encontrada entre todas as partículas) é uma boa opção na busca de melhores soluções na PSO.

Assim como este trabalho, outros estudos também focalizam o peso de inércia da partícula para buscar um melhor desempenho da PSO. Em Yang *et al.* (2007), é proposta uma nova estratégia com uma modificação no algoritmo da PSO, utilizando um peso de inércia dinâmico, que se ajusta de acordo com a velocidade de evolução e o grau de agregação do enxame. Em Jiao *et al.* (2008), também é proposta uma PSO que usa um peso de inércia dinâmico, o qual decresce de acordo com a evolução do enxame.

Outra forma para tentar melhorar o desempenho da PSO é a hibridização do algoritmo original. Um exemplo é o trabalho de Lvbjerg *et al.* (2001). Com inspiração nos algoritmos genéticos, estes autores trabalham com uma forma de reprodução entre os indivíduos do enxame. Outra ideia testada em Lvbjerg *et al.* (2001) é o conceito de subpopulações, em que é realizada a divisão das partículas em várias subpopulações, para tentar manter a diversidade e, dessa forma, tentar fugir da convergência para mínimos locais.

Todas essas variações da PSO mostram-se eficientes, melhorando, em vários casos, o resultado obtido no processo de otimização, em comparação ao algoritmo original da PSO.

4 Inércia da partícula

A partir de agora, é dado um enfoque ao aspecto principal deste trabalho. Apresenta-se o termo de inércia do algoritmo original da PSO, sendo discutida uma forma por meio de que, alterando o modo original de atuação deste termo, é possível obter benefícios durante a otimização em relação a PSO padrão, para problemas complexos.

4.1 Termo de inércia

O peso de inércia tem características que são remanescentes do parâmetro temperatura no *Simulated Annealing* (Eberhart e Shi, 1998). Na PSO, ele foi introduzido a fim de balancear a pesquisa local e global. Um alto peso de inércia facilita uma busca global, enquanto um baixo peso de inércia favorece a busca local.

Na concepção original da PSO, o processo de otimização tem um bom desempenho com o termo de inércia decaindo linearmente (Shi e Eberhart, 1999). Tendo um valor de peso de inércia alto no início da execução e decrescendo linearmente para um valor pequeno durante o processo de otimização, a PSO terá, inicialmente, uma capacidade maior de pesquisa global, e obterá uma maior capacidade de busca local, ao aproximar-se do fim do processo. Cientes dessas características, já em seu trabalho inicial, Kennedy e Eberhart (1995) sugerem o peso de inércia monotonicamente decrescente. Esta é uma subtração linear entre os valores 0,9 e 0,4.

4.2 Controle de inércia

Uma característica observada em Shi e Eberhart (1999) é que a PSO pode necessitar da capacidade de pesquisa global no final de uma execução, dada a estagnação em um mínimo local, ao se utilizar um peso de inércia linearmente decrescente. Tal capacidade é ainda mais necessária, caso o problema a ser resolvido seja muito complexo. Portanto, na maioria das vezes, a PSO pode ser deficiente para encontrar o ótimo global para esses problemas.

Observando essa necessidade, foi introduzida uma modificação na forma de trabalhar com a inércia durante a execução da PSO. Nos testes iniciais, verificou-se que, ao impor um peso de inércia maior que 1, as partículas começavam a se espalhar pelo espaço de busca. Nesse caso, a velocidade do instante anterior passa a ter maior importância no deslocamento atual da partícula. Isso ocorre porque, ao multiplicar a velocidade anterior da partícula por um número maior que 1,0, a nova velocidade resultante será maior (terá maior importância) que a velocidade anterior, fazendo com que as partículas ganhem energia (aumentem a velocidade) e se afastem do ponto onde estavam.

Assim, em vez de a inércia w decair linearmente ao longo da otimização, ela passa a oscilar durante todo o processo de otimização, segundo uma função periódica com comportamento ondulatório. Para isso, com base na função cosseno, a inércia w da partícula assume esse comportamento não monotônico, de acordo com a equação 3:

$$w = \left[\cos \left(\frac{\pi i}{2 \text{ ciclos}} \right) \times m \right] + s \quad (3),$$

na qual i é a iteração corrente; *ciclos* corresponde ao número de ciclos não monotônicos utilizados; m é um multiplicador, gerado pela diferença do valor máximo de w pelo valor mínimo de w (informados antes do início do processo) que, depois, é dividida por 2 $((w_{\text{max}} - w_{\text{min}})/2)$, responsável por calcular a metade da altura da função para os valores de w utilizados. O deslocamento da função é calculado por s , que o faz com a soma de m com o valor mínimo de w $(m + w_{\text{min}})$. Com isso, pode-se deslocar a função no eixo y do plano cartesiano, o que não seria possível, se utilizada somente a fórmula original do cosseno. O comportamento desta função pode ser observado na Figura 1.

Com o valor do peso de inércia assumindo esse comportamento não monotônico, a otimização ocorreria da seguinte forma: quando o termo de inércia da PSO reduz, tendendo ao seu valor mínimo, as partículas perdem energia, se estabilizando em torno de um ponto de mínimo. Se este termo de inércia recebe valores maiores que 1, a tendência é a velocidade da partícula no instante anterior possuir maior importância, podendo causar um espalhamento do enxame. Se o termo de inércia atingir novamente valores menores, a tendência será o enxame se aproximar novamente, efetuando uma busca em torno de um mínimo local. Este mínimo, não será necessariamente igual ao primeiro. Portanto, o espalhamento do enxame pode ajudar o processo da PSO a fugir de mínimos locais, pois propicia uma busca global em um momento no qual seria menos provável encontrar outra melhor solução.

Com isso, a inércia w pode atuar na diversificação e intensificação da PSO. O número de oscilações da função que controla a inércia ao longo da PSO é o número de tentativas de fugas de mínimos locais, já que a cada vez que w ultrapassar 1, o enxame pode se espalhar.

Vale ressaltar que a PSO com controle de inércia não monotônico segue todos os princípios da PSO padrão. A única alteração ocorre durante a atualização do termo de inércia, feita pela equação 3, distinta da PSO padrão, que realiza uma subtração linear para a atualização desse termo.

Para melhor ilustrar o efeito desse ciclo não monotônico do peso de inércia neste trabalho, relatam-se e discutem-se os resultados experimentais com 4 funções de teste não lineares.

5 Configurações experimentais

Para a comparação entre os dois modelos, foram utilizadas 4 funções não lineares bem estudadas na literatura, ocorrendo, em todos os casos, problemas de minimização. As duas primeiras tratam de funções unimodais e as duas últimas, de funções multimodais. Todas apresentam seu mínimo global correspondente ao valor $(0,0,...,0)$, de acordo com o seu número de coordenadas n .

A primeira função proposta é a função *Sphere*, descrita pela seguinte equação:

$$f_1(x) = \sum_{i=1}^n x_i^2$$

Nesta, como em todas as funções, x é um vetor n -dimensional de números reais, e x_i é o i -ésimo elemento deste vetor.

