



UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO  
OTIMIZAÇÃO COMBINATÓRIA E META-HEURÍSTICAS

2019/2

1º TRABALHO

Marcos A. Spalenza

**PSO: *Particle Swarm Optimization***

Vitória, ES

2019

# 1 Introdução

*Particle Swarm Optimization* - *PSO* é uma meta-heurística baseada no comportamento de coletivo de animais. Este tipo de comportamento biológico é modelado através da *Swarm Intelligence*, subárea da Computação Evolutiva. A *Swarm Intelligence* caracteriza-se pela modelagem descentralizada de indivíduos de modo autoorganizado, seja ele natural ou artificial (ZHANG; WANG; JI, 2015).

Esse conjunto de técnicas bio-inspiradas foram criadas diretamente da modelagem computacional evolutiva do comportamento coletivo animal. Assim como o *PSO*, outras diversas seguem o mesmo princípio para criar eficientes modelos de otimização combinatórios. Enquanto o espaço de busca apresenta-se como inviável para buscas exaustivas, fica a cargo das heurísticas e meta-heurísticas definirem como encontrar valores que atendam à função objetivo.

Assim, o *PSO* é responsável pela busca em *swarm*, ou seja, pela distribuição de indivíduos no hiperespaço. Nesse formato, os indivíduos se distribuem no espaço de busca, procurando coesão com a melhor solução alcançada até o momento. De fato, essa coesão é representada pelo direcionamento de todas as partículas ao melhor resultado, enquanto continuam as tentativas por melhorias (MARINI; WALCZAK, 2015). Desta forma, cada partícula adapta sua posição e seu local com base na sua melhor posição e na melhor posição alcançada pelo *swarm*.

Assim, o trabalho a seguir é apresentado em 5 seções. Na Seção 2 apresentamos um breve histórico do *PSO*. Na Seção 3 definimos as principais características dessa técnica e descrevemos a sua estrutura. Na Seção 4 apresentamos as técnicas que são o estado da arte da literatura. Por fim, na Seção 5 descrevemos algumas aplicações e alguns problemas que foram tratados por esta técnica.

## 2 Histórico

Conforme descreve (SENGUPTA; BASAK; PETERS, 2019), a *Swarm Intelligence* teve diferentes propósitos antes dos modelos de otimização. Geralmente permeando a área de modelagem de dados, um dos primeiros trabalhos relativos a análises das partículas foi apresentado por (REEVES, 1983). Nele, são apresentados modelos de enxame para caracterização de objetos particulados em imagem, como luz, fogo, água ou nuvens. Após foi descrita a interação entre essas mesmas partículas por (REYNOLDS, 1987), chamado de *Boid Model*. Através deste foram descritos métodos de interação entre partículas aumentando sua autonomia.

Anos mais tarde, foi apresentado o *PSO* (EBERHART; KENNEDY, 1995), representando certos avanços ao observar indivíduos de forma mais complexa do que algoritmos

como o *Ant Colony*. Diferentemente dos demais sistemas de *swarm*, o PSO adiciona certa independência dos indivíduos enquanto parte do coletivo, direcionando-os para pontos considerados até o momento estratégicos. Inspirado nos grupos de pássaros voando e os cardumes de peixes, o algoritmo dispõe características como a dualidade entre independência e relação social, o que torna seu indivíduo de certa forma mais complexo. Portanto, o modelo apresentado pelos autores salienta a liberdade da partícula de seguir para um local de forma independente, enquanto tende à seguir ao encontro da melhor solução.

### 3 Definição e Descrição

O PSO, é um algoritmo inspirado nas formas naturais de enxame que são traduzidas em uma estrutura de controle independente de uma população de partículas. Cada partícula individualmente corresponde a um indivíduo no enxame que se adapta conforme as informações atuais junto com cunho social colaborativo (MARINI; WALCZAK, 2015). Portanto, para a função objetivo o estado do indivíduo e do conglomerado impactam na inércia do movimento das partículas. Esse comportamento das partículas e sua convergência pode ser visualizado na Figura 1.

