

Particle Swarm Optimization

PSO



UFES

Marcos A. Spalenza

Doutorando em Ciência da Computação
Laboratório de Computação de Alto Desempenho - LCAD
Programa de Pós-Graduação em Informática - PPGI

Apresentação

- Introdução
- Histórico
- Definição e Descrição
- Estado da Arte
- Exemplo de Aplicação

Introdução

Particle Swarm Optimization - PSO é uma técnica de *Swarm Intelligence* biologicamente inspirada baseada no comportamento coletivo.

A *Swarm intelligence* é um campo de estudo dentro da *Computação Evolutiva* (Sengupta et. al., 2019) ao qual destaca-se como um dos primeiros trabalhos a modelagem de objetos no campo de animação em *Computer-Generated Imagery - CGI* da *LucasFilm Ltd* (Reeves, 1983).

REEVES, William T. Particle Systems — A Technique for Modeling a Class of Fuzzy Objects. **ACM Transactions On Graphics (TOG)**, v. 2, n. 2, p. 91-108, 1983

SENGUPTA, Saptarshi; BASAK, Sanchita; PETERS, Richard Alan. Particle Swarm Optimization: A survey of historical and recent developments with hybridization perspectives. **Machine Learning and Knowledge Extraction**, v. 1, n. 1, p. 157-191, 2019.

Histórico

Swarm Intelligence caracteriza-se pelo estudo de modelos de comportamento coletivo descentralizado, auto-organizado, natural ou artificial (Zhang et.al. 2015).

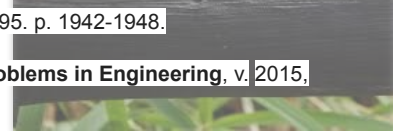
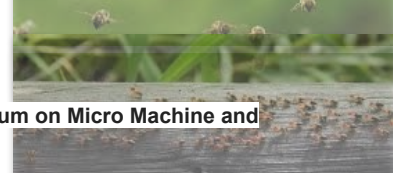
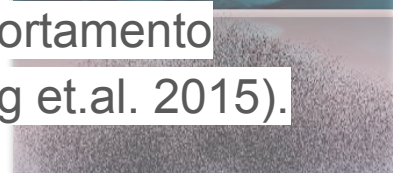
Além do PSO enquadra-se neste tópico o *Ant Colony Optimization - ACO* e o *Bacterial Foraging Optimization - BFO*.

O PSO (Eberhart and Kennedy, 1995) origina-se da observação de agrupamentos de pássaros e peixes, caracterizando computacionalmente suas relações individuais e sociais.

EBERHART, Russell; KENNEDY, James. *A New Optimizer Using Particle Swarm Theory*. In: **MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science**. IEEE, 1995. p. 39-43.

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

ZHANG, Yudong; WANG, Shuihua; JI, Genlin. *A Comprehensive Survey on Particle Swarm Optimization Algorithm and its Applications*. **Mathematical Problems in Engineering**, v. 2015, 2015.



Definição

O PSO como algoritmo simula a independência de cada partícula (indivíduo) enquanto parte de uma estrutura social.

Cada partícula tem suas características independentes de busca na função objetivo mas é impactado diretamente pelo aspecto social.

São aspectos relevantes dentro do PSO a população, a posição e a velocidade de cada partícula, a inércia e as influências individual e social.



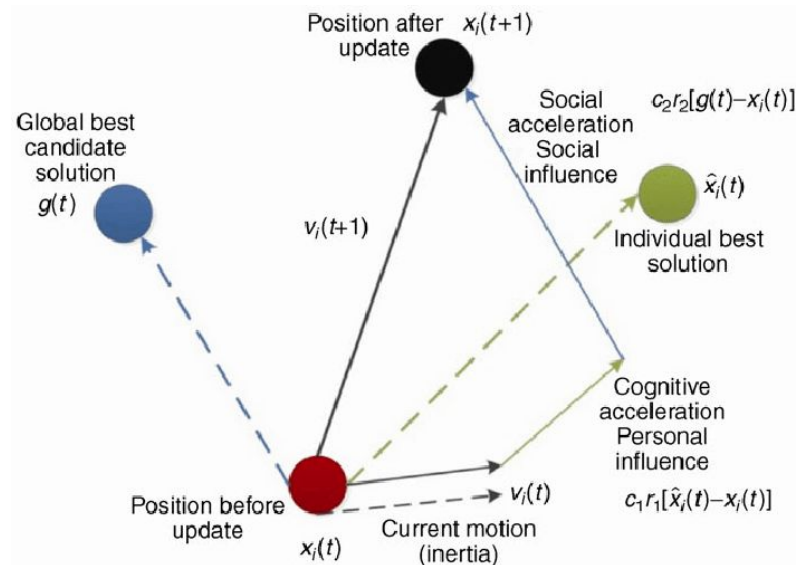
Finding Nemo - Pixar Animation. 2003

Descrição

Em um determinado tempo t :

Cada partícula i têm definida uma posição x e uma velocidade v .