A segunda função é a função *De Jong F4 – no noise*, dada pela equação:

$$f_2(x) = \sum_{i=1}^n i \cdot x_i^4$$

A terceira é a função *Rastrigin*, descrita pela equação:

$$f_3(x) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10)$$

A quarta e última função é a função *Griewank*, que é dada pela equação:

$$f_4(x) = \sum_{i=1}^n (x_i^2 / 4000) - \prod_{i=1}^n \cos(x_i / \sqrt{i}) + 1$$

A Tabela 1 lista o domínio de cada uma.

Tabela 1. Domínio da função.

Table 1. Function domain.

Função	Domínio
f_1	$-5.12 \leq x_i \leq 5.12$
f_2	$-20 \leq x_i \leq 20$
f_3	$-5.12 \leq x_i \leq 5.12$
f_4	$-600 \leq x_i \leq 600$

Para ambos os modelos de PSO, o número de iterações para cada função foi fixado em 2000, para cada uma das cinco dimensões testadas, correspondendo a 10, 20, 40, 60 e 80 coordenadas. O tamanho da população em cada execução foi estabelecido em 1000, sendo gerada a partir de uma distribuição uniforme, dentro dos intervalos especificados na Tabela 1. Os valores de c_1 e c_2 foram definidos iguais a 1,4. A vizinhança do tipo estrela foi utilizada.

Para a PSO padrão, conservando a ideia original do algoritmo, o valor da inércia w foi definido como uma subtração linear entre os valores 0,9 e 0,4.

Para a PSO com o controle de inércia w , encontram-se os seguintes parâmetros: o valor superior de w foi fixado em 1,2; o valor inferior de w foi 0,4 e os ciclos não monotônicos são concluídos a cada 320 iterações. Todos esses valores

paramétricos da PSO com controle de inércia foram obtidos por intermédio de um processo de ajuste paramétrico com base em testes estatísticos sobre a função *Rastrigin*, em que se fez a análise dos resultados obtidos. Esse processo é semelhante ao descrito em Oliveira *et al.* (2006).

Uma característica observada durante o estabelecimento de tais parâmetros para a PSO com o controle de inércia concerne ao número de ciclos não monotônicos ao longo da otimização. A utilização de muitas iterações para completar um ciclo não monotônico não resulta em um bom desempenho do algoritmo, pois a otimização se torna demorada, em vista do elevado número de iterações para o espalhamento das partículas. A utilização de poucas iterações para completar um ciclo não monotônico também não resulta em um bom desempenho do algoritmo, o que se deve à rápida convergência do termo de inércia. Uma convergência muito rápida para o valor mínimo da inércia pode fazer com que o mínimo local encontrado se torne, muitas vezes, o melhor ponto social e cognitivo das partículas. Se a maioria das partículas tiver essa característica na primeira redução do termo de inércia, a probabilidade de encontrar soluções mais favoráveis será reduzida, pois as partículas se espalharão somente a partir desse melhor ponto, fazendo com que o processo de otimização não seja tão diversificado como na PSO padrão, fato que pode comprometer o resultado.

A Figura 1 mostra quais os valores a inércia w assume ao longo da execução do algoritmo para as configurações apresentadas, tanto para o modelo padrão, com um decrescimento linear dessa inércia, quanto para o modelo com o controle da inércia da partícula, com as oscilações não monotônicas.

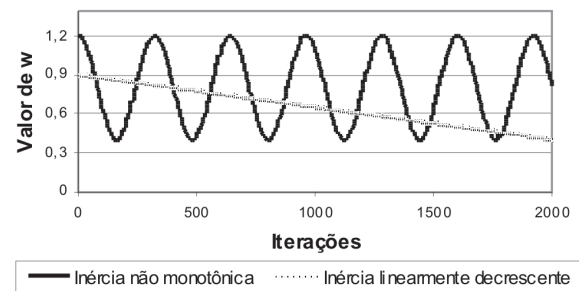


Figura 1. Valor da inércia w da partícula ao longo da execução do algoritmo.

Figure 1. Inertia value of the w particle during the algorithm execution.

Um total de 30 testes em ambos os modelos, para cada configuração experimental, foi conduzido.

6 Resultados e discussões

Nesta seção, são comparados os resultados obtidos da PSO padrão e a PSO com o controle de inércia w ,

para as funções de benchmark descritas. Para isso, são explorados os efeitos da complexidade do problema, com a necessidade de espalhamento das partículas.

A Tabela 2 mostra o resultado obtido na execução dos dois modelos da PSO, o padrão e o com controle de inércia. Ela lista a função utilizada para o espaço de busca, e, para sua dimensão, os respectivos valores médios da melhor partícula encontrada durante o processo de otimização, além do teste estatístico utilizado para a comparação dos resultados. Mostra, também, se houve diferença significativa entre esses resultados.

Para a análise dos resultados, foi, inicialmente, aplicado o teste de Shapiro-Wilks, a fim de verificar a normalidade dos dados. Quando a distribuição normal foi aceita, aplicou-se o teste *t* de Student (teste paramétrico). Nas situações em que a distribuição normal não pôde ser aceita, foi aplicado o teste U de Mann-Whitney (teste não paramétrico). As diferenças entre os resultados foram consideradas estatisticamente significantes, quando o valor de “p” (*p-value*) foi menor que 0,05 (95% de confiança).

As Figuras de 2 a 6 são os gráficos que ilustram alguns experimentos mostrados na Tabela 2; mostram o desenvolvimento da otimização, para cada função utilizada com dimensão 60. No eixo y do gráfico, verifica-se a avaliação da partícula, enquanto, no eixo x, observam-se as iterações do processo de otimização. A Figura 6 mostra a mesma otimização apresentada na Figura 4, porém em uma escala maior.

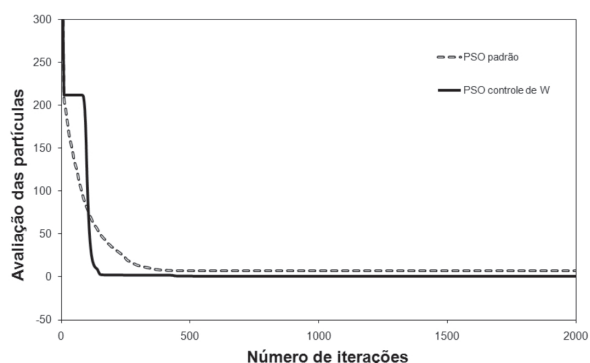


Figura 2. PSO padrão *versus* PSO com controle de inércia para a função *Sphere* com dimensão 60.

Figure 2. Standard *versus* Inertia Control PSO to the Sphere function with dimension 60.