Como podemos ver no problema simples representado na Figura 1, a tendência das partículas é a convergência para alcançar os melhores resultados em conjunto. Porém, para isso cada indivíduo explora seu posicionamento enquanto incide sobre ele a influência social. Portanto, este modelo pode ser visto segundo os aspectos fundamentais para o PSO:

- a população inicial :  $N$
- a função objetivo :  $f(x_i)$
- a posição da partícula :  $x_i$
- a velocidade da partícula :  $v(x_i)$
- a inércia :  $w(x_i)$
- a influência individual :  $C_1$
- a influencia social :  $C_2$

Desta forma, são geradas  $N$  partículas em *swarm*, para otimizar essa função objetivo  $f(x_i)$ . A população usualmente é gerada randomicamente, para evitar o enviesamento da busca. Para cada partícula, então, estabelecemos uma velocidade e uma posição inicial. Dado um tempo  $t + 1$ , reposicionando as partículas, evidencia-se as características de influência. A velocidade anterior, através da inércia associa-se à pesos  $r_1$  e  $r_2$  para avaliar

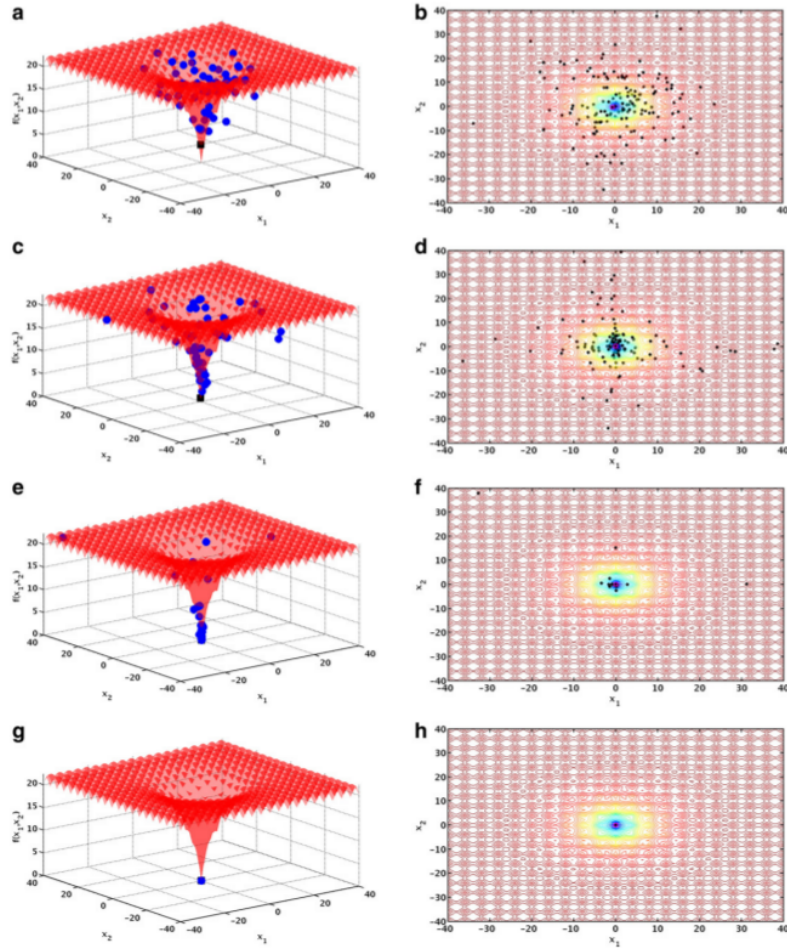


Figura 1 – Representação em um problema simples do comportamento de swarm.

a modificação no deslocamento. Os pesos indicam respectivamente graus de alteração dado pelo coeficiente individual do melhor resultado alcançado pela partícula  $C_1$  e pelo *swarm*  $C_2$ . O método de readaptação de cada partícula em meio a essas variáveis é apresentado na Figura 2.

