

Uma Análise da Otimização por Enxame de Partículas e suas Aplicações

Eduardo Max A. Amaral¹

¹Programa de Pós-Graduação em Informática – Disciplina Otimização Combinatória e Metaheurísticas – Professora D.Sc Maria Cristina Rangel – Universidade Federal do Espírito Santo (UFES) – Vitória, ES – Brazil

eduardomax@ifes.edu.br

Abstract. *This paper aims to present an overview of the key concepts relating with metaheuristic Particle Swarm Optimization (PSO). Are analyzed some of its variants and, finally, some applications are described.*

Resumo. *Este trabalho tem por objetivo apresentar uma visão geral dos principais conceitos relativos a metaheurística Otimização por Enxame de Partículas (PSO). São analisadas algumas de suas variantes e, por fim, algumas aplicações são descritas.*

1. Introdução

Otimização é um problema matemático com muitas aplicações no “mundo real”. Consiste em encontrar os mínimos ou máximos de uma função de várias variáveis, com valores dentro de uma determinada região do espaço multi-dimensional. Os responsáveis pela tomada de decisões nos mais variados campos da atividade humana defrontam-se, cotidianamente, com esse tipo de necessidade. Às vezes, a índole do problema, a demanda de resultados precisos, ou a própria curiosidade, leva a formalizar variáveis, restrições e objetivos, de maneira que a natureza matemática do problema emerge. Esse é o processo de modelagem, que descobre isomorfismos entre a realidade empírica e o idealismo dos objetos matemáticos. No entanto, a correspondência entre experiência e modelo formal está longe de ser perfeita: a tradução está sujeita a erros, simplificações e falhas de comunicação. Notavelmente, a problemática de adequar um modelo matemático a uma situação real também pode ser formulada como um problema matemático, quase sempre de otimização [Martínez and Santos 1998].

O modelo matemático básico de um problema de otimização obedece ao seguinte raciocínio: minimizar uma função-objetivo $f(x)$, de acordo com as restrições impostas pela função $g(s)$, onde s representa o conjunto de soluções permitidas para a função em questão e x é variável que deve ser otimizada.

Alguns destes problemas envolvem apenas modelos lineares, nos quais as variáveis são contínuas e apresentam comportamento linear, tanto em relação à função objetivo quanto às restrições. Os outros problemas são conhecidos como problemas de otimização não-lineares, cujas características são as de exibir qualquer tipo de não-linearidade ou não continuidade, seja na função objetivo ou em qualquer das suas restrições [Greig 1980]. Define-se que a solução ótima encontrada pode ser de natureza

global ou local. É dita global a melhor solução dentre todas as soluções existentes para aquele determinado problema.

Não-linearidades e interações complexas entre variáveis de um problema formam um espaço de busca (S) que pode conter várias soluções, como no caso de um problema de minimização. Por causa da possibilidade de encontrar mínimos locais ou soluções subótimas, os métodos baseados em gradiente podem não ser bons candidatos como algoritmos de otimização eficientes quando aplicados a uma ampla gama de projetos de problemas.

Por outro lado, muitos problemas também não podem ser tratados através de métodos analíticos, seja por causa da dificuldade de formulação da modelagem ou do esforço matemático exigido na solução, principalmente quando estão envolvidas funções não-diferenciáveis ou descontínuas [Serapiao 2009].

Neste sentido, os métodos de otimização e busca estocástica baseados nos princípios e modelos da evolução biológica natural têm recebido crescente interesse nas últimas décadas, devido principalmente a sua versatilidade para a resolução de problemas complexos, nas áreas de otimização e aprendizado de máquina. O desenvolvimento de modelos computacionais, inspirados nos mecanismos evolutivos, caracteriza-se pela configuração de algoritmos de otimização robustos e sistemas adaptativos.

Os Algoritmos Evolucionários (AEs) [Fogel, Owens and Walsh 1966], [Schwefel 1995], metodologias da área computação evolucionária ou evolutiva (CE), não são algoritmos computacionais em seu significado usual, mas formam uma classe de métodos regidos por princípios similares. Estes princípios oriundos do “mundo biológico” são baseados na teoria da evolução Darwiniana e tentam abstrair e imitar alguns dos mecanismos evolutivos à resolução de problemas que requerem adaptação, busca e otimização [Oliveira, Silva and Aloise 2004].

Otimização por Nuvem de Partículas (*Particle Swarm Optimization - PSO*) [Kennedy and Eberhart 1995] é uma técnica de computação evolucionária baseada em dinâmica de populações. Foi desenvolvida por James Kennedy, um psicólogo social, e por Russell Eberhart, um engenheiro elétrico, em 1995, inspirada na simulação de um sistema social simplificado. A intenção original era simular graficamente o comportamento de um bando de pássaros em vôo com seu movimento localmente aleatório, mas globalmente determinado. Este método consiste na otimização de uma função objetivo através da troca de informações entre indivíduos (partículas) de uma população (enxame).

Este trabalho, através de um estudo bibliográfico de publicações em periódicos, congressos e na literatura em geral, tem por objetivo apresentar uma visão geral dos principais conceitos relativos a metaheurística Otimização por Enxame de Partículas (PSO), um dos algoritmos mais importantes das técnicas de Inteligência de Enxames. Serão também analisadas algumas de suas variantes e aplicações.

2. Histórico e evolução

O algoritmo de otimização por enxame de partículas (PSO) foi introduzido por James Kennedy e Russell Elberhart em 1995 para tratar problemas no domínio contínuo. O PSO emergiu de experiências com algoritmos que modelam o “comportamento social”

observado em muitas espécies de pássaros e cardumes de peixes, e até mesmo do comportamento social humano.

