

Algoritmo PSO para otimização binária

PSO algorithm for binary optimization

RESUMO

O algoritmo de otimização por enxame de partículas (PSO) é uma metaheurística de busca que é inspirado no comportamento coletivo de grupos de animais, como abelhas, como o modo de trabalho das formigas, a busca por alimento dos cardumes de peixes, voo dos pássaros, entre várias outras possibilidades. O objetivo do PSO é encontrar a solução para um problema complexo usando como base a forma que estes animais solucionam os seus próprios problemas, ou seja, tenta-se emular o comportamento de enxame (cardume, formigueiro, bando, etc.) por meio de uma linguagem de programação. Foi desenvolvido neste estudo um algoritmo de PSO binário e testado no OneMax Problem com diversos valores de entradas. Os resultados obtidos mostram a viabilidade da proposta.

PALAVRAS-CHAVE: BPSO. Algoritmos. Otimização.

ABSTRACT

The particle swarm optimization algorithm (PSO) is a search metaheuristic that is inspired in the collective swarm behavior, such as bees, as the way ants work, the search for food from schools of fish, flight of birds, among many other possibilities. The goal of PSO is to find the solution to a complex problem based on the way these animals solve their own problems, i.e. it tries to emulate swarm behavior (shoal, tingling, flock, etc.) through programming. A binary PSO algorithm was developed and tested in the OneMax problem with several input values. The results showed the viability of the proposal

KEYWORDS: BPSO. Algorithm. Optimization

Lucas Henrique Biuk
lucasbiuk1902@gmail.com
Universidade Tecnológica Federal
do Paraná, Ponta Grossa, Paraná,
Brasil
Hugo Valadares Siqueira
hugosiqueira@utfpr.edu.br
Universidade Tecnológica Federal
do Paraná, Ponta Grossa, Paraná,
Brasil

Recebido: 19 ago. 2019.

Aprovado: 01 out. 2019.

Direito autoral: Este trabalho está licenciado sob os termos da Licença Creative Commons-Atribuição 4.0 Internacional.



INTRODUÇÃO

Os algoritmos de otimização bio-inspirados são métodos da área de Inteligência Computacional (IC). Diferentemente da Inteligência Artificial (IA), que possui foco em desenvolver modelos computacionais inspirados no raciocínio humano, tem como princípio a observação da natureza para resolução de problemas reais (KENNEDY, J.; EBERHART, R., 1995). Para isso, desenvolveram-se diversos tipos de algoritmos, tendo em vista a vastidão de comportamentos que podem ser observados na natureza, desde plantas e animais até sistemas imunológicos.

A Computação Natural (computação inspirada na natureza) vem como uma alternativa às soluções já existentes para determinados problemas, isto é, a sua aplicação é mais conveniente quando se possui um problema de alta complexidade, com grande quantidade de variáveis e possíveis soluções, uma vez que, se o problema não necessita desse tratamento, provavelmente ele possui alguma solução específica. Sendo assim, tornou-se interessante entender melhor o funcionamento dos algoritmos deste tipo, para que se possa então fazer a correta aplicação de cada um deles com suas respectivas vantagens com tipo de problema. Para isso, deseja-se realizar uma comparação entre diversos algoritmos para descobrir quais as vantagens de uns em relação aos outros.

OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS (PSO)

O algoritmo de otimização por enxame de partículas (PSO) foi apresentado por James Kennedy e Russell Eberhart em 1995, e desde então foi muito utilizado para soluções de problemas no campo das ciências e das engenharias (ALAM, Mahamad Nabab, 2016) como pode ser amplamente observado ao ser utilizado, por exemplo, na otimização de parâmetros de um controle PID aplicado sobre um conversor Buck, onde o intuito foi utilizar o PSO para encontrar os valores das constantes do controlador (Puchta et al., 2016). A área de Inteligência Computacional vem numa crescente quando o assunto é otimização, pois como se sabe, surgem cada vez mais tecnologias que necessitam, por demanda geral, rapidez, confiabilidade, precisão, dentre outras características, que por sua vez, são difíceis de serem otimizadas.

Uma observação simples da natureza pode ser vista assim: um número X de indivíduos estão dispostos aleatoriamente em uma área/campo onde há (ou pode haver) muitos morros e vales, lugares com maior ou menor quantidade de alimento, lugares que oferecem mais segurança, ou melhor oportunidade de caça. O conjunto de todos estes agentes é chamado população, que possui por objetivo alcançar o melhor espaço para si.