Como a Figura 2 deixa explicitado, cada partícula tem seu reposicionamento independente conforme sua posição  $x$  e sua velocidade. Porém, pelo comportamento social adaptativo, o reposicionamento de todas as partículas depende diretamente da melhor solução global no determinado tempo  $t$ . Isso visa garantir que todas as partículas sigam o viés da melhor solução momentânea como um fator de liderança do *swarm* em busca de uma melhoria. Dentre os itens listados,  $C_1$  e  $C_2$  e  $N$  são os parâmetros de entrada junto com a função objetivo  $f(x_i)$  e a condição de parada. Esta última, por exemplo, pode ser dada pelo número de iterações do algoritmo. Assim, o algoritmo segue o ciclo representado pelo pseudo-código da Figura 3.

Como vemos pela Figura 3, podemos dividir em quatro partes a meta-heurística: a criação da população, a avaliação da função objetivo de cada partícula, o loop de atualização e a condição de parada. Seguindo o mesmo padrão dos algoritmos evolucionários,

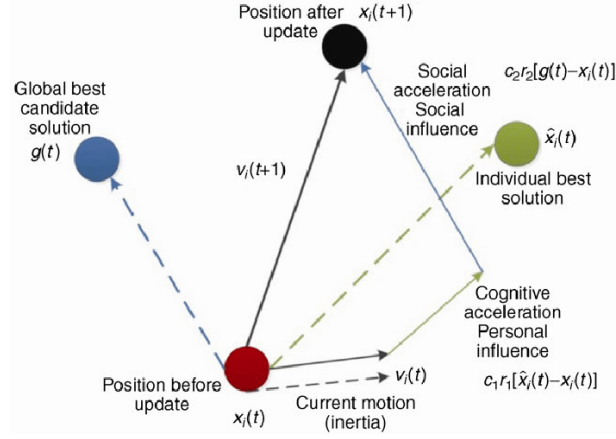


Figura 2 – Readaptação de cada partícula do PSO com base na melhor solução por indivíduo e por *swarm*.

```

Step 1. Initialization
For each particle  $i = 1, \dots, N_p$ , do
  (a) Initialize the particle's position with a uniformly distribution as  $P_i(0) \sim U(LB, UB)$ , where LB and UB represent the lower and upper bounds of the search space
  (b) Initialize  $pbest$  to its initial position:  $pbest(i, 0) = P_i(0)$ .
  (c) Initialize  $gbest$  to the minimal value of the swarm:  $gbest(0) = \operatorname{argmin} f[P_i(0)]$ .
  (d) Initialize velocity:  $V_i \sim U(-|UB - LB|, |UB - LB|)$ .
Step 2. Repeat until a termination criteria is met
For each particle  $i = 1, \dots, N_p$ , do
  (a) Pick random numbers:  $r_1, r_2 \sim U(0, 1)$ .
  (b) Update particle's velocity. See formula (2).
  (c) Update particle's position. See formula (3).
  (d) If  $f[P_i(t)] < f[pbest(i, t)]$ , do
    (i) Update the best known position of particle  $i$ :  $pbest(i, t) = P_i(t)$ .
    (ii) If  $f[P_i(t)] < f[gbest(t)]$ , update the swarm's best known position:  $gbest(t) = P_i(t)$ .
  (e)  $t \leftarrow (t + 1)$ ;
Step 3. Output  $gbest(t)$  that holds the best found solution.

```

Figura 3 – Pseudo-Código básico de um PSO.

o PSO têm como diferença o constante reposicionamento das partículas pelos fatores da inércia, comportamento individual e comportamento global. Para além disso, as populações geralmente são iniciadas randomicamente, se não houver um conhecimento prévio do hiperespaço estudado. A função objetivo  $f(x_i)$  avalia a qualidade de cada uma dessas soluções e as ranqueia, mantendo o controle da melhor solução. O loop de atualização refere-se ao cerne da proposta, mantendo o deslocamento das partículas até a determinada função de parada, geralmente associada à um determinado número de ciclos.