Kennedy e Eberhart inspiraram-se nos trabalhos de [Heppner and Grenander 1990] e [Wilson 1975], nos quais defendiam que os membros individuais de um bando podem-se beneficiar com as descobertas e experiências de cada um dos membros durante a busca por alimentos. Estas afirmações sugerem que o compartilhamento social de informações pode oferecer uma vantagem evolucionária: esta hipótese foi fundamental para o desenvolvimento do método de otimização por enxame de partículas.

Uma teoria sócio-cognitiva muito simples está por trás da PSO. Cada indivíduo de uma população possui sua própria experiência e é capaz de estimar a qualidade dessa experiência. Como os indivíduos são sociais, eles também possuem conhecimentos sobre como seus vizinhos comportam-se. Esses dois tipos de informação correspondem à aprendizagem individual (cognitiva) e à transmissão cultural (social), respectivamente. Portanto, a probabilidade de que um determinado indivíduo tome uma certa decisão será uma função de seu desempenho no passado e do desempenho de alguns de seus vizinhos [Serapiao 2009].

[Kennedy, Eberhart and Shi 2001] utilizaram três princípios para resumir o processo de adaptação cultural:

- Avaliar – os indivíduos possuem a capacidade de sentir o ambiente de forma a estimar seu próprio comportamento;
- Comparar – os indivíduos usam uns aos outros como referência comparativa;
- Imitar – a imitação é central em organizações sociais humanas e é importante para a aquisição e manutenção das habilidades mentais.

Trazendo esta idéia para o contexto dos pássaros ou dos peixes, percebe-se que dentro da sua vida em grupos, eles sofrem duas grandes influências:

- a) uma **influência social**: os indivíduos do grupo são diretamente influenciados pelos “melhores” indivíduos do grupo;
- b) uma **influência individual**: cada indivíduo é influenciado diretamente pela suas experiências positivas.

Sendo assim, o algoritmo de otimização por enxame de partículas é um tipo de inteligência de enxame inspirado no comportamento de bandos de pássaros. A busca por alimentos e a interação entre aves ao longo do vôo são modeladas como um mecanismo de otimização.

Fazendo uma analogia, o termo **partícula** é adotado para simbolizar os pássaros e representar as possíveis soluções do problema a ser resolvido. A área sobrevoada pelos pássaros é equivalente ao **espaço de busca** e encontrar o local com comida, ou o ninho, corresponde a encontrar a **solução ótima**. Para que o bando de pássaros sempre se aproxime do objetivo, ao invés de se perder ou nunca alcançar o alvo focado, utiliza-se o indicador denominado **fitness**, função que irá avaliar o desempenho das partículas. Para alcançar o alvo focado, sejam os alimentos ou os ninhos, os pássaros fazem uso de suas experiências e da experiência do próprio bando. O termo indicador da experiência ou conhecimento individual de cada partícula, isto é, seu histórico de vida, é o **pbest**. Em

uma abordagem mais simples, o responsável por representar o conhecimento do enxame como um todo é o **gbest**. A (Tabela 1) apresenta de forma resumida as nomenclaturas descritas acima:

Tabela 1. Identificação dos termos do PSO.

Termo	Significado
Partícula	Pássaro
Enxame	Bando de pássaros
Espaço de Busca	Área sobrevoada pelos pássaros
Posição	Localização de cada pássaro durante o voo
Solução ótima	Localização do pássaro onde ele encontrou o alimento ou o ninho
Fitness	Função de avaliação
pbest	Melhor posição conhecida pelo pássaro (Experiência individual)
gbest	Melhor posição conhecida pelo enxame (Experiência coletiva)

3. Algoritmo PSO

No algoritmo PSO, os indivíduos da população são representados por pontos, denominados de partículas, que voam em um espaço de busca \mathbb{R}^d , onde d é a dimensão do espaço. As variações nos atributos desses pontos levam a novos pontos no espaço, ou seja, correspondem a movimentações no espaço. Uma ideia inspirada em sistemas cognitivos é a de que essas partículas tenderão a mover-se em direção umas às outras e irão influenciar umas às outras.

A maior parte dos algoritmos de PSO empregam dois princípios sócio-métricos, que representam dois tipos de informação importante no processo de decisão. O primeiro princípio (g_{Best}) conecta conceitualmente todos os membros de uma população entre si. Como consequência, o comportamento de cada partícula é influenciado pelo comportamento de todas as outras partículas. A segunda métrica (p_{Best}) cria uma vizinhança para cada indivíduo composta por ele próprio e seus vizinhos mais próximos. Ambas as métricas são medidas por uma função de avaliação ($f(p)$), também chamada função objetivo ou de aptidão (*fitness*), que corresponde à optimalidade da solução do problema.

Uma partícula p_i irá se mover em uma determinada direção que é função da posição atual da partícula $x_i(t)$, de uma velocidade $v_i(t+1)$, da posição da partícula que levou ao seu melhor desempenho até o momento (p_{Best}), e do melhor desempenho global do sistema até o momento (g_{Best}). A velocidade da partícula será dada por (1):

$$v_i(t+1) = v_i(t) + \varphi_1 * rand_1(.) * (p_B - x_i(t)) + \varphi_2 * rand_2(.) * (g_B - x_i(t)) \quad (1)$$

onde: φ_1 e φ_2 são constantes limitadas a um intervalo finito, em que Kennedy denomina-os como sendo respectivamente os componentes “cognitivo” (coeficiente de individualidade) e “social” (coeficiente de sociabilidade). $rand_1(.)$ e $rand_2(.)$ são duas funções aleatória no intervalo [0,1].

Uma vez que a velocidade da partícula é calculada, a posição da partícula i na próxima iteração é estabelecida como uma influência aditiva da posição antiga e da velocidade calculada, sendo expressa por (2):

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

Para limitar a velocidade de uma partícula para que o sistema não extrapole o espaço de busca, são impostos limites (v_{max}) para seus valores em cada dimensão (d) do espaço de busca:

Se $v_i > v_{max}$ então $v_i = v_{max}$,
Senão se $v_i < -v_{max}$ então $v_i = -v_{max}$.

