

Comparação entre os Algoritmos de Otimização por Enxames de Partículas (PSO), FSS e ABC

2ª Prática de Inteligência de Enxames

Marlon Castro
Disciplina: Computação Natural
Prof. Carmelo Bastos
Curso de Engenharia da
Computação

Poli UPE – Escola Politecnica
Recife – PE
E-mail: msc@ecompp.poli.br

Resumo - Esta prática consiste na execução e análise dos algoritmos de pso – algoritmo de otimização por enxames, muito utilizado em reconhecimento de padrões e busca de soluções ótimas para problemas com grande quantidade de variáveis. Bem como a comparação dos do PSO com os algoritmos FSS e ABC.

I. INTRODUÇÃO [3]

PARTICLE SWARM OPTIMIZATION

Proposto por Russel Eberhart e James Kennedy em 1995, o PSO inicialmente fora usado para a solução de otimização de problemas não lineares, citando exemplos, problemas referentes a pesos e estruturas de redes neurais, composição de músicas, e aprendizado em jogos foram os alvos iniciais do algoritmo.

Baseado no comportamento das revoadas, o PSO se sustenta no comportamento de comunicação entre as partículas – relevando as topologias – a comunicação entre todo o enxame tende a convergir as partículas na melhor posição possível no espaço de busca. Esse comportamento se dá pelo fato de cada partícula comunicar qual sua posição, e avaliando cada partícula, a que estiver na melhor posição é o alvo de encontro das demais.

O PSO original foi desenvolvido utilizando a topologia global como padrão, assim, cada partícula do enxame se comunica com todas as outras, informando suas posições e definindo a cada passo do algoritmo, aquela partícula que possui o melhor fitness, ou seja, o melhor valor naquele passo. O algoritmo segue até que seja encontrado entre todos os passos o melhor valor de fitness, o pBest.

A movimentação das partículas durante cada passo é calculada através de uma fórmula, a qual influencia a posição da partícula. Essa fórmula é dependente de outras variáveis e constantes, onde:

- Velocidade da partícula na iteração atual.
- Posição da partícula na iteração atual.
- Constante de individualidade, que influencia no retorno a melhor posição já encontrada por ela mesma.
- Constante de sociabilidade, que influencia na ida à melhor posição encontrada de quaisquer vizinhos.
- Valores reais randômicos.
- Coeficiente de inércia, que controla os valores da velocidade da iteração anterior
- Melhor posição já encontrada pela partícula.
- Melhor posição já encontrada entre todas as partículas

Algoritmo ABC

O Algoritmo de Colônia de Abelhas Artificiais, ou Artificial Bee Colony (ABC), foi proposto por Karaboga em 2005 [1] com o intuito de resolver problemas de otimização baseando-se no comportamento inteligente das abelhas na natureza [2]. Na área da Inteligência Computacional, é um algoritmo identificado com a Otimização por Enxames de Partículas (Particle Swarm Optimization - PSO), usando apenas alguns ajustes de parâmetros, tais como tamanho da colônia de abelhas e número máximo de ciclos.

Desde que foi proposto, o algoritmo ABC tem sido utilizado em várias aplicações, como treinamento de redes neurais artificiais, otimização de portfólios, dimensionamento de lotes, identificação de objetos em imagens digitais, clusterização [2-6]. Em particular, há uma publicação recente com aplicação na área de sistemas elétricos de potência [7], tratando da resolução do problema de fluxo de carga ótimo multiobjetivo.

O algoritmo é desenvolvido de forma a explorar as características de auto-organização e divisão do trabalho, típicas dos enxames de abelhas que produzem mel.

Numa colônia, existem diversos tipos de abelhas, mas as abelhas utilizadas no algoritmo são de três tipos apenas: campeiras (scout), operárias (employed) e seguidoras (recruited). As campeiras são responsáveis por procurar fontes de alimentos, a partir do momento que uma abelha campeira explora uma fonte de alimento, ela torna-se uma abelha operária. Quando a fonte de alimento esgota, a abelha operária volta a ser uma abelha campeira. E as abelhas seguidoras são as abelhas que ajudam as operárias a explorar a fonte de alimento.