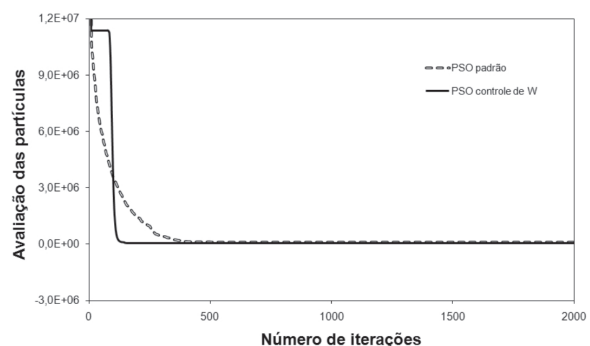


Figura 3. PSO padrão *versus* PSO com controle de inércia para a função *De Jong F4* – no noise com dimensão 60.

Figure 3. Standard *versus* Inertia Control PSO to the *De Jong F4* function (no noise) with dimension 60.

Tabela 2. Média da avaliação da melhor partícula
Table 2. Mean evaluation of the best particle.

Função	Dimensões	PSO padrão	PSO com controle de inércia	Teste estatístico	Diferença entre os resultados
f_1	10	0.000×10^0	0.000×10^0	t de Student	Insignificante
	20	0.000×10^0	0.000×10^0	t de Student	Insignificante
	40	0.000×10^0	4.807×10^{-5}	U de Mann-Whitney	Significante
	60	6.991×10^0	8.882×10^{-1}	U de Mann-Whitney	Significante
	80	2.447×10^1	6.421×10^0	U de Mann-Whitney	Insignificante
f_2	10	0.000×10^0	0.000×10^0	t de Student	Insignificante
	20	0.000×10^0	0.000×10^0	t de Student	Insignificante
	40	5.333×10^3	9.575×10^{-6}	U de Mann-Whitney	Significante
	60	1.067×10^5	3.200×10^4	U de Mann-Whitney	Significante
	80	6.667×10^5	2.988×10^5	U de Mann-Whitney	Insignificante
f_3	10	0.000×10^0	8.623×10^{-1}	U de Mann-Whitney	Significante
	20	7.796×10^0	7.428×10^0	U de Mann-Whitney	Insignificante
	40	9.366×10^1	7.263×10^1	U de Mann-Whitney	Significante
	60	2.276×10^2	2.001×10^2	t de Student	Significante
	80	3.866×10^2	3.024×10^2	t de Student	Significante
f_4	10	2.575×10^{-2}	5.208×10^{-2}	t de Student	Significante
	20	2.305×10^{-2}	2.304×10^{-2}	U de Mann-Whitney	Insignificante
	40	7.957×10^{-3}	6.278×10^{-2}	U de Mann-Whitney	Significante
	60	1.505×10^1	4.961×10^{-1}	U de Mann-Whitney	Significante
	80	7.528×10^1	2.263×10^1	U de Mann-Whitney	Insignificante

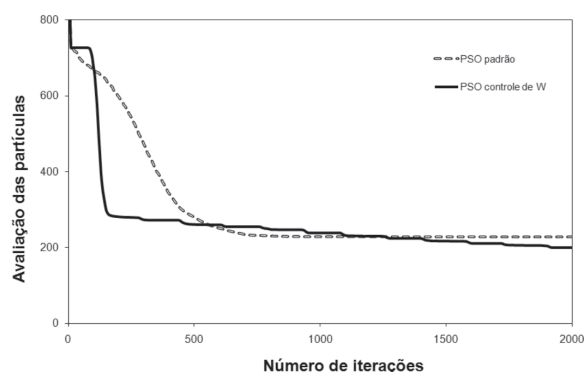


Figura 4. PSO padrão *versus* PSO com controle de inércia para a função *Rastrigin* com dimensão 60.
Figure 4. Standard *versus* Inertia Control PSO to the *Rastrigin* function with dimension 60.

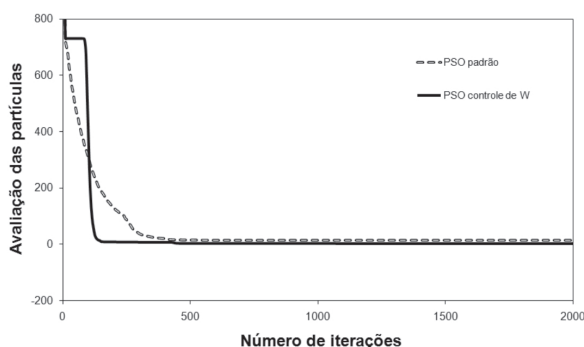


Figura 5. PSO padrão *versus* PSO com controle de inércia para a função *Griewank* com dimensão 60.
Figure 5. Standard *versus* Inertia Control PSO to the *Griewank* function with dimension 60.

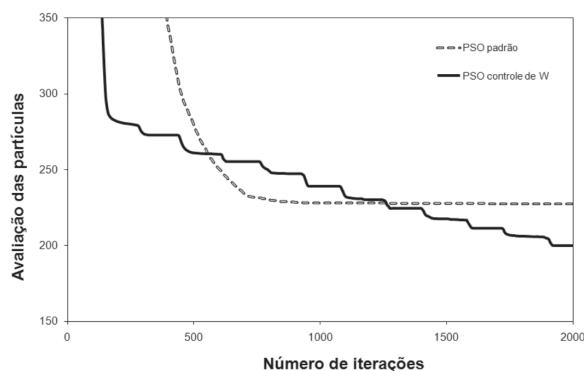


Figura 6. PSO padrão *versus* PSO com controle de inércia para a função *Rastrigin* com dimensão 60 em escala ampliada.
Figure 6. Standard *versus* Inertia Control PSO to the *Rastrigin* function with dimension 60 (amplified scale).

Observando as Figuras de 2 a 5, nota-se que, no início do processo, há uma pequena e rápida otimização. Isso se deve à aleatoriedade do sistema quanto à posição e a velocidade das partículas, que é interrompida mo-

mentaneamente pela estabilização das velocidades das partículas do sistema e pelo termo de inércia, que se inicia com o valor acima de 1,0.

Um ponto observado pelas Figuras de 2 a 6 é a velocidade de convergência e a capacidade de encontrar uma boa solução da PSO com controle de inércia não monotônico. Como o termo de inércia chega ao seu ponto de mínimo por volta da iteração 160, a PSO com o controle de inércia converge rapidamente para uma boa solução nessa iteração, em todos os casos. Para problemas simples, a melhor solução já seria encontrada, dispensando, assim, o espalhamento das partículas. Para problemas complexos, encontra certamente um mínimo local. Nesse ponto, o espalhamento das partículas seria interessante para tentar encontrar uma solução mais promissora.

Uma característica sensível do controle de inércia não monotônico é sua capacidade de refinamento, que é bem compreendida ao se observar a Figura 6. O controle de inércia não monotônico, como visto, tem uma rápida capacidade de encontrar uma boa solução, mas sua capacidade de refinamento é mais lenta que a PSO padrão. Pela Figura 6, nota-se que, durante todas as iterações, a PSO com controle de inércia ainda tentava buscar um melhor ponto de estabilização, enquanto a PSO padrão já havia se estabilizado por volta da iteração 1000. Essa característica deve ser considerada, pois, na maioria dos problemas, a estabilização da PSO com controle de inércia é mais demorada, principalmente para problemas mais complexos. Uma explicação para esse fato é que, a partir do momento de espalhamento das partículas até que a inércia volte a assumir valores menores que 1,0, em tais iterações, não há a busca por melhores soluções, mas o afastamento do melhor ponto já encontrado. Isso gera certa demora para encontrar a solução desejada.

