



Estado da Arte em Métodos Numéricos para Algoritmos Meta-heurísticos

Felipe Tozim Demiti¹ – felipetozim2@gmail.com
Cid Marcos Gonçalves Andrade¹ – cmgandrade@uem.br

1 – Universidade Estadual de Maringá – UEM Maringá

Resumo: Este trabalho tem o objetivo de realizar uma breve revisão na literatura sobre métodos numéricos para algoritmos meta-heurísticos, em especial AG, DE e PSO, para melhor compreender sua estrutura e dinâmica. Inicialmente é feita uma breve introdução sobre o tema, onde são apresentados alguns conceitos a cerca de algoritmos meta-heurísticos. Após, são expostos os formatos padrão (mais básicos ou originais) desses algoritmos. Depois são apresentados uma série de trabalhos que também abordam os algoritmos AG, DE e PSO, porém com modificações e combinações com outras técnicas, a fim de obter melhores resultados em relação à forma padrão. Ao final, conclui-se que os algoritmos modificados têm melhor desempenho que os tradicionais.

Palavras-chave: Estado da arte; Meta-heurística; Revisão.

1. Introdução

Métodos numéricos podem ser definidos como técnicas que utilizam operações aritméticas, repetidas vezes, para a resolução de problemas matemáticos (CHAPRA; CANALE, 2008). Eles são capazes de determinar soluções tão próximas dos valores verdadeiros quanto se queira.

Inserido nesse contexto estão os algoritmos meta-heurísticos. Ele são uma técnica de busca de alto nível, ou de uma forma mais completa, é uma estrutura generalizada de um algoritmo que pode ser utilizado para a resolução de diversos problemas de otimização, por meio de uma pequena quantidade de modificações que a ajustem para a análise de cada problema característico (ZAPELINI, 2009).

Dentre os algoritmos que atendem a essas características e assim podem ser considerados meta-heurísticos estão os Algoritmos

Evolutivos (AE). Algoritmos que em essência utilizam um método construtivo para a geração da população inicial (soluções iniciais), além de um processo específico de evolução e troca de informação entre os indivíduos, considerando também o uso de uma técnica de busca local com o intuito de aprimorar a solução da população (SERAPIÃO, 2009). Como exemplos de algoritmos que podem ser assim classificados estão: de evolução diferencial (*Differential Evolution* – DE) (MELO; FAMPA; RAUPP, 2010), otimização por enxame de partículas (*Particle Swarm Optimization* – PSO) (MENDONÇA et al., 2011), otimização por colônia de formigas (*Ant Colony Optimization* – ACO), algoritmos genéticos (*Genetic Algorithms* – GA), recozimento simulado (*Simulated Annealing* – SA), busca tabu (*Tabu Search* – TS) entre outros (SERAPIÃO, 2009).

O objetivo deste trabalho é realizar uma revisão na literatura, com o intuito de desenvolver uma análise da estrutura e da dinâmica dos algoritmos meta-heurísticos, já que estes podem ser utilizados como ferramenta para a sintonia de controladores. Este trabalho será limitado apenas a alguns algoritmos que fazem uso de GA, DE e PSO, tanto em suas formas mais básicas e/ou primitivas como em modelos mais recentes e modificados.

2. Estrutura e dinâmica básica dos algoritmos

2.1. GA

O GA é um algoritmo matemático que se baseia no princípio da seleção natural, proposto pelo naturalista britânico Charles Robert Darwin. Além do conceito de herança genética, que foi inicialmente desenvolvido no trabalho de Gregor Mendel em 1865. O GA foi concebido por John Holland na década de 1970 (HOLLAND, 1975).

No GA inicialmente é gerada uma população (geralmente de modo aleatório). Na sequência todos os indivíduos são avaliados por meio da função de avaliação, os melhores indivíduos são selecionados para o processo de reprodução, para formar uma nova geração, então nestes são aplicados as operações de cruzamento (*crossover*) e mutação (ESHELMAN, 1997). A cada geração, soluções bem avaliadas se reproduzem, enquanto que soluções mal avaliadas são eliminadas. As mutações garantem a variabilidade genética da população. Elas podem acontecer em pequena escala e produzirem uma modificação aleatória no material genético. Todo esse processo se repete até que o critério de parada estabelecido seja atendido.