O algoritmo foi inspirado, como mencionado anteriormente, no comportamento inteligente das abelhas na natureza para

encontrar, de forma otimizada, fontes de alimentos, as quais são pontos no espaço de busca do algoritmo [6]. A Tabela 1 mostra os principais parâmetros do algoritmo e a Figura abaixo mostra o fluxograma do algoritmo.

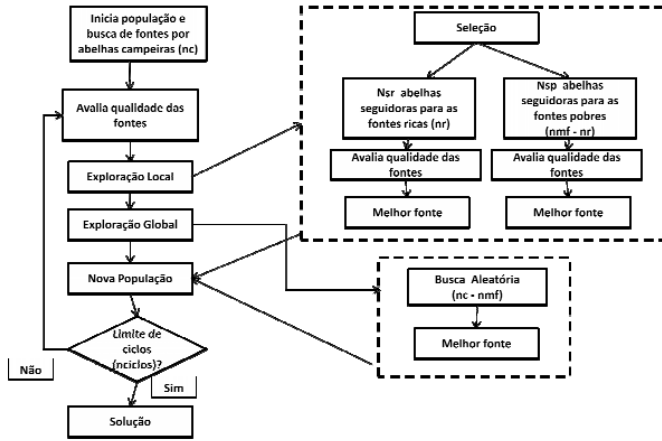


Figura 8. Fluxograma do algoritmo das abelhas

Otimização por cardume (FSS)

Na Topologia Focal, é eleito uma partícula base para que todas as outras do enxame se bassem nesta. E Observe que se a partícula principal cair num mínimo rapidamente, o algoritmo cessa a execução não trazendo resultados ótimos. Essa topologia também é conhecida como estrela (do inglês: star).

O algoritmo de otimização baseado em cardumes foi proposto em 2008 [1] e, como já mencionado, baseia-se na busca de alimentos por parte de um cardume. Foram observados os seguintes aspectos presentes em cardumes reais:

1. Os peixes se organizam em cardumes de maneira a evitar o ataque de predadores e facilitar a obtenção de comida;
2. Cada peixe possui certa independência para buscar alimento por conta própria sem por isso abandonar o cardume;
3. O cardume também nada para uma mesma direção no oceano;
4. Quando a comida torna-se abundante em um certo ponto do cardume, ele contrai-se em torno daquela região para alimentar-se. Em contrapartida, quando a comida torna-se esparsa, ocorre uma dilatação no cardume com o intuito de aumentar a região de busca;

Em uma adaptação do cardume real para o algoritmo, cada peixe torna-se uma entidade e a busca por comida corresponde à busca por melhores soluções (ou fitness) na função a ser otimizada. Em outras palavras, os peixes movem-se em um grande aquário de n dimensões onde cada dimensão representa uma variável a ser otimizada e onde os limites são os valores máximo e mínimo que cada variável da função pode assumir.

Antes de descrever o comportamento dos peixes no algoritmo, faz-se necessário definir alguns conceitos intrínsecos à busca por cardume. O primeiro deles é o peso individual dos peixes. Esse peso é a medida de quão gordo (e bem-sucedido) é o peixe. Peixes gordos representam um histórico de sucesso e exercem uma maior influência no cardume que os magros.

O segundo conceito é o de passo. O passo, como o nome sugere, representa quanto o peixe irá nadar em certa dimensão. Sendo assim, ele representa um deslocamento no espaço de busca em certa dimensão e controla a busca por profundidade e por largura do algoritmo.

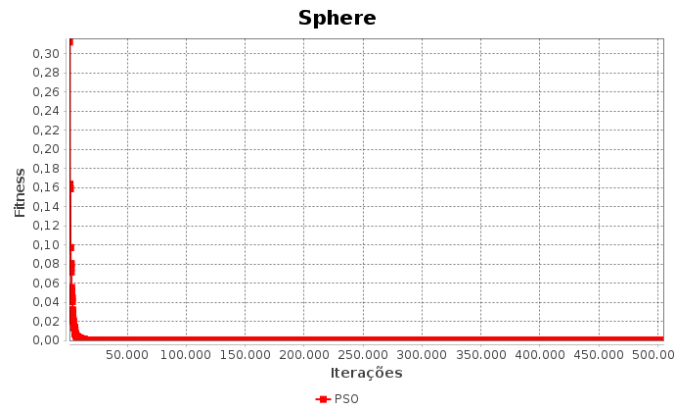
O terceiro e último conceito é a divisão do movimento do cardume em duas categorias, que são: movimento individual e movimento coletivo. Cada movimento possui uma correspondência direta com as características observadas na natureza. Segue abaixo o Pseudocódigo do exemplo utilizado neste relatório.