## 4 Estado-da-Arte

Observando a literatura podemos ver que a capacidade de adaptação e a simplicidade do algoritmo de PSO representa uma abertura para seu uso em várias aplicações, modelos e melhorias. Dado os diferentes híbridos encontrados, podemos encontrar combinações entre modelos com aplicação conjunta. Para além disso, a topologia do modelo, a forma

de distribuição do *swarm*, e as diferentes inspirações em sistemas naturais ou não, indicam propostas inovadoras para o PSO clássico. Assim, como podemos ver nas recentes propostas híbridas a combinação de meta-heurísticas e a modificação do *swarm* são os principais tópicos abordados recentemente (SENGUPTA; BASAK; PETERS, 2019).

Dentre os híbridos, podemos destacar a mistura entre os modelos de Algoritmos Genéticos (PSO-GA) (LI et al., 2018), Evolução Diferencial (DEPSO-Scout) (BOONSERM; SITJONGSATAPORN, 2017), *Simulated Annealing* (SA-PSO) (LI et al., 2017) e *Ant Colony* (HOA) (LU et al., 2016) com o PSO. Destacamos ainda a forma como estes híbridos interagem, como por exemplo o PSO-GA que controla a participação mútua dos algoritmos. Nessa abordagem, o PSO é responsável pela modificação dos indivíduos ao tempo que o Algoritmo Genético é o responsável pelo controle da população dentre as iterações.

Por outro lado, algoritmos podem delimitar escopos diferentes de busca. HOA por exemplo, é uma proposta de combinação entre o PSO e o *Ant Colony*. Durante a convergência prolongada os resultados tornam-se responsabilidade deste primeiro enquanto as convergências prematuras tornam-se responsabilidade do segundo.

## 5 Exemplos de Aplicações

PSO é uma meta-heurística bem versátil, do mesmo modo que ela é frequentemente parte de algoritmos híbridos, também apresenta inúmeros casos de uso. Um exemplo de uso é o *Vehicle Routing Problem with Simultaneous Pickup and Delivery - VRPSPD* (AI; KACHITVICHYANUKUL, 2009). Um pouco mais específico que o problema clássico de roteamento, a meta-heurística fica aqui responsável pelo alocamento dos veículos para determinadas rotas, resguardando-se de uma área de cobertura para o mesmo, evitando o alocamento por demanda. Assim, nesse caso estudado 5 regras devem ser atendidas:

- Cada rota começa e termina no depósito.
- Cada cliente só pode ser visitado apenas uma vez por um veículo.
- A carga total do veículo não pode exceder a capacidade.
- A duração total de cada rota, entre viagem e serviço, não pode exceder o limite.
- O custo total deve ser minimizado.

Essas regras visam garantir que haja eficiência no alocamento dos itens sem extrapolar a capacidade de cada envolvido. Para avaliação o autor compara seu método com outros 3 trabalhos e seus respectivos dados. Dentre esses dados, diferentes números de instâncias desafiam a adaptabilidade do PSO. Nos resultados obtidos o algoritmo apresenta maior eficiência quando comparado com os *benchmarks*, sendo superado em poucas vezes

em, no máximo, 6% de diferença. A melhoria então ocorre em quase todo dataset, restando poucas amostras ao qual o PSO não representa uma melhoria, mas chega próximo ao resultado.

Outra aplicação bem distinta desta é a seleção de características. Em meio a classificação em alta dimensionalidade, a seleção de características espera definir os itens mais relevantes para melhorar o processo de categorização. Nesse caso, os autores aplicaram o PSO para seleção dos principais erros linguísticos referentes ao método avaliativo da Competência 1 do Exame Nacional do Ensino Médio - ENEM (JÚNIOR; SPALENZA; OLIVEIRA, 2017). O objetivo portanto é minimizar o erro do classificador automático nas 954 redações quando comparadas com as notas atribuídas pelo avaliador humano.

Dentre as características desses documentos, o algoritmo observou a quantidade de parágrafos, frases, palavras, caracteres, vírgulas, pontos, pontos de interrogação e exclamação somados com a identificação das 18 principais classes gramaticais da língua portuguesa. Além disso, as principais características incluem os erros ortográficos e os 33 tipos frequentes de erros gramaticais. Nesse modelo, a população foi gerada aleatoriamente com vetores binários que correspondem a manutenção ou não de cada característica, visando a redução de dimensionalidade e aumento da consistência da classificação. Considerando uma flexibilidade de 0.3 pontos, comum da avaliação entre especialistas, a acurácia do sistema foi de 72% enquanto a literatura apresentava cerca de 52% até o momento.

