



# UMA PROPOSTA DE METODOLOGIA PARA O ENSINO: UM SIMULADOR 3D DO ALGORITMO DE OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS

Diogo Xavier de Noronha<sup>(1)</sup>, Josimar Alves Pereira<sup>(1)</sup>, Ciniro Aparecido Leite Nametala<sup>(1)</sup>

<sup>(1)</sup> Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais (IFMG) - Campus Bambuí

dexisene@hotmail.com, josimaar.alves@gmail.com, ciniro.nametala@ifmg.edu.br

## RESUMO

Este trabalho tem como objetivo apresentar um simulador 3D do algoritmo de Otimização por Enxames de Partículas (PSO) elaborado para auxiliar como método didático. O PSO foi concebido com base no comportamento coletivo de agentes simples interagindo entre si e com o ambiente. Para o desenvolvimento do simulador foi utilizado a linguagem *C#* por meio de um *game engine*. A interface da ferramenta permite que o usuário, como educador, desperte a atenção dos alunos para ter o máximo de aproveitamento do *software* como ferramenta didática, a fim de lecionar sobre o algoritmo PSO e sua eficácia como ferramenta de otimização.

**Palavras-chave:** Otimização. Otimização por Enxame de Partículas. Aprendizagem baseada em simulação.

## 1 INTRODUÇÃO

Nessa era digital, um grande desafio para os professores é obter ferramentas capazes de envolver e despertar curiosidade dos alunos. É fundamental que o educando aprenda para questionar, e é justamente este questionamento que faz romper os próximos paradigmas e construir novas teorias.

Santos (2009), aponta a evidência de que ao usar o computador em aulas de matemática os alunos retêm de forma mais concreta o conhecimento e defende que as Tecnologias da Informação e Comunicação (TIC's) deram um novo suporte ao ato de ensinar e aprender, e estão cada vez mais presentes no cotidiano de alunos e professores.

De acordo com Morán (2015), a educação formal está num impasse diante de muitas mudanças na sociedade, tal como evoluir para tornar-se relevante e conseguir que todos aprendam de forma competente a conhecer, a construir seus projetos de vida e a conviver com os demais.

Atividades desenvolvidas com o propósito de ensinar deve ser apreciada por todos aqueles que participam (ALVES, 2011). Neste sentido, Mitre et al. (2008), descreve que a produção de novos saberes exige o entendimento de que a mudança é possível, o exercício da intuição, da curiosidade, da emoção e da capacidade crítica de observar e perseguir o objetivo de aprender.

Almeida (2010), afirma que os métodos tradicionais de ensino eram essenciais quando era difícil o acesso à informação. Com a Internet e a divulgação aberta de conteúdos, pode-se

aprender em qualquer lugar, a qualquer hora e com muitas pessoas diferentes. Assim sendo, é complexo, necessário e até mesmo um pouco assustador, pois não há modelos prévios bem sucedidos para aprender de forma flexível numa sociedade altamente conectada.

Seguindo esta linha de raciocínio, este trabalho tem como objetivo a criação de um objeto de aprendizagem que auxilie o professor em sala de aula e aos alunos no aprendizado, especificamente a otimização por enxame de partículas.

## 2 METODOLOGIA

A pesquisa deste trabalho pode ser classificada como uma pesquisa qualitativa, exploratória e aplicada, que tem por objetivo produzir conhecimento para uma aplicação prática, contribuindo para o ensino-aprendizagem por meio de uma simulação do algoritmo PSO.

Para o desenvolvimento foi utilizado um *game engine* chamado *Unity 3D*. Trata-se de um motor de jogo, ou *game engine* que fornece funcionalidades como renderização de gráficos, detecção de colisão, suporte para criação de animações, efeitos sonoros e suporte a linguagens como *JavaScript* e *C#*. Neste trabalho optou-se pelo uso de *C#* devido a facilidade oferecida pelas características da linguagem orientada a objetos e pelo material de estudos, que é mais completo e abrangente.

O objeto 3D texturizado e animado que pode ser visto na Figura 1, utilizado como agente na simulação, foi adquirido por meio do *website* TurboSquid, que o disponibiliza gratuitamente. Este objeto após instanciado torna-se apenas uma representação visual de um *GameObject* que é instanciado para cada partícula do PSO.