Pelas análises efetuadas, ambos os métodos encontram um bom resultado durante a otimização, mas, por meio do controle de inércia não monotônico da partícula, ter-se-á uma probabilidade maior de fuga de um mínimo local encontrado, dada a capacidade de busca global gerada pela energia inserida no sistema pelo controle de inércia. Isso pode, em vários casos, levar a otimização a resultados mais satisfatórios, principalmente para problemas complexos.

7 Redes Neurais Artificiais otimizadas com a PSO

Conforme já mencionado, a PSO tem, em sua concepção original, a resolução de problemas de otimização que possuem um domínio de soluções pertencente ao conjunto dos números reais. Logo, uma grande variedade

de problemas desse tipo pode ser modelada, de modo que a PSO seja aplicável.

Assim, como forma de mostrar a aplicabilidade e verificar a eficiência que ambas as PSOs descritas no trabalho desempenham sobre um problema real, é apresentado um clássico problema da área de inteligência artificial que foi modelado para a arquitetura da PSO: o problema de configuração de pesos sinápticos de redes neurais artificiais *Multi-Layer Perceptron* (MLP). Para um melhor entendimento, são apresentadas algumas definições básicas sobre redes neurais artificiais e as redes MLP.

7.1 Redes Neurais Artificiais

As Redes Neurais Artificiais (RNAs) são modelos matemáticos que tiveram seu desenvolvimento inspirado no comportamento de uma rede neural biológica, na qual o conhecimento é adquirido por meio da experiência com um conjunto de dados.

Uma RNA é composta por unidades simples de processamento, em que essas se conectam com outras unidades para realizar cálculos de funções, geralmente não lineares. Com isso, elas têm a propensão de armazenar conhecimento e torná-lo disponível para ser utilizado quando necessário (Haykin, 1998).

O modelo de neurônio artificial mais conhecido é o proposto por McCulloch e Pitts (1943), composto por uma unidade de n entradas e uma saída. As unidades simples de processamento (neurônios) se interligam por meio de conexões geralmente unidirecionais, chamadas de sinapses. No modelo biológico, sinapse é a região onde dois neurônios entram em contato e por meio da qual os impulsos nervosos são transmitidos entre eles. Para o modelo matemático, a sinapse representa um valor que é utilizado para ponderar as entradas recebidas de cada unidade a elas conectadas. Tais pesos representam o conhecimento adquirido que está armazenado por essa RNA.

Outra característica apresentada por um neurônio artificial é um peso individual denominado bias e sua função de ativação. A função de ativação calcula o peso que cada neurônio terá para o processamento de uma determinada entrada, enquanto o peso bias é responsável por controlar a entrada líquida dos valores na função de ativação do neurônio artificial.

7.2 Multi-Layer Perceptron

Dentre os modelos de RNAs, um dos mais conhecidos e utilizados, que é a base de utilização

neste trabalho, é o Perceptron de Múltiplas Camadas, ou MLP (*Multi-Layer Perceptron*). Esta rede neural formou-se como uma extensão do modelo Perceptron de uma Camada, proposto por Rosenblatt (1962), que adquire o conhecimento mediante um aprendizado supervisionado. Uma característica importante das MLPs é que a introdução de camadas de Perceptron permite generalizar a rede, de modo a conseguir tratar problemas de não separabilidade linear, comuns, por exemplo, em problemas de reconhecimento de padrões (Carvalho, 2007).

Segundo Haykin (1998), a arquitetura de uma MLP é representada por:

- *Camada de entrada*: camada única, representada pelo número de componentes da base de conhecimento que são as entradas para a rede. Essa camada não possui pesos ajustáveis;
- *Camada intermediária*: possui uma ou mais camadas nessa fase. Nesse caso, a quantidade de camadas intermediárias (ou escondidas) são independentes da base de conhecimento utilizada, das entradas e das saídas. São responsáveis por representar a não linearidade apresentada pelo conjunto de entradas e de saídas da rede.
- *Camada de saída*: camada única, que possui uma quantidade de neurônios correspondente ao número de componentes do espaço de saída. Aqui, cada neurônio possui pesos ajustáveis, além de utilizar da função de ativação.

Assim, a MLP consegue desenvolver complexas estruturas, conferindo-lhe a capacidade de resolver problemas de não separabilidade linear, o que é importante para o problema de reconhecimento de padrões. Este é abordado neste trabalho, já que assume exatamente essas características.

7.3 Modelagem da MLP via PSO

Por serem problemas com representações distintas, é necessário fazer com que as redes neurais tenham seus valores mapeados para a arquitetura da PSO, para que a otimização seja possível. Assim, a forma de representação de cada peso sináptico e bias de um neurônio artificial da rede MLP é expressa no vetor de números reais de cada partícula do enxame, onde cada peso ou bias da rede tem sua representação em uma determinada posição desse vetor. A Figura 7, com base em Carvalho (2007), apresenta como é esse mapeamento, para a utilização de apenas uma camada escondida na rede MLP.