Algoritmo 1: Pseudocódigo do FSS.

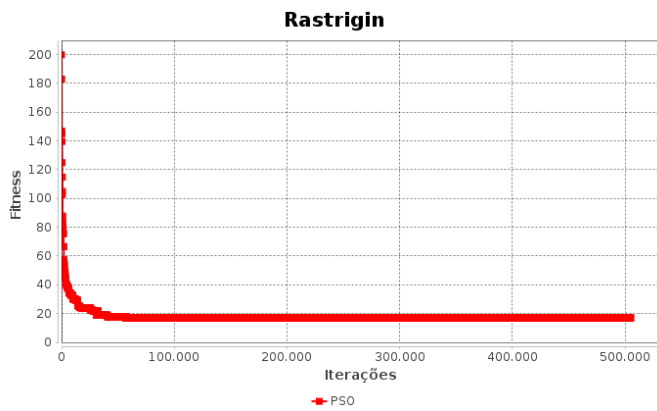
```

1 início
2   Inicialize todos os peixes em posições  $\vec{x}_i(0)$  aleatórias e distantes do mínimo global;
3   Inicialize aleatoriamente o peso  $\vec{w}_i(0)$  de todos os peixes;
4   enquanto critério de parada não for alcançado faça
5     para cada peixe faça
6       Encontre posição vizinha (2.1);
7       Avalie a variação no fitness de acordo com (2.2) e mova o peixe apenas se o fitness melhorou;
8       Alimente os peixes utilizando (2.7);
9     Calcule o vetor do movimento instintivo utilizando (2.3);
10    para cada peixe faça
11      Execute o movimento instintivo utilizando (2.5);
12    Calcule o baricentro utilizando (2.6);
13    para cada peixe faça
14      Execute o movimento volitivo usando (2.8);
15    Atualize  $s_{ind}$  e  $s_{vol}$ .
  
```

ANÁLISES

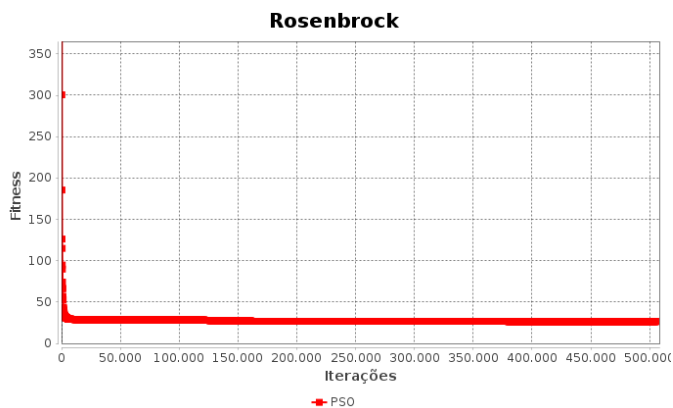
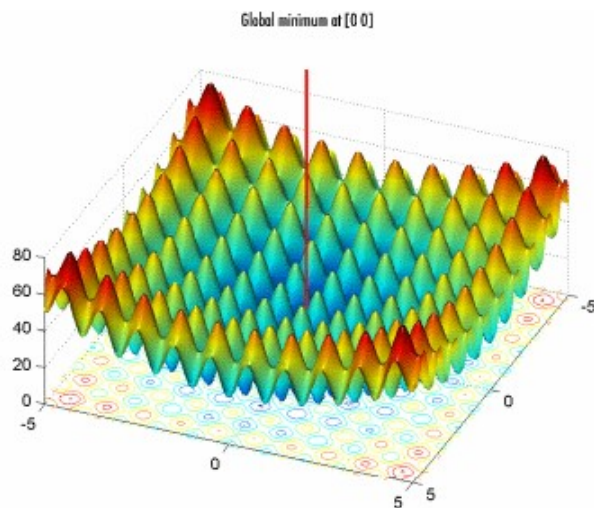


O PSO na topologia local apresenta rápida convergência, dado o fato do problema Sphere não tem mais de um mínimo da função, a otimização acontece sem problemas de prisão em mínimos locais.