O algoritmo básico de otimização por enxame de partículas pode ser descrito brevemente utilizando os seguintes passos: dada uma população inicial de partículas, atualiza-se o vetor posição a partir do vetor velocidade de cada partícula até que se atinja o critério de parada pré-definido.

A (Figura 1), mostrada a seguir, ilustra a movimentação de uma partícula de sua posição inicial (PI) para sua posição final (PF) por influência do fator de individualidade (FI), do fator de sociabilidade (FS) e da velocidade inercial (VI), de acordo com o que foi apresentado nas equações (1) e (2).

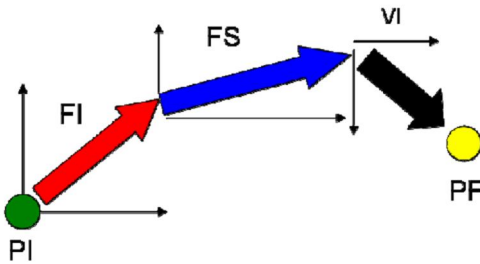


Figura 1. Movimentação de partícula de sua posição inicial para sua posição final

O pseudocódigo do algoritmo, em sua forma original, é descrito no Algoritmo 1.

Algoritmo 1: Pseudocódigo do PSO.

1. Determine o número de partículas P da população.
2. Inicialize aleatoriamente a posição inicial (x) de cada partícula p de P .

3. Atribua uma velocidade inicial (v) igual para todas as partículas.
 4. Para cada partícula p em P faça:
 - (a) Calcule sua aptidão $f_p = f(p)$.
 - (b) Calcule e melhor posição da partícula p até o momento (p_{Best}).
 5. Descubra a partícula com a melhor aptidão de toda a população (g_{Best}).
 6. Para cada partícula p em P faça:
 - (a) Atualize a velocidade da partícula pela fórmula:

$$v_i(t+1) = v_i(t) + \varphi_1 * rand_1(.) * (p_B - x_i(t)) + \varphi_2 * rand_2(g_B - x_i(t))$$
 - (b) Atualize a posição da partícula pela fórmula:

$$x_i(t+1) = x_i(t) + v_i(t+1)$$
 7. Se condição de término não for alcançada, retorne ao passo 4.
- A (Figura 2) representa um exemplo de convergência do algoritmo PSO.

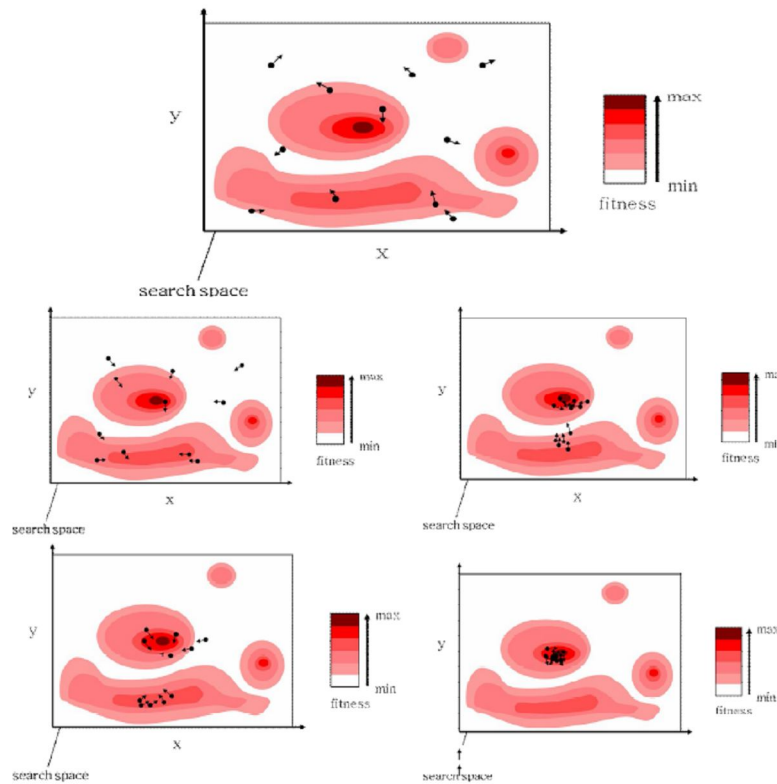


Figura 2. Exemplo de convergência do algoritmo PSO

4. Variações e avanços do algoritmo PSO

Para evitar os fenômenos de convergência prematura ou divergência que ocorriam com frequência no processo de otimização com o uso de PSO (especialmente em problemas

complexos multidimensionais) e também para melhorar a velocidade de convergência e a precisão do PSO muitos tipos de esquemas foram introduzidos para realçar o PSO. Alguns dos esquemas mais representativos são: peso de inércia da velocidade (W) [Shi and Eberhart, 1998]; fator de constrição (K), que um fator de amortecimento baseado nos parâmetros cognitivo (φ_1) e social (φ_2) para limitar a velocidade da partícula [Clerc 1999] e [Eberhart & Shi, 2000].

4.1. Peso inercial (W)

Em [Shi and Eberhart, 1998] é introduzido um novo parâmetro no algoritmo PSO, denominado peso inercial (W), com o objetivo de melhorar a sua taxa de convergência. Tal parâmetro tem por objetivo equilibrar a exploração global e local. Valores superiores à unidade irão resultar em uma maior exploração global, ao passo que valores muito menores que a unidade, numa maior exploração local. A partir da Equação (1) obtém-se a nova equação de atualização da velocidade:

$$v_i(t+1) = W * v_i(t) + \varphi_1 * rand_1(.) * (p_B - x_i(t)) + \varphi_2 * rand_2(g_B - x_i(t)) \quad (3)$$

Em [Shi and Eberhart 1998] é proposto que o peso inercial inicie com um valor relativamente alto e vá decrescendo a cada iteração do algoritmo. Desta forma, é possível encontrar soluções mais refinadas com um menor número de iterações. Os autores realizaram experimentos com o peso inercial variando linearmente entre 0,9 e 0,4, obtendo bons resultados.