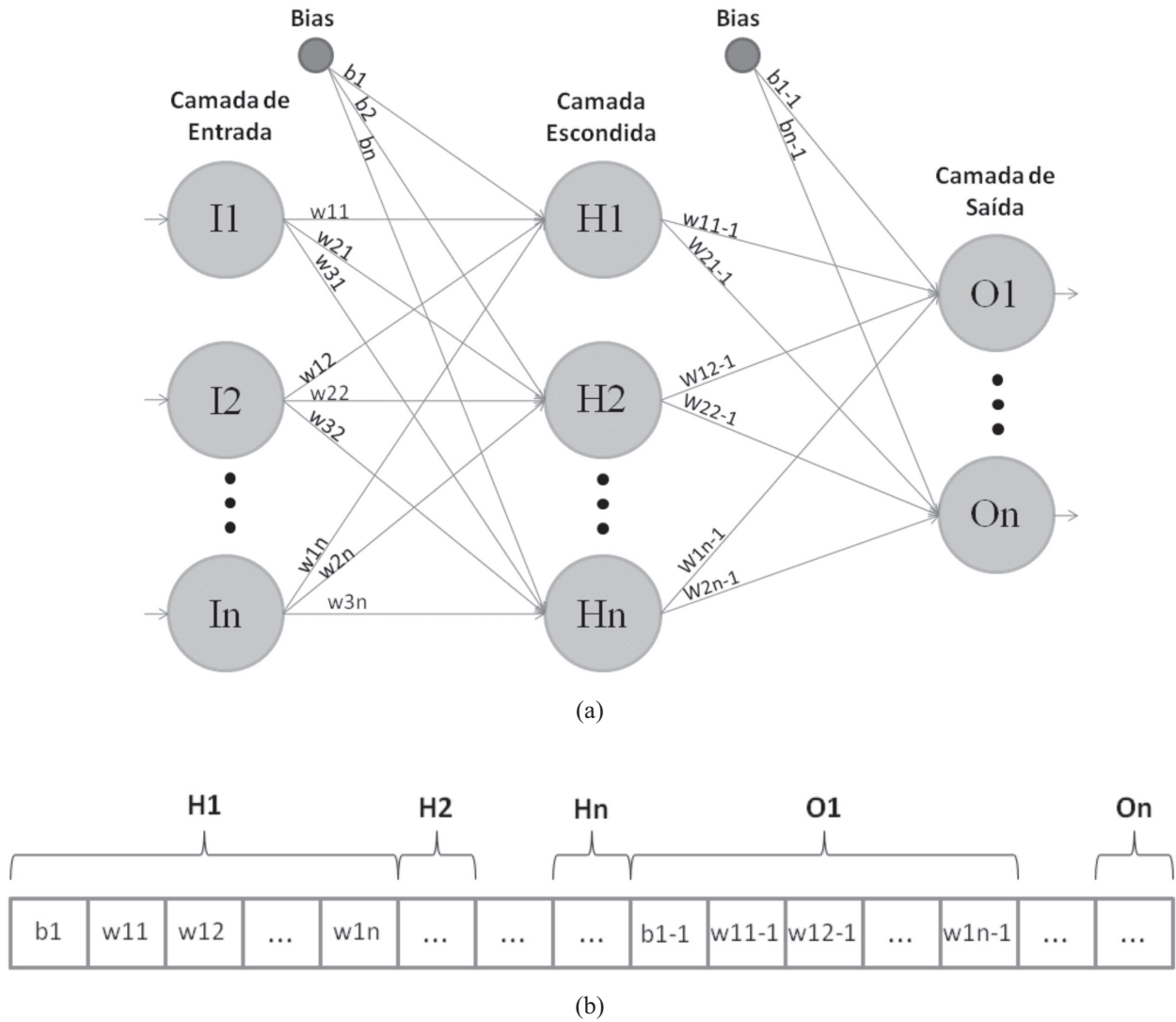


Figura 7. Modelagem do problema de representação de uma rede neural artificial, em que os (a) pesos sinápticos e bias da rede são mapeados para (b) vetor de números reais utilizados pela PSO.

Figure 7. Artificial Neural Network representation – Synaptic weights and bias of the network (a) are mapped to the vector used in the PSO (b).

Nesse caso, a camada de entrada da rede MLP não possui seus pesos representados no vetor utilizado pela PSO, pois esses pesos correspondem aos padrões de entrada para a RNA e são utilizados somente para calcular qual é a saída da rede neural para os pesos sinápticos oriundos do treinamento.

A dimensão do problema a ser tratado pela PSO varia em função do número de pesos que compõem a rede. Considerando uma arquitetura MLP com uma única camada escondida, a dimensão do problema é determinada pela equação 4, sendo I o número de neurônios da camada de entrada; H , o número de neurônios da única camada escondida e O , o número de neurônios da camada de saída. As constantes da equação 4 representam

o número de bias presentes na camada escondida e na camada de saída da rede.

$$\text{Dimensão} = (I + 1) \cdot H + (H + 1) \cdot O \quad (4)$$

Esta dimensão assumida é interessante para verificar o tamanho do problema que está sendo tratado pela PSO.

7.4 Estudo de caso

Os parâmetros de configuração de ambos os modelos de PSO, o padrão e o com controle de inércia não monotônico, seguiram basicamente a mesma configuração

do experimento anterior com as funções de benchmark não lineares (*Sphere*, *De Jong F4 – no noise*, *Rastrigin*, *Griewank*). Para a PSO padrão, o termo de inércia segue o decaimento linear padrão de 0,9 para 0,4. Para a PSO com controle de inércia não monotônico, o valor superior de w utilizado continua sendo 1,2; o valor inferior de w de 0,4; e os ciclos não monotônicos concluídos a cada 320 iterações. Para ambos os modelos, os valores também se mantêm como no experimento anterior, com c_1 e c_2 iguais a 1,4, utilizando a vizinhança do tipo estrela em uma otimização com 2000 iterações. A alteração em relação ao experimento anterior está ligada ao tamanho da população, agora definida em 30 partículas, para um espaço de busca entre os intervalos de $-1,0$ a $1,0$.

A rede neural foi testada com 5 configurações diferentes, fixados o número de neurônios da camada de entrada (9 neurônios, correspondendo ao número de características de entrada) e o número de neurônios da camada de saída (1 neurônio, equivalente à saída unária da rede). A camada escondida foi testada com 5, 10, 15, 20 e 25 neurônios, correspondendo a um problema de dimensão 56, 111, 166, 221 e 276, respectivamente. A ativação do neurônio utilizou a função tangente hiperbólica.

A base de dados usada para treinamento da rede MLP foi obtida de repositório Proben1 (Prechelt, 1994), correspondendo a dados do problema de classificação de padrões da base de Câncer. Esta base está constituída de 699 padrões normalizados (dispostos de 3 modos diferentes, formando os padrões Câncer1, Câncer2 e Câncer3), que são classificados como tumores benigno ou maligno de um diagnóstico microscópico de câncer de mama. Tais padrões foram obtidos dos hospitais da Universidade de Wisconsin, EUA (Wolberg e Mangasarian, 1990).

Para o problema de ajuste dos pesos sinápticos da rede MLP abordado, os padrões foram divididos em 2 subconjuntos: subconjunto de treinamento, correspondendo a 50% dos padrões, onde a PSO conhece tais dados para realizar o ajuste dos pesos sinápticos da rede neural; e subconjunto de teste, equivalendo a 25% dos padrões, com a base não conhecida pela rede neural, que é utilizada para verificar o aprendizado que a rede obteve durante o ajuste de seus pesos sinápticos e bias. É importante destacar que 25% dessa base não foi utilizada, pois se trata dos dados utilizados para validação cruzada (*crossval*). Para a PSO com controle de inércia, essa validação não seria interessante, dada a sua oscilação no termo de inércia provocar o espalhamento das partículas para a fuga dos mínimos locais durante todo o processo, o que dificultaria a obtenção do momento exato de interromper a sua otimização. Outro ponto a ser observado é a saída da rede, que foi tratada como um problema de classificação unária (1 classe).

Para a evolução do enxame durante a otimização, a função objetivo foi baseada na minimização do erro quadrático médio normalizado de todos os padrões de treinamento, conforme a seguinte equação:

$$f_5(x) = \frac{100}{N} \sum_{i=1}^N (\text{esperado}_i - \text{obtido}_i)^2,$$

em que N é o número de padrões para treinamento; *esperado* é o valor real da saída para o padrão de treinamento correspondente; e *obtido* é o valor encontrado na saída. Ao realimentar a RNA com o padrão de treinamento corrente, para os pesos sinápticos encontrados pela PSO.