O Problema Rastrigin é mais complexo pois se trata de uma função com inúmeros mínimos locais, o algoritmo apresenta dificuldades na convergência para o menor mínimo.

Gráfico da função Rastrigin

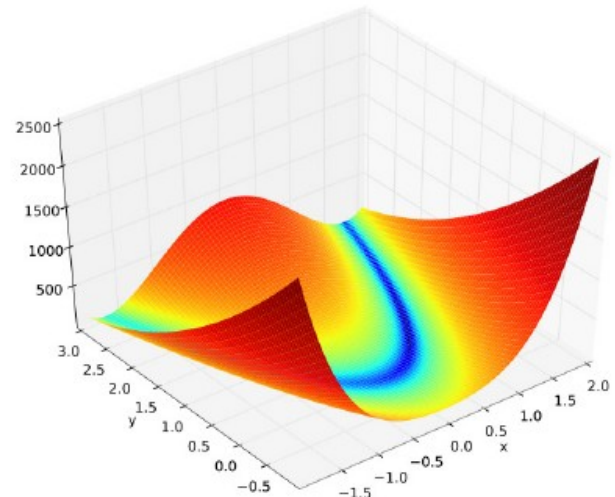


O Problema Rosenbrock é uma função com o mínimo global em (0,0,...) de acordo com a quantidade de dimensões repassadas. Apresenta boa convergência porém o algoritmo às vezes se perde em mínimos locais.

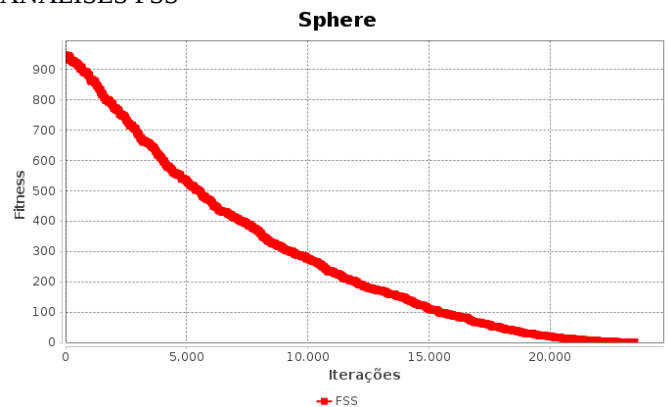
Observação (Rosenbrock): Pode-se observar que mesmo o coeficiente de inércia estando em 0.8, a velocidade de deslocamento continuava dificultando a otimização da partícula, porém com a versão decremental, com o passar das

iterações, foi possível otimizar os valores, e também, devido ao melhor deslocamento em alguns casos, obter resultados melhores que o fator de Clerc, que por sua vez mostrou uma excelente velocidade para otimizar mais rapidamente o problema.