4.2. Fator de constrição

Segundo Clerc e Kennedy (2002), a inclusão de um fator de constrição (K) ajuda a assegurar a convergência do algoritmo, controlando a magnitude das velocidades. O fator K desempenha um papel semelhante à velocidade máxima (v_{max}), limitando a velocidade da partícula, visto que um valor muito alto pode fazer com que a partícula ultrapasse uma posição ótima. A equação da atualização usando este coeficiente é a seguinte:

$$v_i(t+1) = K * [W * v_i(t) + \varphi_1 * rand_1(.) * (p_B - x_i(t)) + \varphi_2 * rand_2(g_B - x_i(t))] \quad (4)$$

sendo:

$$K = \frac{2}{|2 - \alpha - \sqrt{\alpha^2 - 4\alpha}|} \quad (5)$$

onde $\alpha = \varphi_1 + \varphi_2$, com $\alpha > 4$.

5. Topologias e troca de informação

A troca e disseminação de informação entre as partículas do enxame é o principal componente de qualquer enxame inteligente baseado no algoritmo de otimização por enxame de partículas. O PSO, assim com os outros algoritmos de otimização por enxame tem mecanismos próprios para disseminar as melhores posições encontradas durante a execução do algoritmo [Engelbrecht 2005]. Um mecanismo utilizado pelo enxame para distribuir essas informações é denominado de topologia. Esta estrutura rege a organização das partículas no enxame e como as mesmas se comunicam. Logo, variações sobre essa estrutura podem influenciar no desempenho do algoritmo.

A escolha da topologia influencia na avaliação da velocidade das partículas. A depender de como as partículas se comunicam entre si e do problema a ser tratado, a busca pela solução ótima pode priorizar tanto a velocidade de convergência, a qualidade da solução ou ambas. As principais topologias utilizadas como mecanismos de comunicação entre as partículas são: a topologia global e a topologia local. A (Figura 3) apresenta a estrutura de tais topologias.

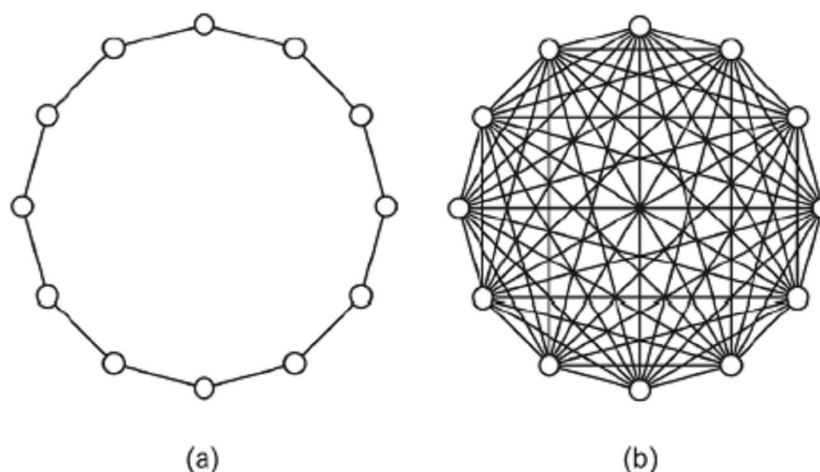


Figure 3. (a) Topologia Local e (b) Topologia Global

Na topologia local, conforme ilustrado na (Figura 4.a), o enxame está organizado em formato de anel e cada partícula deste possui dois vizinhos. Logo, a partícula troca informações apenas com seus vizinhos diretos.

Esse mecanismo de troca de informações baseado na vizinhança local também é chamado de l_{Best} e apresenta vantagens em relação à topologia global [Engelbrecht 2005]. Devido à distância lógica existente entre as partículas, a topologia local mostra-se bastante eficiente em não convergir rapidamente para mínimos locais. Com isso, diferentes regiões do espaço de busca podem ser exploradas ao mesmo tempo, sobre o qual as partículas procuram se mover em direção à melhor solução encontrada pelos seus vizinhos.

Na topologia global, conforme ilustrado na (Figura 4.b), o enxame está organizado em formato estrela e todas as partículas estão conectadas entre si. Esta

topologia utiliza o mecanismo de vizinhança global, também denominado de g_{Best} para a troca de informação.

Ao contrário da topologia anteriormente descrita, esta topologia permite uma convergência mais acelerada, visto que a informação da melhor posição é disseminada rapidamente entre todas as partículas do enxame. A comunicação é análoga a uma imensa comunidade onde todas as decisões tomadas são rapidamente conhecidas por todos os membros. Entretanto, o risco do algoritmo convergir a um mínimo local, não explorando devidamente todo o espaço de busca, é grande. Esta estrutura provê maior rapidez na busca da melhor solução para problemas multimodais, porém não garante a qualidade da solução obtida. Nestes casos o algoritmo pode atingir um mínimo local, devido a sua convergência precoce.

Diversos aspectos de diversas topologias já foram analisados em vários trabalhos originando novas propostas na modificação do comportamento do enxame. O foco dessas modificações demonstra que a melhora de desempenho no algoritmo está relacionada diretamente em como as partículas se interagem dinamicamente dentro do enxame.

6. Otimização Discreta por Enxame de Partículas

O PSO foi originalmente desenvolvido para ser utilizado na solução de problemas de natureza contínua. Para tornar possível sua utilização em problemas discretos foi preciso realizar algumas adaptações nos algoritmos e na representação das partículas e suas velocidades.