Vale destacar que o algoritmo de Holland tem três características que o distingue de outros algoritmos evolutivos: a representação usada – bitstrings; o método de seleção – seleção proporcional; o método primário de produzir variação – cruzamento (ESHELMAN, 1997).

O método de seleção escolhe os indivíduos para o cruzamento probabilisticamente, atribuindo a cada um uma probabilidade proporcional ao seu desempenho observado. Assim, indivíduos melhores têm mais oportunidades de produzir descendentes (ESHELMAN, 1997). Com relação as taxas de mutação e cruzamento, se elas forem baixas o suficiente, é possível que alguns dos descendentes produzidos sejam idênticos aos seus pais (ESHELMAN, 1997).

2.2 DE

O algoritmo DE é um otimizador global que funciona de modo confiável em ambientes não-lineares e multimodais (COELHO; PESSÔA, 2011). Ele foi proposto por Storn e Price em 1995.

No DE, inicialmente é gerada uma população de N_p vetores com dimensão D , N_p não varia ao longo do processo (STORN; PRICE, 1997). Se uma solução prévia estiver disponível, a população inicial pode ser gerada somando-se à solução nominal, desvios aleatórios com distribuição normal (STORN; PRICE, 1997).

O DE produz novos vetores de parâmetros acrescentando a diferença ponderada entre dois vetores da população a um terceiro vetor, operação conhecida como mutação (STORN; PRICE, 1997). Os parâmetros do vetor que sofre mutação são então misturados com os parâmetros de outro vetor predeterminado, o vetor alvo, para produzir o chamado vetor de teste, processo denominado de *crossover*. Caso o vetor de teste gere um valor de função de custo menor que o vetor de destino, o vetor de teste substituirá o vetor de destino na geração seguinte – seleção. Cada vetor populacional deve servir uma vez como vetor alvo para que ocorram as competições entre os N_p vetores em uma geração (STORN; PRICE, 1997). Todo esse processo é repetido até que o número máximo de gerações seja atingido.

A operação de mutação é executada conforme (1). Nela $x_{i,G}$ com $i \in \{1, 2, \dots, N_p\}$ é um vetor alvo com índices aleatórios $r_1, r_2, r_3 \in \{1, 2, 3, \dots, N_p\}$, inteiros e mutuamente diferentes. G simboliza a geração e F é um fator real e constante $\in [0, 2]$ que controla a amplificação da diferença entre os vetores (STORN; PRICE, 1997).

$$v_{i,G+1} = x_{r_1,G} + F \cdot (x_{r_2,G} - x_{r_3,G}) \quad (1)$$

Outro operador de que o DE se utiliza é o operador cruzamento. Ele atua de modo a verificar se o vetor $v_{i,G+1}$ será aceito ou não na próxima geração. Isso é feito produzindo-se um número aleatório e comparando este número com a constante de cruzamento (CR), se ele for menor ou igual à CR , então ele é aceito (STORN; PRICE, 1997). Caso contrário o vetor $x_{i,G}$ permanece na população atual.

Com relação aos fatores N_p , F (conhecido como taxa de perturbação (SOUZA, 2011)) e CR (também denominado como probabilidade de cruzamento (SOUZA, 2011)) existem alguns valores que podem ser escolhidos para se obter bons resultados com o DE. O valor mínimo de N_p deve ser pelo menos 4, para que o algoritmo tenha um número suficiente de vetores diferentes para trabalhar (STORN; PRICE, 1997). Aconselha-se que N_p esteja entre 5 e 10 vezes o número de variáveis

do problema (STORN; PRICE, 1997). F com valor de 0,5 é considerada uma boa escolha inicial, porém valores menores que 0,4 e maiores que 1,0 são raramente efetivos (STORN; PRICE, 1997). Com relação ao CR , 0,1 é um valor inicial apropriado, entretanto 0,9 ou 1,0 também são interessantes para avaliar a possibilidade de uma solução rápida, já que grandes valores de CR normalmente aceleram o processo de busca (STORN; PRICE, 1997). Por fim, em situações em que a população converge prematuramente, recomenda-se aumentar o(s) valor(e)s de F e/ou N_p (STORN; PRICE, 1997).