A velocidade atual indica um valor de inércia que associada às influências pessoal $c1$ e social $c2$ indicam a próxima velocidade da partícula, ponderadas cada qual por um coeficiente entre 0 e 1 aleatório ($r1$ e $r2$ respectivamente).



Atualização das partículas durante cada iteração do algoritmo.
Implementing the Particle Swarm Optimization (PSO) Algorithm in Python

<https://medium.com/analytics-vidhya/>

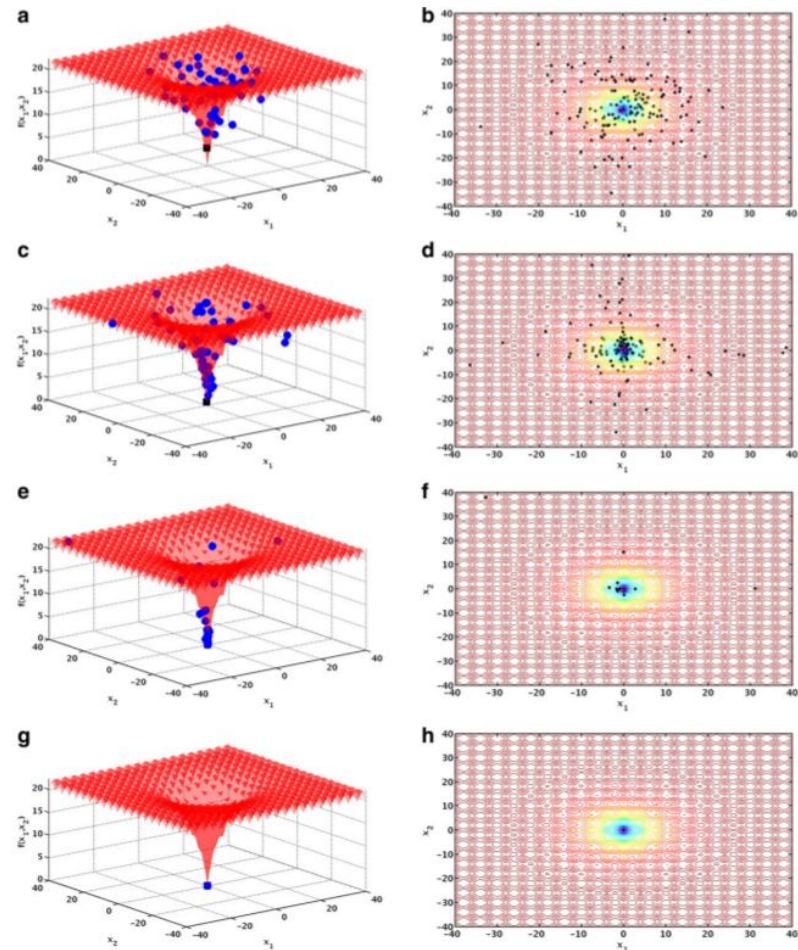
```

FOR each particle  $i$ 
  FOR each dimension  $d$ 
    Initialize position  $x_{id}$  randomly within permissible range
    Initialize velocity  $v_{id}$  randomly within permissible range
  End FOR
END FOR
Iteration  $k=1$ 
DO
  FOR each particle  $i$ 
    Calculate fitness value
    IF the fitness value is better than  $p\_best_{id}$  in history
      Set current fitness value as the  $p\_best_{id}$ 
    END IF
  END FOR
  Choose the particle having the best fitness value as the  $g\_best_d$ 
  FOR each particle  $i$ 
    FOR each dimension  $d$ 
      Calculate velocity according to the equation
      
$$v_{id}(k+1) = w v_{id}(k) + c_1 \text{rand}_1(p_{id}-x_{id}) + c_2 \text{rand}_2(p_{gd}-x_{id})$$

      Update particle position according to the equation
      
$$x_{id}(k+1) = x_{id}(k) + v_{id}(k+1)$$

    END FOR
  END FOR
   $k=k+1$ 
WHILE maximum iterations or minimum error criteria are not attained

```



Estado da Arte

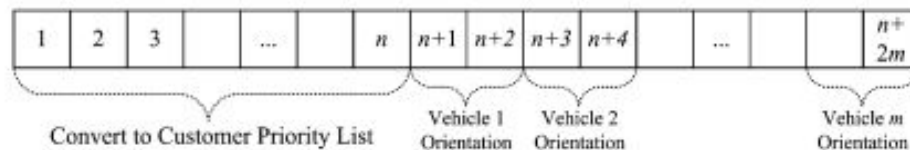
Múltiplas topologias, abordagens e combinações determinam melhorias em potencial para o PSO, cada qual com um detalhe. Dentre elas destacam-se:

Híbridos no estado da arte:

- Algoritmos Genéticos: PSO-GA (Li et.al. 2018)
- Evolução Diferencial: DEPSO-Scout (Boonserm and Sitjongsataporn, 2017)
- Simulated Annealing SA-PSO (Li et.al. 2017)
- Ant Colony : HOA (Junliang et.al. 2017)

Exemplo de Aplicação

AI, The Jin; KACHITVICHYANUKUL, Voratas. A Particle Swarm Optimization for the Vehicle Routing Problem with Simultaneous Pickup and Delivery. **Computers & Operations Research**, v. 36, n. 5, p. 1693-1702, 2009.