Tais adaptações preservaram a estrutura do algoritmo do PSO contínuo. As equações de atualização de posição e velocidade permaneceram inalteradas. Inicialmente, as operações realizadas para atualização da velocidade e da posição das partículas no plano discreto foram as propostas em [Clerc 2000]. Essas adaptações transformam cada componente das equações (antes simplesmente operações matemáticas) em operações discretas.

As adaptações realizadas são:

- 1) **Posição da partícula:** a posição da partícula passou a ser representada como um vetor (ou lista) com valores discretos. No PSO contínuo a partícula era representada por um vetor de valores contínuos. Por exemplo, no caso do Problema do Caixeiro Viajante, os valores discretos representariam as cidades da solução e o vetor, como um todo, uma solução completa;
- 2) **Velocidade da partícula:** a velocidade passou a ser vista como uma lista de transposições (swaps) realizadas no vetor posição das partículas, enquanto que no PSO contínuo era caracterizada simplesmente por um valor aritmético para cada uma das N dimensões do espaço de busca;
- 3) **Aplicação da velocidade a uma posição:** no PSO contínuo os valores armazenados no vetor posição das partículas eram somados algebricamente aos valores do vetor velocidade. No PSO discreto a aplicação da velocidade a uma posição é realizada a partir da troca dos valores do vetor posição de acordo com os valores contidos no vetor velocidade.

Podemos ilustrar tal procedimento com um exemplo. Dada uma posição $P1 = \{2,4,1,3,5\}$ e uma velocidade $V1 = \{(1,4),(3,5)\}$ ao aplicarmos $V1$ à posição $P1$ obteremos uma posição $P2 = \{3,4,5,2,1\}$;

4) **Adição entre duas velocidades:** passou a ser realizada simplesmente pela concatenação das listas de transposição de cada uma das velocidades, enquanto que no PSO contínuo era realizada pela soma algébrica dos componentes de velocidade correspondentes (de mesmas dimensões).

Podemos ilustrar tal procedimento com um exemplo. Dada uma velocidade $V1 = \{(2,3)\}$ e outra $V2 = \{(3,4)\}$, $V3 = V1 + V2 = \{(2,3),(3,4)\}$ será a velocidade resultante da soma entre $V1$ e $V2$;

5) **Multiplicação de coeficientes por uma velocidade:** Nessa operação existem quatro possibilidades para determinar o resultado:

- Se $C < 0$, $C*V$ não é definido;
- Se $C = 0$, $C*V = \text{nulo}$;
- Se C pertence $(0,1]$, truncamos $C*|V|$;
- Se $C > 1$, aplica-se a velocidade V , C vezes na partícula, mais a parte decimal $C*|V|$;

Podemos ilustrar tal procedimento com um exemplo. Dado o coeficiente $C = 2,5$ e a velocidade $V = \{(1,2),(3,4)\}$, então $C*V = \{(1,2),(3,4),(1,2),(3,4),(1,2)\}$;

6) **Subtração de posições:** Ao se subtrair duas posições quaisquer deve-se obter uma velocidade correspondente. Dada duas posições A e B , obtém-se a velocidade através de $V = A - B$ onde V será a lista de transposições necessárias para ir da posição B até a posição A .

Podemos ilustrar tal procedimento com um exemplo. Dadas as posições $A = \{1,2,3,4,5\}$ e $B = \{2,3,1,5,4\}$. Os passos a seguir mostram como essa velocidade foi obtida a partir da subtração das posições A e B ;

- a) $B' = B + \{(1,3)\} = \{1,3,2,5,4\}$
- b) $B'' = B' + \{(2,3)\} = \{1,2,3,5,4\}$
- c) $B''' = A = B'' + \{(4,5)\} = \{1,2,3,4,5\}$

Portanto, $A - B$ é a velocidade $V = \{(1,3),(2,3),(4,5)\}$.

7. Estado da Arte

Após a publicação do artigo pioneiro *Particle Swarm Optimization* [Kennedy and Eberhart 1995], as aplicações da técnica de PSO têm sido amplamente investigadas na literatura. Existem vários artigos propondo soluções ou melhorias utilizando PSO em diversas áreas do conhecimento, como em engenharia da produção [Toledo et. al 2010] e robótica [Ayala and Coelho 2007] e seria impossível listar todos aqui. Assim, são listados apenas 2 artigos propondo melhorias em suas respectivas áreas utilizando PSO.

7.1. Otimização aplicada ao problema geral de dimensionamento de lotes e programação da produção

O trabalho descrito em [Toledo et al 2010], propõe um método baseado em Otimização por Enxame de Partículas (OEP) associado à programação matemática para resolução do Problema Geral de Dimensionamento de Lotes e Programação da Produção (PGDLPP). A OEP fixa as variáveis binárias do modelo enquanto um método exato é utilizado na determinação das variáveis contínuas. O PGDLPP estudado também considera penalização para demandas não atendidas, onde uma formulação matemática é apresentada para essa variante do problema. Instâncias são solucionadas por uma ferramenta de modelagem matemática cujas soluções servem para avaliação do desempenho da OEP proposta. A OEP também é comparada a um AG anteriormente utilizado para resolver o mesmo problema.

O problema é modelado dividindo o horizonte de planejamento em T macro-períodos. Por sua vez, cada macro-período t possui um número fixo de micro-períodos S . Um aspecto neste tipo de modelagem é que o tamanho de cada micro-período s varia de forma proporcional ao tamanho do lote do produto a ele atribuído. Assim, as variáveis de dimensionamento dos lotes e de atribuição de produtos às linhas e períodos estão indexadas por produtos e micro-períodos. O modelo assume que um único produto é atribuído e produzido em cada micro-período.