2.3 PSO

O PSO foi proposto por James Kennedy e Russell Elberhart em 1995 (ANDRADE; COSTA; ANGÉLICO, 2013). Seu desenvolvimento foi baseado no “comportamento social” verificado em várias espécies de pássaros e de cardumes (SERAPIÃO, 2009).

Esta técnica, difere de outros algoritmos evolutivos por não utilizar operadores evolucionários em suas manipulações, tais como mutações e cruzamentos, além disso o conjunto de soluções é denominado de partículas (FALCUCCI, 2007). As partículas se movimentam ou “evoluem” baseadas em equações de posição e de velocidade.

Em outras palavras, no PSO cada partícula corresponde a um ponto do espaço de soluções do problema, que é avaliada a cada iteração de acordo com uma função objetivo. Toda partícula se desloca ao longo do espaço de busca, utilizando-se de comparações entre sua posição atual, sua melhor posição atingida até o momento ($pbest$) e a posição obtida pela melhor partícula do enxame ($gbest$) (SERAPIÃO, 2009).

O PSO é um algoritmo bem simples. Inicialmente deve-se inicializar uma matriz de partículas com velocidades e posições aleatórias nas D dimensões do problema (EBERHART; KENNEDY, 1995). Depois as partículas são avaliadas com a função objetivo para determinação e atualização dos valores de $pbest$ e $gbest$. O próximo passo consiste em calcular a velocidade e a posição de cada partícula com (2) e (3) respectivamente (EBERHART;

KENNEDY, 1995). A última etapa consiste em verificar se o critério de parada foi atendido. Em caso afirmativo, então é retornado o valor de $gbest$, mas em caso negativo o PSO fecha o laço de repetição voltando a etapa de avaliação.

$$v_i(t+1) = v_i(t) + c_1 \cdot r_1 \cdot (pbest - x(t)) + \dots + c_2 \cdot r_2 \cdot (gbest - x(t)) \quad (2)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (3)$$

As constantes c_1 e c_2 são chamadas de constantes de aceleração, já r_1 e r_2 são valores randômicos (EBERHART; KENNEDY, 1995). Em Kennedy (1997) o produto entre c e r é substituído por ϕ , um número positivo e aleatório. No segundo membro de (2), o segundo termo (que contém $pbest$), Kennedy (1997) o denomina de parcela cognitiva, enquanto o terceiro termo (que possui $gbest$) é a parcela social.

Em Kennedy (1997) são realizados alguns testes. O primeiro deles trabalha com variações de ϕ e do limite de velocidade V_{max} . Nele notou-se que geralmente, quando o V_{max} estava baixo, as partículas tinham mais dificuldade em sair das regiões de ótimo local (KENNEDY, 1997). Outro modelo testado foi o que descarta a parcela social de (2). Os indivíduos deste modelo apresentaram uma tendência de exploração apenas das áreas em que foram inicializados e, pelo menos quando V_{max} e ϕ eram pequenos, não conseguiram se mover para regiões ótimas. Também notou-se a necessidade de mais iterações em relação ao modelo anterior (modelo completo) em 47 das 49 condições testadas (KENNEDY, 1997). No modelo que desconsiderou a parcela cognitiva o algoritmo social foi constatado como um otimizador mais eficiente do que as versões somente cognitivas e completas testadas anteriormente para o problema de rede neural em particular (KENNEDY, 1997).

3. Variações nos algoritmos e suas implicações

Nesta seção são apresentados algoritmos (AG, DE e PSO) que utilizam versões modificadas das meta-heurísticas. O intuito é analisar as consequências que cada tipo de

modificação produziu.

3.1 GA

Em Santos, Velho e Chiwiacowsky, (2011) o GA é combinado com um Método de Gradiente Conjugado (MGC), a finalidade é avaliar danos estruturais. O MGC consiste em um método iterativo de busca para a determinação do mínimo local de uma função.

O MGC é usado para produzir uma estimativa inicial para o GA e para efeito de comparação (SANTOS; VELHO; CHIWIACOWSKY, 2011). O critério de desempenho utilizado foi o erro de estimação do dano, para os casos teóricos, além do erro mínimo para aplicações práticas (SANTOS; VELHO; CHIWIACOWSKY, 2011).