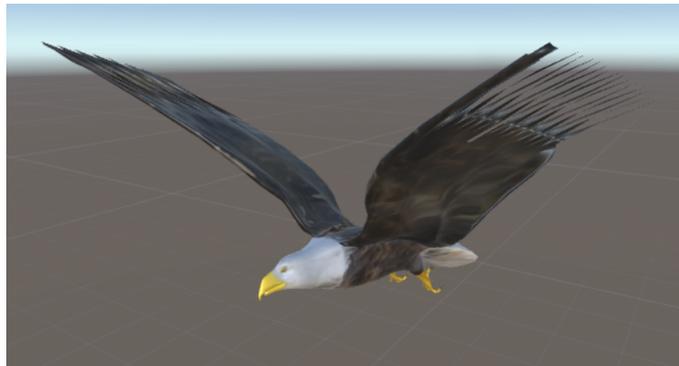


Figura 1 – Objeto 3D.

O código implementado conta com 2 classes principais:

- 1. AveScript:** Responsável por controlar cada partícula do PSO. Ela conta com atributos chave para a solução do problema, tais como: posição, velocidade, constante de inércia, fator cognitivo (*pbest*) e fator social (*gbest*). O fator social difere dos demais, pois é um atributo estático com a finalidade de ser compartilhado com todas as partículas. Dessa maneira, não se faz necessária a comunicação entre as partículas.

- 2. GeneralController:** Responsável por iniciar a simulação instanciando as partículas no espaço, assim como pausar ou finalizar a simulação. É responsável por pegar os dados

inseridos pelo usuário para calibrar o PSO e calcular a imagem da função na posição de cada partícula a cada iteração do algoritmo.

### 3 RESULTADOS E DISCUSSÃO

O *software* desenvolvido conta com uma interface que permite a usuário informar dados tais como constante de inércia, modificador do fator cognitivo, modificador do fator social e o número de partículas. Além disso, informa constantemente o valor da posição ótima encontrada a cada iteração. Uma imagem da interface pode ser vista na Figura 2.



Figura 2 – Interface Gráfica

Também é permitido ao usuário escolher entre três funções a serem otimizadas. A primeira, linear (1), tem  $Y$  igual a soma dos módulos dos eixos  $X$  e  $Z$ , a segunda é a soma dos quadrados de  $X$  e  $Y$  (2), e a terceira é uma função senoidal (3) bidimensional somada do módulo de  $X$  e  $Y$ . O domínio em todas as três funções é fixo e limitado em  $-100 < X < 100$  e  $-100 < Y < 100$ . Os resultados de todas essas equações são divididos ou multiplicados por valores ideais para facilitar a visualização, como pode-se observar na Figura 3, mas que não têm influência significativa na execução do algoritmo. Os resultados das representações visuais em *Unity* dessas funções podem ser observados nas Figuras 3, 4 e 5, respectivamente.

$$y = \frac{|x| + |z|}{5}. \quad (1)$$

$$y = \frac{x^2 + z^2}{100}. \quad (2)$$

$$y = 5 * \sin\left(\frac{x}{5}\right) + 5 * \sin\left(\frac{x}{5}\right) + \frac{|x|}{10} + \frac{|y|}{10} \quad (3)$$

O *software* ainda conta com uma câmera controlável, que proporciona ao usuário a possibilidade de movê-la pelo espaço e ter o ângulo e a distância de visão desejada. Espera-se com isso, que se possa ter o máximo de aproveitamento do *software* como ferramenta didática.

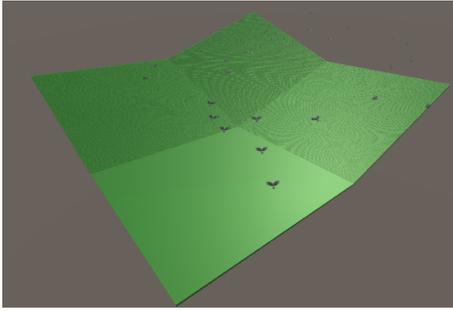


Figura 3 – Função Linear

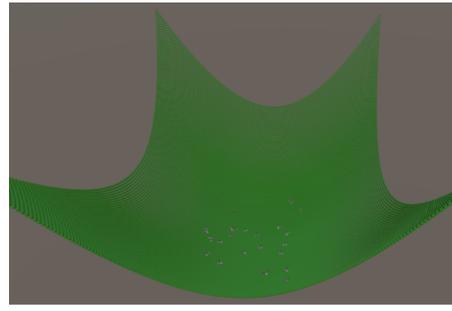


Figura 4 – Função Esfera

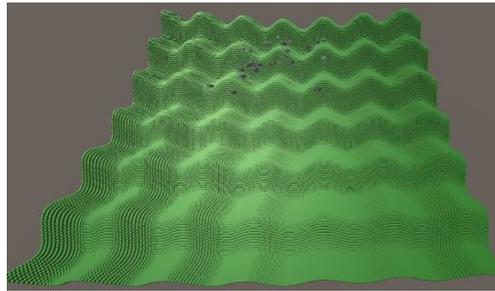


Figura 5 – Função Senoidal.