A (Figura 4) apresenta um exemplo dessa situação, supondo três produtos e três micro-períodos em cada macro-período em uma linha $L1$. O tamanho dos micro-períodos $s1, s2, s3, s4, s5$ e $s6$ varia de forma proporcional à quantidade produzida. Por exemplo, o lote em $s2$ de $P2$ é maior que o lote de $P2$ em $s4$ na (Figura 5). Assim, os micro-períodos $s1, s4$ e $s6$ ocupam 1 unidade de tempo, $s2$ ocupa 2 unidades e $s5$ ocupa 3 unidades de tempo.

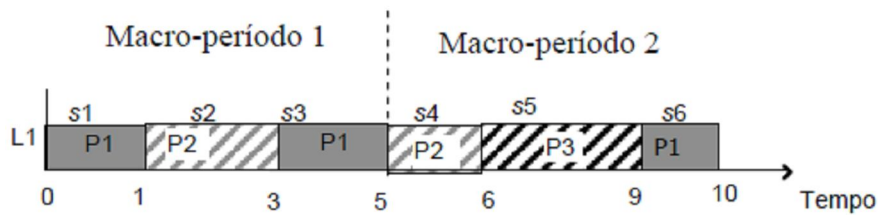


Figure 4. Variação dos micro-períodos [Toledo et al, 2010]

Um total de J produtos e L linhas são considerados. Abaixo são listados os demais parâmetros do problema:

- Ct : Capacidade em unidades de tempo disponível no macro-período t .
- TPl,j : Tempo de processamento do produto j na linha l .
- $Minj$: Lote mínimo do produto j .
- Hj : Custo de estoque do produto j .
- $CTij$: Custo de troca do produto i para o produto j .
- Dit : Demanda do produto i no macro-período t .

- I_j^0 : Estoque inicial do produto j no início do horizonte de tempo.
- $Y_{l,j,0}$: 1, se o produto j está ajustado inicialmente para a linha l ; 0 caso contrário.
- M : Penalização por unidade de demanda não atendida.
- $I_{jt} \geq 0$: Estoque do produto j ao final do macro-período t .
- $q_{ljs} \geq 0$: Quantidade do produto j produzido no micro-período s da linha l .
- $q_{0j} \geq 0$: Quantidade de demanda do produto j que não foi produzida
- $y_{l,j,s}$: 1 Se o produto j é atribuído à linha l no micro-período s ; 0 caso contrário.
- $z_{lijs} \geq 0$: $z_{lijs} = 1$ se há troca do produto i para j no micro-período s da linha l ; $z_{ijs} = 0$, caso contrário.

O modelo matemático para o PGLDPP com máquinas paralelas e penalização de demandas é descrito a seguir:

$$\text{Minimize } \sum_{j=1}^J \sum_{t=1}^T H_j I_{jt} + \sum_{i=1}^I \sum_{j=1}^J \sum_{l=1}^L \sum_{s=1}^{T \cdot S} CT_{ij} Z_{lijs} + M \sum_{j=1}^J q_j^0 \quad (6)$$

A função objetivo minimiza os custos de estoque e os custos de troca envolvendo produtos.

Os resultados obtidos pela OEP são comparados aos retornados pelo AMPL/CPLEX (método exato) e pelo Algoritmo Genético (AG).

A (Figura 5) apresenta os desvios total médio obtidos por cada método considerando os conjuntos de instâncias. Observe que OEP1 e OEP2 apresentam praticamente o mesmo desvio em todos os conjuntos e superam o desempenho médio do AG.

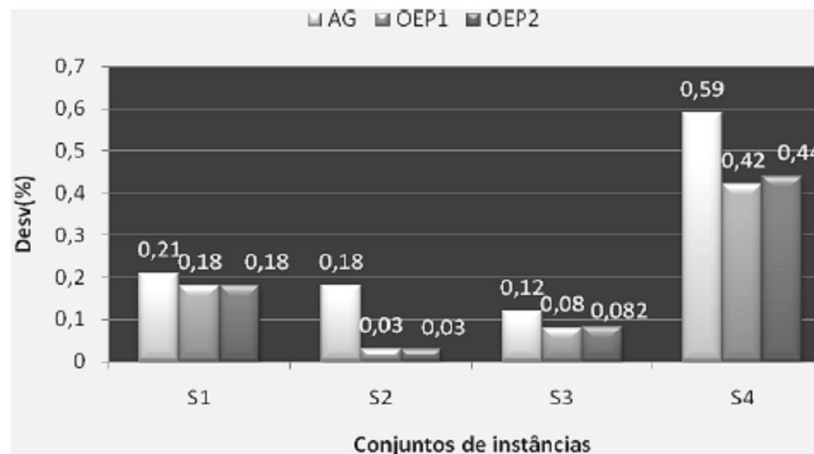


Figure 5. Desvio médio em S1, S2, S3 e S4 [Toledo et al, 2010]

O AMPL/CPLEX não foi capaz de retornar soluções ótimas ao final de 1 hora de execução em P1, P2 e P3. Por outro lado, as metaheurísticas propostas foram executadas por 0,5h e superaram os valores das soluções finais obtidas pelo AMPL/CPLEX na

maioria das instâncias. Os resultados revelam o melhor desempenho obtido pela OEP proposta.

7.2. Otimização aplicada à Robótica de Manipuladores

Em [Ayala and Coelho 2007] é apresentado um estudo sobre a otimização dos ganhos de controladores PD (Proporcional-Derivativo) e PID (Proporcional-Integral-Derivativo) visando o controle de um manipulador robótico com dois graus de liberdade utilizando o método PSO. O objetivo da otimização do algoritmo PSO é minimizar o erro das variáveis de posição e velocidade e também o esforço de controle.