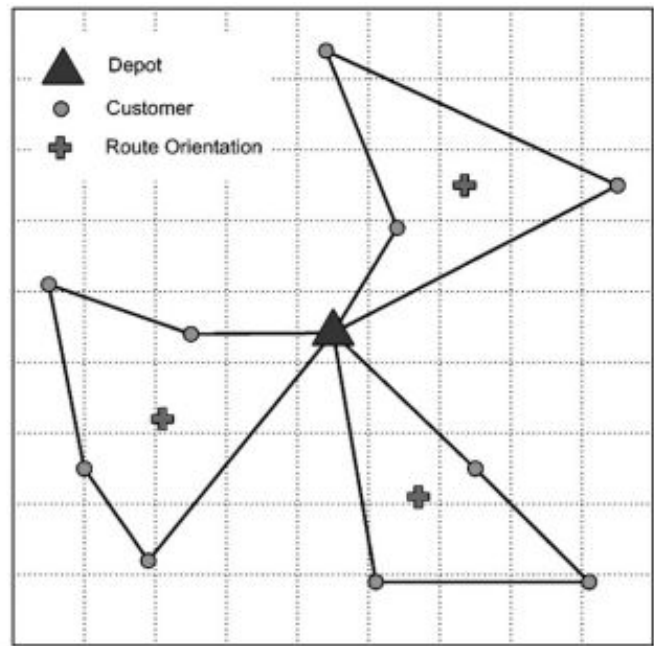
São 5 regras:

- 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 (viagem + serviço) não pode exceder o limite.
- O custo total deve ser minimizado

Exemplo de Aplicação

Dentre os dados utilizados coletados de três artigos com diferentes escalas e distâncias, o PSO apresenta maior eficiência quando comparado com os *benchmarks*.

Em apenas algumas poucas amostras dentre todos os dados que o algoritmo não representa uma melhoria. Dentre esses o resultado do PSO reporta apenas pequenos erros de acordo com a solução proposta.



Exemplo de atendimento aos clientes demarcando as rotas prioritárias por veículo (Route Orientation)..

Exemplo de Aplicação

JÚNIOR, Celso RCA; SPALENZA, Marcos A.; DE OLIVEIRA, Elias. 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.

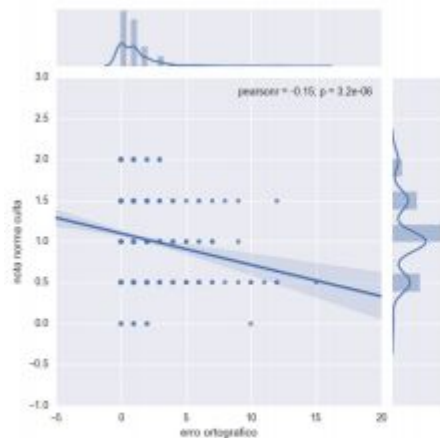
Seleção das características textuais mais relevantes para avaliação automática da Competência 1 do *Exame Nacional do Ensino Médio - ENEM*. O objetivo portanto é minimizar o erro da classificação para 954 redações avaliadas quando comparadas com as notas atribuídas pelo avaliador humano.

Foi avaliado pelo algoritmo a quantidade de parágrafos, frases, palavras, caracteres, erros ortográficos, 33 tipos de erros gramaticais, vírgulas, pontos, pontos de interrogação e exclamação somados com a identificação das 18 principais classes gramaticais da língua portuguesa.

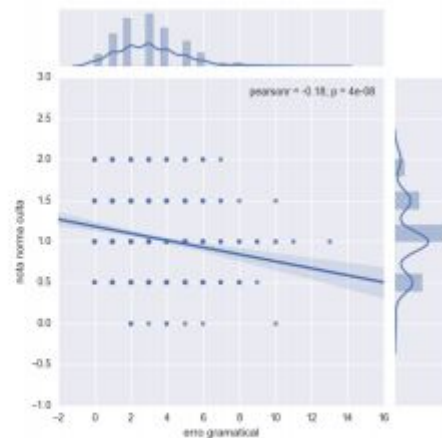
Exemplo de Aplicação

PSO agiu tentando maximizar os resultados de classificação de acordo com a seleção de características representadas através de vetores binários.

Considerando a flexibilização do sistema em 0.3 pontos de nível de erro, a acurácia do sistema foi de 72% enquanto a literatura apresentava cerca de 52% até o momento.



(a) Erros ortográficos



(b) Erros gramaticais

OTIMIZAÇÃO COMBINATÓRIA E META-HEURÍSTICAS



UFES

Marcos A. Spalenza

Doutorando em Ciência da Computação
Laboratório de Computação de Alto Desempenho - LCAD
Programa de Pós-Graduação em Informática - PPGI