Constatou-se que a hibridização permitiu obter melhores resultados, em média, para os casos com poucos elementos danificados, mesmo com alta intensidade do dano, mas para o caso de muitos elementos com dano de baixa intensidade o MGC (sozinho) apresentou melhores resultados, em média (SANTOS; VELHO; CHIWIACOWSKY, 2011). O MGC como originalmente desenvolvido dificilmente encontra soluções inversas satisfatórias em problemas hiperbólicos, isso acontece por causa da não eliminação de incertezas, como flutuações e ruído, nas condições iniciais (SANTOS; VELHO; CHIWIACOWSKY, 2011).

O algoritmo híbrido soluciona este problema, por já começar com uma solução, em média, melhor que uma condição aleatória (SANTOS; VELHO; CHIWIACOWSKY, 2011). Este início mais próximo à região do mínimo global oferece uma maior probabilidade de a solução final estar mais próxima da solução desejada (SANTOS; VELHO; CHIWIACOWSKY, 2011). Por fim, a hibridização conseguiu melhorar a média global de estimação do erro e também foi capaz de reduzir o limitante superior do erro (SANTOS; VELHO; CHIWIACOWSKY, 2011). Isso tudo sem comprometer a eficiência e o tempo para obtenção do resultado (tempo de computação da solução), que foi de 55 milissegundos para o método híbrido e de 53 milissegundos para o MGC (SANTOS; VELHO; CHIWIACOWSKY,

2011).

3.2 DE

A proposta de Lopes e Freitas (2017) é a implementação de um Modelo Celular em Ilhas ICM – Island Cellular Model) para o algoritmo de Evolução Diferencial e avaliação do seu desempenho. O Modelo Celular (CM - *Cellular Model*) neste trabalho, faz com que a população de indivíduos seja espacialmente distribuída em uma grade, nela só é permitido que cada indivíduo interaja para reproduzir com um grupo de indivíduos vizinhos (LOPES; FREITAS, 2017). Normalmente, o CM separa os indivíduos de uma mesma população com um arranjo em grade bidimensional, no formato toroidal (LOPES; FREITAS, 2017).

O Modelo em Ilhas (IM – *Island Model*) trabalha em essência com a divisão da população de soluções candidatas em subpopulações denominadas de ilhas (LOPES; FREITAS, 2017). Durante a evolução, as subpopulações se mantêm relativamente isoladas e evoluem de modo independente executando o seu próprio AE dentro de um espaço de tempo (LOPES; FREITAS, 2017). Periodicamente, pode ocorrer o processo de migração, em que soluções são trocadas entre as ilhas. Este processo é repetido até que se atinja o critério de parada do AE (LOPES; FREITAS, 2017).

No ICM, os modelos IM e CM compõem sua estrutura hierárquica. Simplificadamente a ideia geral é projetar duas camadas para distribuir a população, uma superior e outra inferior (LOPES; FREITAS, 2017). Para a camada superior do ICM, a população de soluções candidatas é dividida em subpopulações. Depois o CM é utilizado em cada subpopulação anteriormente definida (LOPES; FREITAS, 2017).

A versão do algoritmo DE com a implementação do CM limita o operador mutação, limitando o conjunto de indivíduos viáveis para determinação de novas soluções (LOPES; FREITAS, 2017). Assim, somente soluções que pertençam ao conjunto de indivíduos vizinhos são selecionadas para reprodução (LOPES; FREITAS, 2017).

Versões do ICM e do DE clássico foram

implementadas, os parâmetros do DE foram escolhidos como $CR = 0,9$ e F determinado aleatoriamente dentro do intervalo $[0,5; 1,0]$ para cada novo indivíduo gerado e um tamanho de população igual à 25 foi adotado (LOPES; FREITAS, 2017). Para o ICM, os valores utilizados em alguns parâmetros foram: número de ilhas igual a 4, 8 e 16; tamanho da população de 25 indivíduos por ilha; frequência de migração de 5, 15 e 25 gerações; taxa de migração de 1 indivíduo; entre outros (LOPES; FREITAS, 2017).

Em resumo, os resultados indicaram que um menor número de ilhas é capaz de convergir de modo rápido além de determinar boas soluções (LOPES; FREITAS, 2017). O bom desempenho pode ser justificado pela elevação da pressão seletiva provocada por ICM menores (LOPES; FREITAS, 2017). O estudo experimental realizado demonstrou o bom comportamento do ICM, reafirmando sua capacidade de aperfeiçoar o desempenho dos AEs (LOPES; FREITAS, 2017).