As equações dinâmicas do robô constituem um conjunto de equações diferenciais acopladas contendo termos, tais como: inércia variante, torque centrífugo e Coriolis, termos de carregamento e gravidade. O movimento do efetuador final em uma particular trajetória com particular velocidade requer um conjunto complexo de funções de torque para serem aplicadas a atuadores na(s) junta(s) do manipulador robótico.

O modelo dinâmico para o manipulador robótico de dois graus de liberdade utilizado, conforme apresentado na (Figura 6), é descrito abaixo:

$$\begin{aligned} \tau_1 = & m_2 l_2^2 (\ddot{\theta}_1 + \ddot{\theta}_2) + m_2 l_1 l_2 c_2 (2\ddot{\theta}_1 + \ddot{\theta}_2) + \\ & (m_1 + m_2) l_1^2 \ddot{\theta}_1 - m_2 l_1 l_2 s_2 \dot{\theta}_2^2 \\ & - 2m_2 l_1 l_2 s_2 \dot{\theta}_1 \dot{\theta}_2 + m_2 l_2 g c_{12} + (m_1 + m_2) l_1 g c_1 \end{aligned} \quad (7)$$

$$\begin{aligned} \tau_2 = & m_2 l_1 l_2 c_2 \ddot{\theta}_1 + m_2 l_1 l_2 s_2 \dot{\theta}_1^2 + \\ & m_2 l_1 g c_{12} + m_2 l_2^2 (\ddot{\theta}_1 + \ddot{\theta}_2) \end{aligned} \quad (8)$$

onde $s_{12} = \sin(\theta_1)$, $s_2 = \sin(\theta_2)$, $c_1 = \cos(\theta_1)$, $c_2 = \cos(\theta_2)$, e $c_{12} = \cos(\theta_1 + \theta_2)$. Os subscritos 1 e 2 denotam os parâmetros das juntas 1 e 2, respectivamente. Os parâmetros adotados nas simulações são comprimentos $l_1 = 0,8$ m e $l_2 = 0,4$ m, massas $m_1 = m_2 = 0,1$ kg, e aceleração da gravidade de $g = 9,81$ m/s². As restrições impostas aos torques t_1 e t_2 são $[-1000; 1000]$ Nm. Um período de amostragem, T_s , de 10 ms é adotado neste trabalho.

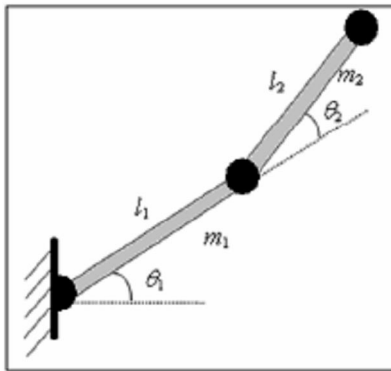


Figure 6. Manipulador robótico de dois graus de liberdade [Ayala and Coelho 2007]

A função objetivo f utilizada no PSO que visa considerar os ganhos dos controladores que melhor minimizem os erros de posição e velocidade do elo 1 e 2 e

também o esforço de controle (torque), é dada pela seguinte equação ponderada a ser minimizada:

$$f = \sum_{t=1}^{tf} [d_1(t)]^2 + 10^{-25} \sum_{t=1}^{tf} [v_1(t)]^2 + 0,01 \sum_{t=1}^{tf} |\tau_1(t)| + \sum_{t=1}^{tf} [d_2(t)]^2 + 10^{-25} \sum_{t=1}^{tf} [v_2(t)]^2 + 0,01 \sum_{t=1}^{tf} |\tau_2(t)| \quad (9)$$

Na (Figura 7) é apresentado o melhor resultado, ou valor mínimo de f , para o controlador PD.

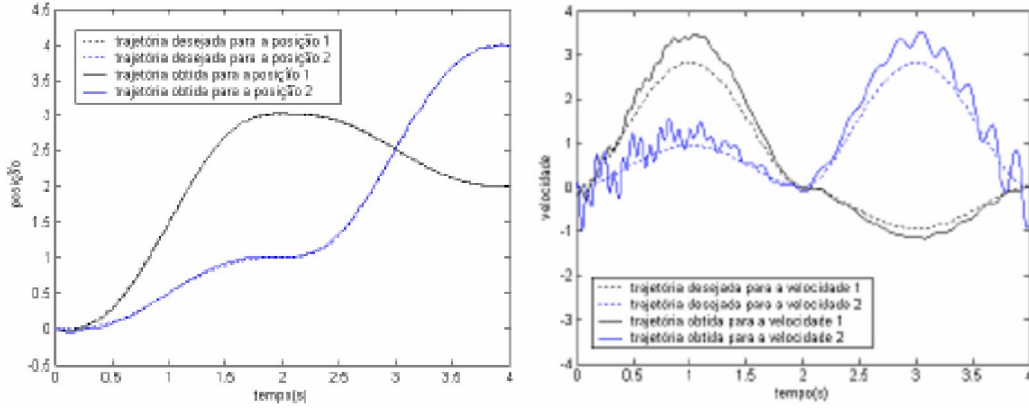


Figure 7. Valor mínimo de f para o controlador PD [Ayala and Coelho 2007]

Na (Figura 8) é apresentado o melhor resultado, ou valor mínimo de f , para o controlador PID.