O *software* utiliza os dados fornecidos pelo usuário para calibrar o PSO e buscar pelo valor mínimo da imagem da função. Enquanto cada partícula procura se mover na direção do mínimo parcial global, chamado de *gbest*, novos valores são encontrados e estes novos valores se tornam uma nova direção para o movimento das partículas. Desta maneira o *gbest* muda sempre que um valor menor de imagem da função é encontrado por qualquer partícula. O resultado final é atingido quando todas as partículas convergem em um único ponto. A Figura 6 mostra o comportamento do *gbest* ao longo das iterações do algoritmo sobre a função esfera.

Durante esta busca por um ponto ótimo além do chamado fator social (*gbest*) que indica o ponto mínimo encontrado por todas as partículas, cada uma delas armazena o seu menor valor alcançado, este é chamado de fator cognitivo (*pbest*). Caso uma partícula encontre um fator cognitivo menor que o fator social (*gbest*), aquele valor se torna o novo fator social e todas as partículas recebem o novo valor. A Figura 7 mostra a média de todos os *pbest*'s a cada iteração do algoritmo executado sobre uma função senoidal, utilizando uma população de 30 indivíduos com constante de inércia igual a 0.8, modificador cognitivo igual a 0.6 e modificador social igual a 2. Para melhor visualização do gráfico estes dados foram normalizados.

No gráfico da Figura 6, observa-se que o algoritmo encontrou o ponto mínimo da função e no gráfico da Figura 7, fica evidente que todas as partículas convergiram para o ponto ótimo. Dessa forma, conclui-se que o *software* obteve os resultados esperados nas otimizações de cada função.

#### 4 CONCLUSÃO

São consideráveis os ganhos que se tem ao utilizar simuladores em auxílio do professor na sala de aula. Para se ensinar algoritmos naturais isso se torna mais notável, devido a sua

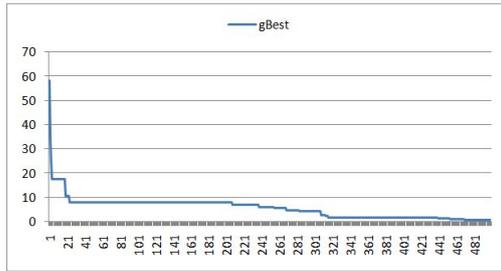


Figura 6 – (a) Comportamento do gBest.

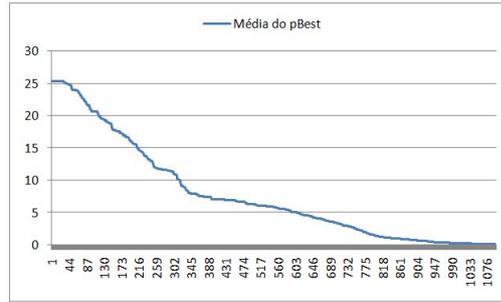


Figura 7 – Média do comportamento do pBest.

origem, baseada em elementos reais da natureza, e devido a sua semelhança na execução que é tão facilmente observável.

Através deste trabalho cria-se a oportunidade para que em trabalhos futuros outros simuladores sejam implementados da mesma maneira, para que se tenha uma maior abstração da concepção e do funcionamento de cada algoritmo natural.

A implementação está disponível em: <[https://github.com/dexisene/PSO\\_Unity](https://github.com/dexisene/PSO_Unity)>

## REFERÊNCIAS

- ALMEIDA, M. E. B. de. Integração de currículo e tecnologias: a emergência de web currículo. 2010.
- ALVES, R. **Conversas sobre educação**. [S.l.]: Verus Editora, 2011.
- MITRE, S. M. et al. Metodologias ativas de ensino-aprendizagem na formação profissional em saúde: debates atuais. **Ciência & saúde coletiva**, SciELO Brasil, v. 13, n. 2, p. 2133–2144, 2008.
- MORÁN, J. Mudando a educação com metodologias ativas. **Coleção Mídias Contemporâneas. Convergências Midiáticas, Educação e Cidadania: aproximações jovens**, v. 2, p. 15–33, 2015.
- SANTOS, J. J. A. dos. Objetos de aprendizagem e o ensino de matemática análise de sua importância na aprendizagem de conceitos de probabilidade. 2009.