3.3 PSO

Em Silveira et al. (2009) é apresentada uma técnica para reduzir as chances do PSO, aplicado em funções não lineares, ficar preso em mínimos locais. A ideia principal do trabalho é utilizar uma forma não monotônica de controlar a inércia da partícula (SILVEIRA et al. 2009). A atualização da velocidade de cada partícula é realizada de acordo com (2), mas ω multiplicando a velocidade atual $V_i(t)$.

A inclusão de ω tem o intuito de equilibrar a busca local e a global. Um valor alto de ω favorece a busca global, ao contrário de um baixo valor, que favorece a busca local.

O PSO com ω não monotônico funciona como o PSO convencional, exceto pela atualização do ω definida por (4) (SILVEIRA et al. 2009). No PSO padrão, este termo é atualizado com uma subtração linear.

$$\omega = \left[\cos \cdot \left(\frac{\pi i}{2 \cdot \text{ciclos}} \right) \cdot m \right] + s \quad (4)$$

Em (4): i representa a iteração; ciclos são o número de ciclos não monotônicos utilizados; m é um multiplicador determinado pela

diferença entre o valor máximo e mínimo de ω (fornecidos antes do início do processo) e depois dividido por 2; s é o deslocamento da função, calculado pela soma de m com o valor mínimo de ω (SILVEIRA et al. 2009).

Em Silveira et al. (2009), foram feitos testes com o PSO padrão e o PSO com o controle de inércia ω não monotônico, além da aplicação de ambos os modelos em um problema real para o ajuste de pesos sinápticos de uma rede neural do tipo *Multi-Layer Perceptron*. Segundo os testes, ambos os métodos determinaram bons resultados porém, com ω não monotônico, existe uma maior probabilidade de fuga de um mínimo local encontrado, devido à capacidade de busca global produzida pela energia fornecida ao sistema através controle de inércia (SILVEIRA et al. 2009). O que pode gerar resultados mais satisfatórios, especialmente em problemas complexos (SILVEIRA et al. 2009). O uso de ω não monotônico é capaz de encontrar uma boa solução em pouco tempo, entretanto seu refinamento é mais demorado que o do PSO tradicional (SILVEIRA et al. 2009).

Para a aplicação de ambos os modelos de PSO em um caso real de otimização, ambos também apresentaram-se interessantes (SILVEIRA et al. 2009). Segundo os testes, Silveira et al. (2009) verificou e concluiu que, em algumas situações, o PSO padrão precisa de uma capacidade de busca global quase no fim do experimento, especialmente para problemas complexos. Concluiu-se também que o ω não monotônico pode fornecer essa capacidade, muitas vezes durante o processo de otimização, melhorando-o (SILVEIRA et al. 2009).

4. Conclusão

Ao final deste trabalho, conclui-se que apesar dos algoritmos AG, DE e PSO padrão representarem um significativo avanço em termos de métodos numéricos, eles possuem deficiências e características que podem ser aperfeiçoadas. A possibilidade de aprimoramento desses algoritmos foi constatada com a apresentação de alguns trabalhos. Neles, o AG, o DE e o PSO sofreram modificação estrutural ou foram modificados e utilizados em conjunto com outras técnicas. O que proporcionou

melhores resultados, principalmente em comparação com os respectivos algoritmos padrão.

Para finalizar, como sugestão de trabalho futuro, considera-se a utilização, em uma aplicação prática, de métodos AG, ED e PSO modificados e/ou combinados com diferentes métodos. Com o intuito de posteriormente compará-los em termos de desempenho e qualidade dos resultados apresentados.

REFERÊNCIAS

CHAPRA, S. C.; CANALE, R. P. **Métodos Numéricos para Engenharia**. Porto Alegre: McGraw-Hill, 2008, pp. 2.

ZAPELINI, C. Z. **Um Estudo Abrangente Sobre Metaheurística**. incluindo um histórico. Monografia. Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2009.

SERAPIÃO, A. B. S. **Fundamentos de Otimização por Inteligência de Exames: Uma Visão Geral**. Revista Controle & Automação, Rio Claro, vol. 20, nº. 3, pp. 271-304, Jul/Ago./Set. 2009.