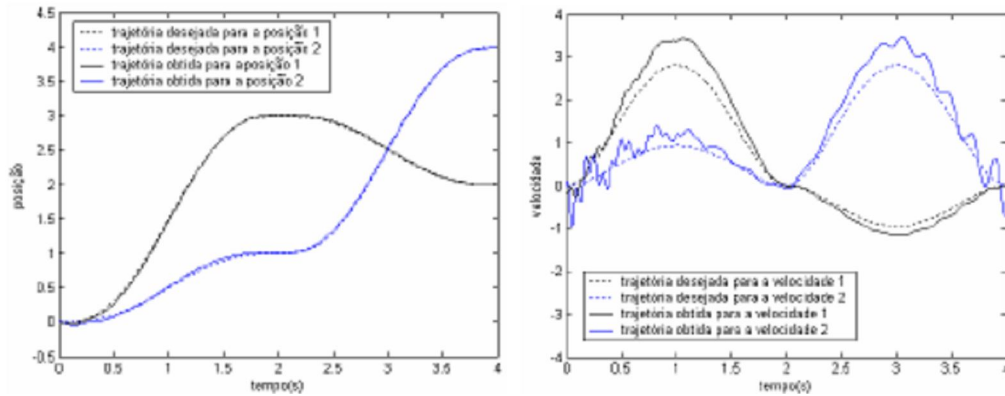


Figure 8. Valor mínimo de f para o controlador PID [Ayala and Coelho 2007]

Nota-se pelos resultados apresentados nas Figuras 7 e 8 que tanto para o melhor resultado de controle PD quanto para o PID houve uma boa aproximação no seguimento da trajetória desejada para a posição dos elos 1 e 2. De acordo com o resultado apresentados, a otimização baseada em PSO foi apropriada no seguimento da trajetória para a posição dos elos.

8. Considerações finais

Nos últimos anos as técnicas de inteligência de enxames têm experimentado um crescimento, tanto no tocante à multiplicidade de algoritmos em desenvolvimento quanto na variedade das aplicações resultantes. Dentre essas técnicas, PSO vem sendo aplicado em muitas pesquisas em diversas áreas do conhecimento, principalmente, em problemas contínuos não-lineares. O objetivo desse trabalho foi apresentar essa metaheurística de otimização, bem como, algumas de suas variações e aplicação em problemas de otimização combinatória.

Os algoritmos de enxame, de modo geral, têm mostrado resultados extremamente promissores em termos de robustez de descoberta de soluções, da velocidade da solução e precisão numérica. Grande parte desse êxito é também devido à simplicidade, generalidade e paralelismo desses algoritmos, sobretudo quando comparados à dificuldade de modelagem matemática de certos problemas [Serapiao 2009].

Por outro lado, esses métodos ainda enfrentam dificuldades na aplicação de problemas de otimização de alta dimensão, principalmente por causa da carga computacional. Contudo, com a interação de pesquisadores multi e interdisciplinares engajados no aprimoramento dos algoritmos e no estudo e desenvolvimento de novos modelos, o futuro das técnicas baseadas em colônias parece ser promissor.

9. Referências

- Ayala, H. V. H. and Coelho, L. S. (2007) “Otimização por Enxame de Partículas de Controle Multivariável em uma Aplicação de Robótica de Manipuladores”, In: *VIII SBAI, Simpósio Brasileiro de Automação Inteligente*, Florianópolis, SC.
- Clerc, M. (1999) “The swarm and the queen: towards a deterministic and adaptive particle swarm optimization”, *Proceedings of the 1999 Congress Evolutionary Computation*, IEEE Press, pp. 1951-1957.
- Clerc, M. (2000) “Discrete Particle Swarm Optimization Illustrated by the Traveling Salesman Problem”, <http://www.mauriceclerc.net>, February.
- Eberhart, R.C., Shi, Y. (2000) “Comparing inertia weights and constriction factors in particle swarm optimization”, *Proceedings of the 2000 Congress Evolutionary Computation*, IEEE Press, pp. 84–88.
- Engelbrecht, P. (2005) “Fundamentals of Computational Swarm Intelligence”, New York, NY: John Wiley & Sons.
- Fogel, L. J., Owens, A. J. and Walsh, M. J. (1966) “Artificial Intelligence Through Simulated Evolution”, New York, NY: John Wiley & Sons.
- Greig, D. M. (1980) “Optimization”, cap 3-4, Longman Inc., New York, USA.
- Heppner, F. and Grenander, U. (1990) “A stochastic nonlinear model for coordinated bird flocks”, In S. S. Krasner, Ed., *The Ubiquity of Chaos*. AAAS Publications, Washington, DC.
- Kennedy, J. and Eberhart, R.C. (1995) “Particle Swarm Optimization”, In *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948.

- Kennedy, J., Eberhart, R.C. and Shi, Y. (2001) "Swarm Intelligence", San Francisco: Morgan Kaufmann / Academic Press.
- Martínez, J. M. and Santos, S. A. (1998) "Métodos Computacionais de Otimização", Departamento de Matemática Aplicada, IMECC-UNICAMP, Campinas, SP.
- Oliveira, M. C. S. de, Silva, T. L. and Aloise, D. J. (2004) "Otimização por Nuvem de Partículas: Diferença entre Aplicações a Problemas Contínuos e Discretos", In XXXVI SBPO, São João Del-Rei, MG.
- Schwefel, H. P. (1995) "Evolution and Optimum Seeking" New York: John Wiley & Sons.
- Serapiao, A. B. de S. (2009) "Fundamentos de otimização por inteligência de enxames: uma visão geral", Sba Controle & Automação, vol.20, n.3, pp. 271-304, ISSN 0103-1759.
- Shi, Y., Eberhart, R.C. (1998) "A Modified Particle Swarm Optimizer", In: *IEEE World Congress on Computational Intelligence*, Anchorage, Alaska, pp.69-73.
- Toledo, C. F. M., Lafitá, T. F. Q., Arantes, M. S., Oliveira, R. R. R., França, P. M. (2010) "Otimização por Enxame de Partículas Aplicada ao Problema Geral de Dimensionamento de Lotes e Programação da Produção", In: *XXX Encontro Nacional de Engenharia de Produção*, São Carlos, SP, Brasil.
- Wilson, E.O. (1975) "Sociobiology: The new synthesis", Belknap Press, Cambridge, MA.