Após o treinamento, a RNA foi realimentada com os padrões de teste. Para verificar a generalização da rede com a utilização das PSOs, foi utilizado o erro percentual de classificação, descrito pela seguinte equação:

$$f_6(x) = \frac{100}{N} \text{errosClassificação},$$

em que N é o número de padrões para teste e *errosClassificação* é o total de erros cometidos para aquele conjunto de padrões.

Novamente, foi conduzido um total de 30 testes em ambos os modelos, para cada configuração experimental.

7.5 Resultados e discussão

Esta seção aborda os resultados obtidos da configuração dos pesos sinápticos da rede neural artificial MLP ajustada com ambas as PSOs apresentadas durante o trabalho, mostrando as características de cada modelo e uma comparação com os melhores resultados obtidos na literatura para a resolução deste problema, ao utilizar a mesma base de dados (base Câncer, antes descrita).

A Tabela 3 mostra as taxas médias de acerto obtidas, utilizando as respectivas bases de teste, após o treinamento da RNA MLP pelas PSOs. Para cada base de treinamento, é exposta, para o número de neurônios que foi utilizado na camada escondida, a porcentagem de acerto para esta configuração e o seu desvio-padrão (entre parênteses). Para cada base de dados, a melhor média obtida pelas PSO, independente do número de neurônios utilizados, é indicada em negrito. Mostra-se, também, a informação sobre a diferença entre as melhores médias de cada abordagem (PSO padrão e PSO com controle de inércia) para a mesma base de dados para treinamento, seguindo os mesmos critérios descritos na Seção 6.

Tabela 3. Média da avaliação da taxa de acertos dos testes de reconhecimento de padrões após o treinamento da rede neural pelas PSOs.

Table 3. Mean evaluation of the right matches in pattern recognition after training the neural network by PSOs.

Base de treinamento	Número de neurônios na Camada Escondida	PSO padrão – Porcentagem de Acertos	PSO com controle de inércia – Porcentagem de Acertos	Diferença entre as melhores médias
<i>Câncer1</i>	5	97,72031 (0,554199)	97,77778 (0,579318)	Insignificante
	10	97,85441 (0,397409)	97,54789 (0,542778)	
	15	97,85441 (0,335225)	97,47126 (0,442605)	
	20	97,64368 (0,461501)	97,70115 (0,452781)	
	25	97,49042 (0,412871)	97,52874 (0,403571)	
<i>Câncer2</i>	5	95,05747 (0,514039)	95,01916 (0,697102)	Insignificante
	10	95,78544 (0,435689)	95,51724 (0,664764)	
	15	95,9387 (0,583237)	95,68966 (0,446448)	
	20	96,18774 (0,353418)	95,95785 (0,533251)	
	25	96,03448 (0,485553)	96,05364 (0,391635)	
<i>Câncer3</i>	5	94,78927 (0,521372)	94,61686 (0,731386)	Insignificante
	10	94,92337 (0,454872)	94,82759 (0,544175)	
	15	95,38314 (0,59388)	95,11494 (0,617695)	
	20	95,22989 (0,625027)	95,0000 (0,547305)	
	25	95,32567 (0,516984)	95,03831 (0,631072)	

Pela Tabela 3, pode-se observar que a base Câncer1 ajustada via PSO foi a que obteve uma melhor capacidade de generalização, quando aplicada a respectiva base de testes, enquanto a base Câncer3, apesar do seu bom desempenho, obteve uma menor capacidade de generalização em relação às outras. A base Câncer2 mostra um fato interessante: apresenta uma taxa de mais de 1% de acerto para ambas as PSOs, variando somente o número de neurônios na camada escondida, enquanto para as outras duas bases essa variação teve menor relevância.

Outro fato importante pode ser observado a partir da Tabela 3. Apesar de os testes estatísticos indicarem que, para todos os resultados das melhores médias entre os dois modelos de PSOs para a mesma base de treinamento, a diferença entre estas médias foi igual, a maior parte das médias obtidas pela PSO com controle de inércia tiveram uma menor taxa de acerto, se comparado aos resultados da PSO padrão. Uma possível explicação seria a complexidade do problema a ser resolvido. Apesar de o problema de reconhecimento de padrões para a base de Câncer não ser de fácil resolução, este não apresenta a complexidade como das funções de benchmark descritas na Seção 5. Mesmo otimizando um problema com uma dimensão grande (utilizando 25 neurônios na camada escondida, equivalendo a uma dimensão de 276 para a PSO, por exemplo), a complexidade do problema não sofreu muita alteração. Assim, como foi concluído pelos resultados obtidos dos testes da Seção 6, a PSO com controle

de inércia não monotônico poderia ter um desempenho inferior à PSO padrão, em vista de a fuga de um mínimo local não influenciar tanto a busca por melhores resultados para problemas que tenham uma menor complexidade.

Os resultados obtidos mostram-se bastante atraentes, se comparados aos melhores resultados obtidos na literatura. A Tabela 4 faz uma comparação entre os melhores valores obtidos por ambas as PSOs e os melhores resultados obtidos da literatura (Ferreira *et al.*, 2009), para a mesma base de dados. Para tal, como base de comparação, foram utilizados os seguintes classificadores:

- Sigm-Tree (ST1 a ST3): sistema de classificação com base em árvores sintáticas (cada um seguindo uma abordagem específica);
- Layered Genetic Programming (ES1 a ES6): sistema de classificação com base em programação genética multipopulacional (cada um utiliza uma variação paramétrica);
- Decision Tree (DT): sistema com base em árvores de decisão;
- Fuzzy Rule-Based System (FRBS): sistema de classificação *Fuzzy* com base em regras;
- Artificial Neural Network (ANN): sistema com base em uma RNA MLP, com treinamento pelo algoritmo de *Backpropagation*;
- Fuzzy Petri-Nets (FPN): sistema com base em redes *fuzzy-petri* com programação genética;

Tabela 4. Média e desvio-padrão da avaliação da taxa de acertos dos testes de reconhecimento de padrões utilizando vários classificadores.

Table 4. Mean and standard deviation of the evaluation of the right matches in pattern recognition using multiple models.

	Câncer1		Câncer2		Câncer3	
	Média	Desv-pad	Média	Desv-pad	Média	Desv-pad
PSO padrão	97,85	0,34	96,19	0,35	95,38	0,59
PSO controle de w	97,78	0,58	96,05	0,39	95,11	0,62
ST1	97,78	0,78	94,64	0,92	95,92	0,86
ST2	98,23	0,65	94,82	0,79	95,69	0,81
ST3	97,62	1,43	95,42	0,73	95,74	0,67
ES1	97,7	0,72	94,89	0,69	96,32	0,78
ES2	97,7	0,54	94,6	0,48	96,09	0,71
ES3	97,82	0,85	94,89	0,92	96,38	0,61
ES4	97,76	0,79	94,94	0,59	96,61	0,57
ES5	97,7	0,77	94,77	0,74	96,32	0,4
ES6	97,82	0,8	94,83	0,47	96,03	0,69
DT	96,21	1,01	95,32	2,18	95,61	1,36
FRBS	95,61	1,42	95,55	1,23	95,1	0,83
ANN	94,34	1,24	91,7	2,16	94,72	1,7
FPN	95,69	0,94	95,17	1,19	95,58	1,43

Apesar de os resultados de ambas as PSOs serem estatisticamente iguais, a Tabela 4 mostra-os separados, para uma melhor avaliação na relação com as outras técnicas presentes na literatura. Deve-se observar também que, para cada base de dados utilizada, a melhor média obtida está destacada em negrito.