MELO, W. A. X.; FAMPA, M. H. C.; RAUPP, F. M. **P. Evolução diferencial aperfeiçoada para otimização contínua restrita**. 42º Simpósio Brasileiro de Pesquisa Operacional, Bento Gonçalves, 2010.

MENDONÇA, I. M.; MOREIRA, T. G.; JÚNIOR, I. C. S.; MARCATO, A. L. M.; DIAS, B. H., et. al., **Planejamento Estático Da Expansão De Sistemas De Transmissão De Energia Elétrica Via Otimização Por Enxame De Partículas**. 43º Simpósio Brasileiro de Pesquisa Operacional, Ubatuba, 2011.

COELHO, L. S.; PESSÔA, M. W. **A Tuning Strategy for Multivariable PI And PID Controllers Using Differential Evolution Combined With Chaotic Zaslavskii Map**, Elsevier. Expert Systems with Applications, vol. 38, pp. 13694-13701, 2011.

HOLLAND, J. H. **Adaptation in Natural and Artificial Systems: An Introductory Analysis with Application to Biology, Control, and Artificial Intelligence**. Ann Arbor, MI: University of Michigan Press, 1975.

ESHELMAN, L. J. **Evolutionary Algorithms and their Standard Instances: Genetic Algorithms**, In: K. Jong, D. B. Fogel e H. S. Schwefel, Handbook of Evolutionary Computation. 1. ed. Bristol: IOP Publishing Ltd and Oxford University Press, ch. B1.2, pp. 44-45, 1997.

SOUZA, D. L.; MAIA, L. F. S.; LOBATO F. S.; ALMEIDA, G. M. **Sintonia de Controladores PID com o Algoritmo de Evolução Diferencial**. X Simpósio Brasileiro de Automação Inteligente. Anais, pp. 105-110, Set. 2011.

STORN, R.; PRICE, K. **Differential Evolution—A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces**. Journal of Global Optimization, vol. 11, nº 4, pp. 341-359, 1997.

ANDRADE, L. H. S.; COSTA, B. L. G.; ANGÉLICO, B. A.; **PSO Aplicado a Sintonia do Controlador PI/PID da Malha de Nível de uma Planta Didática Industrial**. 11º Simpósio Brasileiro de Automação Inteligente, Fortaleza, 2013. Anais eletrônicos. Disponível em: <<http://www.sbai2013.ufc.br/pdfs/7935.p df>>. Acesso em 08 mar. 2018.

FALCUCCI, M.; JÚNIOR, J. N. R. S.; BRAGA, A. P. S.; NASCIMENTO, J. A. **Determinação dos Parâmetros Ótimos de um Controlador PID Aplicado na Regulação de Tensão de Geradores Síncronos Através do Método de Otimização por Enxame de Partículas – PSO**. 1º Simpósio Brasileiro de Inteligência Computacional, Florianópolis. 2007, Anais.

EBERHART, R.; KENNEDY, J. **A New Optimizer Using Particle Swarm Theory**. MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, 1995, pp. 39-43.

KENNEDY, J. **The Particle Swarm: Social Adaptation of Knowledge**. Proceedings of 1997 IEEE International Conference on Evolutionary Computation (ICEC '97), Indianapolis, IN, USA, 1997, pp. 303-308.

SANTOS, L. B. L.; VELHO, H. F. C.; CHIWIACOWSKY, L. D. **Análise de Robustez do Método Híbrido de Estimação de Dano Estrutural**. TEMA (São Carlos) [online]. 2011, vol.12, n.3 [cited 2018-08-21], pp.245-252. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S2179-84512011000300008&lng=en&nrm=iso>. ISSN 2179-8451. <http://dx.doi.org/10.5540/tema.2011.012.03.0245>. Acesso em: 15 ago 2018.

LOPES, R. A.; FREITAS, A. R. R. **Um modelo híbrido de distribuição populacional para o algoritmo de evolução diferencial**. XLIX Simpósio Brasileiro de Pesquisa Operacional, Blumenau, 2017.

SILVEIRA, T.; OLIVEIRA, H. C. B.; SILVA, L. E.; SALGADO, R. M. **Controle De Inércia Não Monotônico Na Otimização Por Enxame De Partículas**. SCIENTIA – Interdisciplinary Studies in Computer Science, vol. 20, ed. 2, pp. 69-82, jul./dec., 2009.