Gráfico da função Rosenbrock

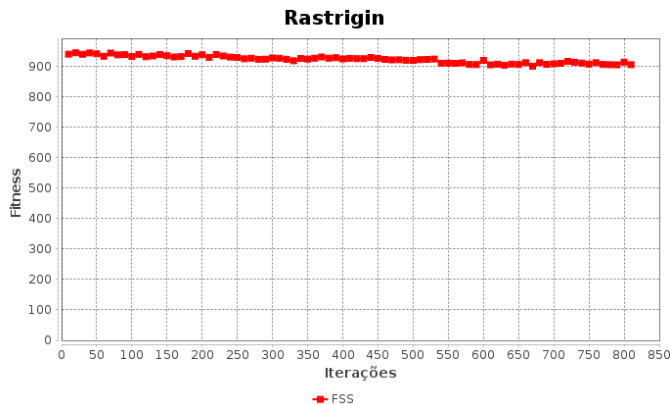


ANALISES FSS

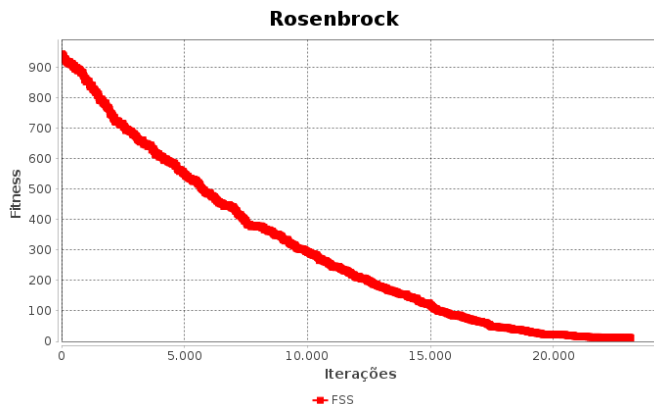


Em comparação com o PSO, o FSS tem uma convergência mais suave abrangendo uma parte de problemas com foco na otimização de funções multimodais. Do Gráfico, para a avaliação de 500.000 iterações foram eliminados ruídos nos fitness duplicados e zerados de forma a compor um gráfico melhor visualizado.

O Problema Rastrigin aparentemente não convergiu nas análises, o motivo não pode ser detectado, foram feitas diversas iterações e mudanças de parâmetros, mas sem sucesso.



O Desempenho do Rosenbrock se assemelha ao Sphere, pelo fato de conter poucos mínimos locais, a convergência permanece suave conforme gráfico abaixo.



O Problema Rastrigin é mais complexo pois possui inúmeros mínimos locais o que pode dificultar a otimização.

Diante das figuras acima, observa-se o desempenho para a convergência das soluções no decorrer das iterações. Veja que em ambas as funções, a topologia local converge rapidamente enquanto a local fica presa num mínimo local.

Conclusão

Observação (Rosenbrock): Pode-se observar que mesmo o coeficiente de inércia estando em 0.8, a velocidade de deslocamento continuava dificultando a otimização da partícula, porém com a versão decremental, com o passar das iterações, foi possível otimizar os valores, e também, devido ao melhor deslocamento em alguns casos, obter resultados melhores que o fator de Clerc, que por sua vez mostrou uma excelente velocidade para otimizar mais rapidamente o problema.

Estes algoritmos formam uma sub-área homônima, conhecido como Inteligência de Enxames. Fish School Search (FSS), uma família de algoritmos de otimização, parte da sub-área de Inteligência de Enxames, foi introduzida para solução de problemas unimodais, em sua maioria. Os mecanismos de alimentação e a movimentação coordenada presentes em

cardumes naturais de peixe foram utilizados como inspiração para a criação de mecanismos de busca coletiva para o algoritmo. Este trabalho investiga como o peso dos peixes no FSS pode(m) ser utilizado(s) para, automaticamente, particionar os cardumes, objetivando encontrar múltiplos ótimos em funções multimodais (i.e. usando apenas informações locais). As principais modificações feitas no FSS original foram: criação de uma regra para formação de ligações virtuais entre peixes, usando apenas o peso dos peixes como critério, e a adição de um fator multiplicador que regula a influência dos operadores de movimento coletivo no final do deslocamento dos peixes.

- [1] Kennedy, J.; Eberhart, R. (1995). «Particle Swarm Optimization». *Proceedings of IEEE International Conference on Neural Networks. IV*. pp. 1942–1948. doi:10.1109/ICNN.1995.488968
- [2] Luiz Felipe Vieira Verçosa. (TCC) *Estudo Paramétrico do FSS II*, Recife, Dezembro de 2014, p. 5.
- [3] Nathalia Maria Temudo. (TCC) *Um algoritmo baseado em Artificial Bee Colony para treinamento de Redes Neurais Artificiais*, Recife, Junho de 2010, p. 22.
- [4] Wikipedia, endereço: https://pt.wikipedia.org/wiki/Otimiza%C3%A7%C3%A3o_por_enxame_de_part%C3%Adculas, acessado em 22 de maio de 2019.
- [5] Códigos fonte do projeto: https://github.com/marlonscastro/computacao_natural/tree/master/java/