As Figuras de 8 a 11 mostram o decaimento do erro cometido pela rede durante o treinamento, ao utilizar a função f_5 como função objetivo. Observa-se que, como nos resultados da Seção 6, as PSOs têm, inicialmente, uma rápida convergência para uma boa solução, dificultando a busca por melhores soluções a partir de certa parte do processo de otimização. Novamente, a PSO com controle de inércia tem um refinamento da solução mais demorado que a PSO padrão, em razão das iterações gastas para o espalhamento das partículas (conforme exposto na Seção 6).

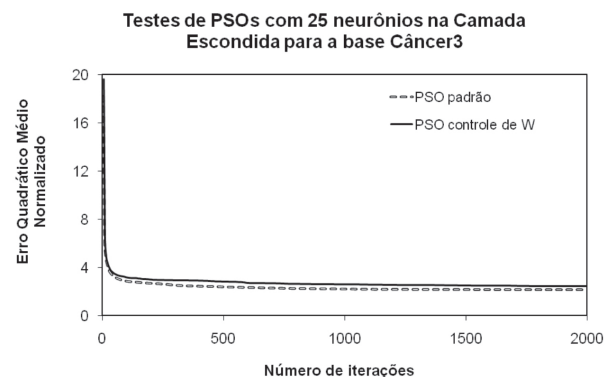


Figura 8. PSO padrão *versus* PSO com controle de inércia na otimização dos pesos sinápticos para a base Câncer3, com 25 neurônios na Camada Escondida.

Figure 8. Standard *versus* Inertia Control PSO in the ANN Optimization to the Cancer3 Data Base. The ANN was made with 25 neurons in the hidden layer.

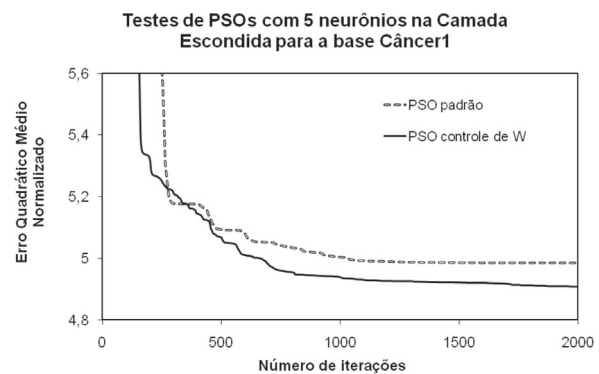


Figura 9. PSO padrão *versus* PSO com controle de inércia na otimização dos pesos sinápticos para a base Câncer1, com 5 neurônios na Camada Escondida.

Figure 9. Standard *versus* Inertia Control PSO in the ANN Optimization to the Cancer1 Data Base. The ANN was made with 5 neurons in the hidden layer.

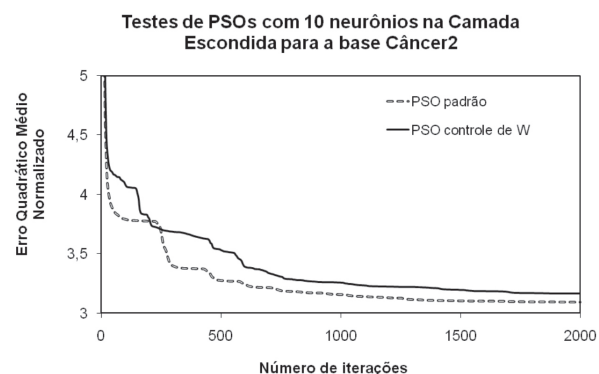


Figura 10. PSO padrão *versus* PSO com controle de inércia na otimização dos pesos sinápticos para a base Câncer2, com 10 neurônios na Camada Escondida.

Figure 10. Standard *versus* Inertia Control PSO in the ANN Optimization to the Cancer2 Data Base. The ANN was made with 10 neurons in the hidden layer.

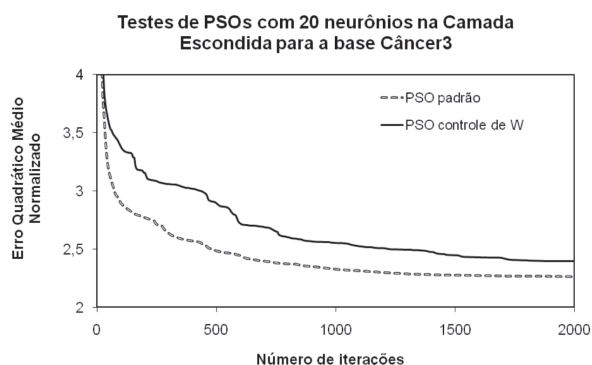


Figura 11. PSO padrão *versus* PSO com controle de inércia na otimização dos pesos sinápticos para a base Câncer3, com 20 neurônios na Camada Escondida.

Figure 11. Standard *versus* Inertia Control PSO in the ANN Optimization to the Cancer3 Data Base. The ANN was made with 20 neurons in the hidden layer.

Assim, ambas as PSOs mostram-se como atrativos mecanismos de otimização para aplicação em problemas reais de otimização, e, apesar de suas abordagens gerais, apresentaram resultados bastante satisfatórios em relação a outras técnicas de otimização para um problema real.

8 Conclusões e trabalhos futuros

Neste trabalho, apresentou-se um modo diferente de trabalhar com a Otimização por Enxame de Partículas. Trata-se de uma forma de controlar a inércia da partícula de modo a permitir, muitas vezes, a fuga de soluções ótimas locais. Esse controle teve base em uma função não monotônica, utilizando, para isso, a função cosseno, o que difere do modelo original, que é linearmente decrescente. Para os experimentos de comparação entre os métodos, foram utilizadas 4 funções de benchmark para o espaço de busca, bastante conhecidas na literatura. Pelos testes, assim como em Eberhart e Shi (1998), verificou-se que, algumas vezes, a PSO necessita de uma capacidade de busca global próximo ao fim do experimento, principalmente para problemas complexos, pelo fato de ser usado um termo de inércia que decai linearmente. Por essa razão, ao se utilizar um termo de inércia não monotônico, obtém-se essa capacidade de busca global várias vezes ao longo do processo de otimização. Isso, segundo o que se verifica, pode trazer benefícios à otimização. Este é o ponto que pode fazer com que a PSO com controle de inércia não monotônico possibilite uma solução mais promissora, principalmente para os problemas complexos.