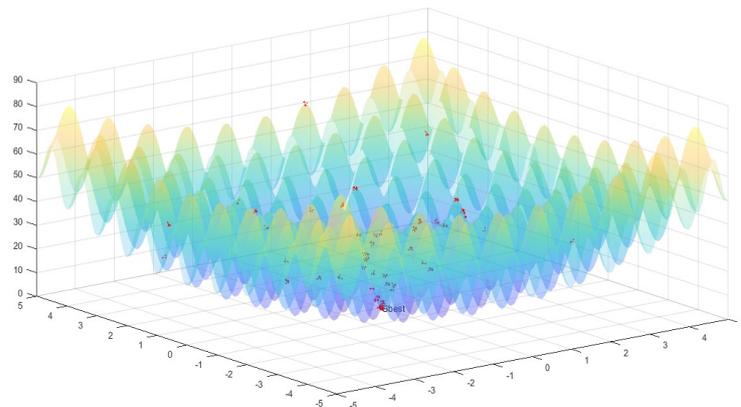
Para isso, cada agente, que até então se encontra em uma posição aleatória, começa a caminhar e trocar informações com os outros, avaliando se a sua posição é melhor ou pior que a do seu vizinho. Assim eles vão se direcionando àquele que possui a melhor posição dentre todos. Durante a caminhada pode ocorrer que algum indivíduo encontre um espaço que é ainda melhor daquele primeiro, e pelas informações trocadas entre os vizinhos eles mudam sua direção para o novo melhor lugar. Outro detalhe é a velocidade com que cada um se move: quando o indivíduo está “perdido” (longe dos vizinhos e

longe do melhor lugar) ele caminha mais depressa pois necessita encontrar a melhor posição, e quanto mais próximo de um pico (ou vale) ele caminha mais lentamente.

No parágrafo anterior lê-se a descrição do que seria uma observação genérica de um comportamento natural, esta será traduzida para uma linguagem de computação e pode ser utilizada para a resolução de algum problema. Trazendo a observação feita para uma linguagem técnica, o que se tem é um enxame de partículas dispersas num meio. Este muitas vezes é a representação gráfica de uma função matemática, como a função de Rastrigin na Figura 1, a qual procura-se otimizar, que é nada mais do que procurar seus pontos de máximos ou mínimos. Ela é uma boa função para se testar algoritmos de otimização por apresentar grande quantidade de mínimos locais (ÁVILA, Sérgio Luciano et al., 2002).

Por exemplo, se existe uma função que representa o dispêndio de energia para realização de um trabalho, procura-se minimizar o seu valor, que otimizado seria o mínimo global desta função. No PSO, cada uma das partículas possui a sua "melhor posição pessoal", nomeado *personal best* (*pbest*), que é onde o indivíduo apresenta seu melhor *fitness*, e a partícula que possui a melhor posição dentre as outras é nomeada *global best* (*gbest*), que é onde se encontra o indivíduo com o melhor *fitness* dentre todos os outros. A seguir, na Figura 2, pode ser observado um fluxograma representando o funcionamento do algoritmo do PSO real.

Figura 1: Função de Rastrigin no intervalo [-5, 5]



Fonte: Autoria Própria

A atualização da posição se dá a partir da Equação 1 abaixo:

$$X_{i(t+1)} = X_{i(t)} + V_{i(t+1)} \quad (1)$$

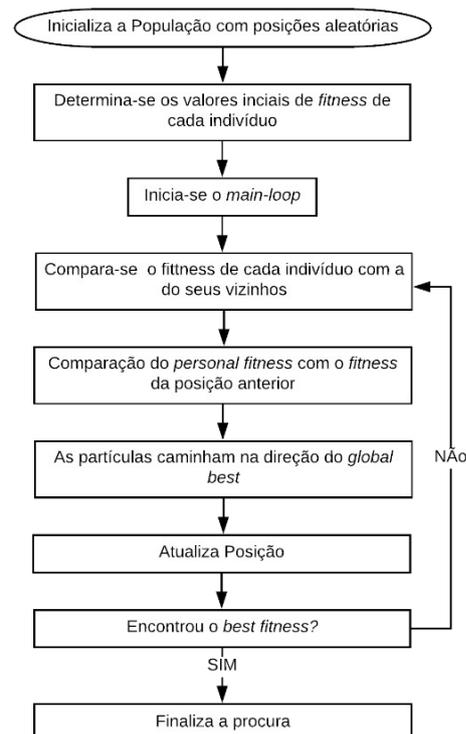
em que:

$$V_{i(t+1)} = w \cdot V_i + K_1 \cdot r_1 \cdot (pbest_i - X_i) + K_2 \cdot r_2 \cdot (gbest - X_i) \quad (2)$$

sendo:

- $V_i(t+1)$: velocidade para a próxima iteração;
- V_i : velocidade atual;
- w : Constante de inércia, a qual decai de 0,9 à 0,4;
- X_i : vetor de posições de todas as partículas vindos do *looping* anterior;
- K_1 : constante cognitiva;
- K_2 : constante social;
- r_1 : constante randômica 1
- r_2 : constante randômica 2
- $pbest$: Melhor posição da partícula sendo analisada (*personal best*)
- $gbest$: Melhor posição global dentre todas partículas (*global best*)

Figura 2: Fluxograma de funcionamento PSO



Fonte: Autoria Própria

O PSO, se comparado com outros algoritmos de otimização, como o Algoritmo Genético (AG) por exemplo, é de implementação simples porque depende, basicamente, de somente um operador, que é a velocidade, contra os vários processos de seleção, mutação e *crossover* do AG, além de ser um algoritmo de convergência rápida. A ação da velocidade sobre as posições se dá a

partir da Expressão (2). Além disso, há também a necessidade de se fazer a atualização do valor da inércia, pois cada interação faz com que as partículas se aproximem mais e mais do *best fitness*, e quanto mais próximo, deseja-se que a partícula caminhe mais lentamente.

PSO BINÁRIO

O projeto de pesquisa se estendeu com o intuito de resoluções de problemas que necessitam de otimização binária, em que se utiliza apenas bits, ou seja, problemas On-Off e/ou de seleção de variáveis. O PSO por ser de fácil implementação, como citado, só depende do operador velocidade e tendendo a uma convergência rápida, se mostra um bom candidato a resolução destes problemas. Sendo assim, iniciou-se a aplicação do PSO binário (BPSO) de duas maneiras: a primeira convertendo os valores reais para valores binários; e a segunda iniciando as populações (vetores) já no domínio binário. Para a conversão dos vetores reais, que representam as populações com as respectivas velocidades de suas partículas, em binários (zeros e uns), utilizou-se funções de transferência, como a sigmoide (S-Shaped) e V-Shaped. Optou-se por usar as funções de transferências por serem a forma mais comum para este processo.

No BPSO as partículas são candidatas a soluções binárias para resolver o problema de otimização endereçada. Uma partícula tem alguns atributos, como a posição no espaço de busca multidimensional e a velocidade da partícula. A função *fitness*, como de costume, é o índice de desempenho, estritamente relacionados com a função de custo (Siqueira et al., 2018).

O estudo de caso aplica o algoritmo de BPSO ao *OneMax problem*. Este foi o problema escolhido por ser um problema clássico e de mais rápida aplicação para testes, muito utilizado atualmente para esta funcionalidade (SANTANA, Clodomir J. et al., 2019). Os resultados computacionais estão sumarizados na Tabela 1. Observe que mesmo para grandes dimensões, a taxa de convergência se aproxima do máximo global.

Tabela 1: Resultados das Otimizações - BPSO

Dimensões	Partículas	Iterações	Convergência	Convergência (%)
50	10	6	47	94,00%
100	10	8	97	97,00%
500	25	10	484	96,80%
5000	100	22	4869	97,38%

Fonte: Autoria Própria

CONCLUSÃO

O presente trabalho mostrou a aplicação da versão binária do algoritmo de Otimização por Enxame de Partículas em sua versão binária ao OneMax. Para tal, o mesmo foi desenvolvido e funções de transferência foram utilizadas para a transformação.

Os resultados computacionais para até 5000 dimensões mostram a viabilidade da proposta para tarefas deste tipo, já que quase sempre o máximo global ficou próximo.

Trabalhos futuros podem ser desenvolvidos no sentido de aplicar tal ferramenta a problemas reais.

AGRADECIMENTOS

Agradecemos Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo suporte financeiro, processo número 405580/2018-5.

REFERÊNCIAS

KENNEDY, J.; EBERHART, R. Particle swarm optimization (PSO). In: Proc. IEEE International Conference on Neural Networks, Perth, Australia. 1995. p. 1942-1948.

ALAM, Mahamad Nabab. Particle swarm optimization: Algorithm and its codes in matlab. ResearchGate, p. 1-10, 2016.

PUCHTA, Erickson Diogo Pereira et al. Controle PID gaussiano com otimização dos parâmetros das funções gaussianas usando algoritmo genético e PSO. 2016. Dissertação de Mestrado. Universidade Tecnológica Federal do Paraná.

SIQUEIRA, Hugo et al. Double-Swarm Binary Particle Swarm Optimization. In: 2018 IEEE Congress on Evolutionary Computation (CEC). IEEE, 2018. p. 1-8.

ÁVILA, Sérgio Luciano et al. Algoritmos genéticos aplicados na otimização de antenas refletoras. 2002.

SANTANA, Clodomir J. et al. SBFSS: Simplified Binary Fish School Search. In: 2019 IEEE Congress on Evolutionary Computation (CEC). IEEE, 2019. p. 2595-2602.