Deste modo, vários exemplos distintos são encontrados na literatura com aplicações do PSO (ZHANG; WANG; JI, 2015). Apesar dos hiperparâmetros empíricos, este algoritmo se destaca pela adaptabilidade entre os problemas, a qualidade dos resultados adquiridos de forma relativamente simples. Portanto essa meta-heurística ainda representa pouca tendência de ficar preso em mínimos locais, com grande capacidade de migrar no hiperespaço de busca.

## Referências

- AI, T. J.; KACHITVICHYANUKUL, V. A Particle Swarm Optimization for the Vehicle Routing Problem with Simultaneous Pickup and Delivery. *Computers & Operations Research*, Elsevier, v. 36, n. 5, p. 1693–1702, 2009. Citado na página 5.
- BOONSERM, P.; SITJONGSATAPORN, S. A Robust and Efficient Algorithm for Numerical Optimization Problem: DEPSO-Scout: A New Hybrid Algorithm Based on DEPSO and ABC. In: IEEE. *2017 International Electrical Engineering Congress (iEECON)*. [S.l.], 2017. p. 1–4. Citado na página 5.
- EBERHART, R.; KENNEDY, J. A New Optimizer Using Particle Swarm Theory. In: IEEE. *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*. [S.l.], 1995. p. 39–43. Citado na página 1.
- JÚNIOR, C. R.; SPALENZA, M. A.; OLIVEIRA, E. de. Proposta de um Sistema de Avaliação Automática de Redações do ENEM Utilizando Técnicas de Aprendizagem de Máquina e Processamento de Linguagem Natural. *Anais do Computer on the Beach*, p. 474–483, 2017. Citado na página 6.
- LI, C. et al. Optimization of a Heliostat Field Layout Using Hybrid PSO-GA Algorithm. *Applied Thermal Engineering*, v. 128, p. 33 – 41, 2018. Citado na página 5.
- LI, P. et al. Energy Management of a Parallel Plug-in Hybrid Electric Vehicle Based on SA-PSO Algorithm. In: IEEE. *2017 36th Chinese Control Conference (CCC)*. [S.l.], 2017. p. 9220–9225. Citado na página 5.
- LU, J. et al. A Hybrid Algorithm Based on Particle Swarm Optimization and Ant Colony Optimization Algorithm. In: SPRINGER. *International Conference on Smart Computing and Communication*. [S.l.], 2016. p. 22–31. Citado na página 5.
- MARINI, F.; WALCZAK, B. Particle Swarm Optimization (PSO). A Tutorial. *Chemometrics and Intelligent Laboratory Systems*, Elsevier, v. 149, p. 153–165, 2015. Citado 2 vezes nas páginas 1 e 2.
- REEVES, W. T. Particle Systems — A Technique for Modeling a Class of Fuzzy Objects. *ACM Transactions On Graphics (TOG)*, ACM, v. 2, n. 2, p. 91–108, 1983. Citado na página 1.
- REYNOLDS, C. W. *Flocks, Herds and Schools: A Distributed Behavioral Model*. [S.l.]: ACM, 1987. Citado na página 1.
- SENGUPTA, S.; BASAK, S.; PETERS, R. A. Particle Swarm Optimization: A Survey of Historical and Recent Developments with Hybridization Perspectives. *Machine Learning and Knowledge Extraction*, Multidisciplinary Digital Publishing Institute, v. 1, n. 1, p. 157–191, 2019. Citado 2 vezes nas páginas 1 e 5.
- ZHANG, Y.; WANG, S.; JI, G. A Comprehensive Survey on Particle Swarm Optimization Algorithm and its Applications. *Mathematical Problems in Engineering*, Hindawi, v. 2015, 2015. Citado 2 vezes nas páginas 1 e 6.