Após a comparação entre os dois modelos de PSOs, um problema real foi aplicado para a verificação do desempenho de ambos os métodos. O problema modelado foi o de configuração de pesos sinápticos de redes neurais artificiais *Multi-Layer Perceptron* (MLP) para o reconhecimento de padrões, ao se utilizar a base de dados

de Câncer do repositório Proben1 (Prechelt, 1994). Como constatado, ambas as PSOs tiveram um desempenho satisfatório, em comparação aos melhores resultados obtidos na literatura, o que torna sua utilização interessante em problemas desse tipo. Além disso, confirma a facilidade de modelagem e implementação da PSO. Contudo, vale ressaltar que, apesar do bom desempenho obtido nesta situação, a PSO com controle de inércia não obteve um desempenho superior em relação à PSO padrão possivelmente em vista da menor complexidade do problema de classificação de padrões em comparação com as funções de benchmark utilizadas nos testes anteriores.

Para trabalhos futuros, refinamentos no método proposto e na aplicação ao problema real de reconhecimento de padrões serão elaborados. Em relação ao método proposto, a meta será desenvolver e testar novas funções para o controle da inércia, a fim de verificar o desempenho da otimização sobre as funções não lineares do espaço de busca. Novas pesquisas deverão, também, investigar formas de melhorar a capacidade de refinamento da solução da PSO com o controle de inércia, que, como se demonstrou, ainda apresenta um menor desempenho em relação à PSO padrão, podendo levar a resultados inferiores. Além disso, outros valores dos parâmetros da PSO, como c_1 , c_2 , dimensão do problema e tamanho da população, deverão ser investigados para se tentar obter uma melhor otimização desses problemas.

Em relação ao problema de reconhecimento de padrões, novas configurações serão propostas. Testes com saída binária para a base de Câncer serão feitos para avaliar se há influência nos resultados. Serão aplicadas também novas funções de avaliação de erro de treinamento e funções de ativação do neurônio para a verificação do desempenho. Por fim, métodos de otimização local poderão ser realizados para tentar melhorar o desempenho da PSO para os resultados obtidos, além concretizados testes com novas bases de dados, a fim de verificar a generalidade do método.

Referências

- ANGELINE, P.J. 1998. Using selection to improve particle swarm optimization. In: IEEE INTERNATIONAL CONFERENCE ON EVOLUTIONARY COMPUTATION, Anchorage, 1998. *Anais...* Anchorage, p.84-89.
- CARVALHO, M.R. 2007. *Uma análise da otimização de Redes Neurais MLP por enxame de partículas*. Recife, PE. Dissertação de Mestrado. Universidade Federal de Pernambuco, 66 p.
- COELLO, C.A.; TOSCANO, G.; LECHUGA, M.S. 2004. Handling multiple objectives with particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 8(3):256-279.
- EBERHART, R.C.; SHI, Y.H. 1998. Comparison between genetic algorithms e particle swarm optimization. In: ANNUAL CONFERENCE ON EVOLUTIONARY PROGRAMMING, 7, San Diego, 1998. *Anais...* San Diego, p. 611-616.

- FERREIRA, J.O.; OLIVEIRA, H.C.B.; PAULA, M.M.V. 2009. Sigm-Tree: otimização de árvores sintáticas aplicadas à classificação de padrões. In: SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL, XLI, Porto Seguro, 2009. *Anais...* Porto Seguro, XLI SBPO, p. 1478-1489.
- HAYKIN, S. 1998. *Neural Networks: A comprehensive foundation*. 2^a ed., Ontario, Prentice Hall, 842 p.
- JIAO, B.; LIAN, Z.; GU, X. 2008. A dynamic inertia weight particle swarm optimization algorithm. *Chaos, Solitons & Fractals*, 37(3):698-705.
- KENNEDY, J. 1999. Small worlds and mega-minds: Effects of neighborhood topology on particle swarm performance. In: CONGRESS OF EVOLUTIONARY COMPUTATION, Washington, DC, 1999. *Anais...* Piscataway, NJ, IEEE Press, 3:1931-1938.
- KENNEDY, J.; EBERHART, R.C. 1995. Particle swarm optimization. In: IEEE INTERNATIONAL CONFERENCE ON NEURAL NETWORKS, Perth, Australia, 1995. *Anais...* Piscataway, NJ, *IEEE Service Center*, IV:1942-1948.
- KENNEDY, J.; EBERHART, R.C.; SHI, Y. 2001. *Swarm intelligence*. San Francisco, Morgan Kaufmann Publishers, 512 p.
- LVBJERG, M.; RASMUSSEN, T.K.; KRINK, T. 2001. Hybrid particle swarm optimiser with breeding and subpopulations. In: GENETIC AND EVOLUTIONARY COMPUTATION CONFERENCE, 3, San Francisco, 2001. *Anais...* San Francisco, p. 469-476.
- MCCULLOCH, W.S.; PITTS, W. 1943. A logical calculus of the ideas imminent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115-133.
- OLIVEIRA, H.C.B.; VASCONCELOS, G.C.; MESQUITA, R.V.; SOUZA, M.M. 2006. Parametrical adjusting of a hybrid system for the vehicle routing problem with time windows. In: HIS'06 - HYBRID INTELLIGENT SYSTEMS, 6, Auckland, 2006. *Anais...* Auckland, IEEE Computer Society, p. 30.
- PRECHTEL, L. 1994. Proben1 – a set of neural network benchmark problems and benchmark rules. In: TECHNICAL REPORT 21/94, Fakultät für Informatik, Universität Karlsruhe, Germany.
- ROSENBLATT, F. 1962. *Principles of Neurodynamics: Perceptrons and the theory of brain mechanisms*. New York, Spartan Books, 616 p.
- SHI, Y.; EBERHART, R.C. 1998. *Parameter selection in particle swarm optimization*. Heidelberg, Springer Berlin, 591 p.
- SHI, Y.; EBERHART, R.C. 1999. Empirical study of particle swarm optimization. In: CONGRESS OF EVOLUTIONARY COMPUTATION, Washington, DC, 1999. *Anais...* IEEE Press, 3:1945-1950.
- SUGANTHAN, P.N. 1999. Particle swarm optimiser with neighbourhood operator. In: CONGRESS OF EVOLUTIONARY COMPUTATION, Washington, DC, 1999. *Anais...* IEEE Press, 3:1958-1962.
- WOLBERG, W.H.; MANGASARIAN, O.L. 1990. Multisurface method of pattern separation for medical diagnosis applied to breast cytology. *Proceedings of the National Academy of Sciences*, 87:9193-9196.
- YANG, X.; YUAN, J.; YUAN, J.; MAO, H. 2007. A modified particle swarm optimizer with dynamic adaptation. *Applied Mathematics and Computation*, 189(2):1205-1213.

Submitted on November 9, 2009.

Accepted on January 20, 2010.