

Metaheurísticas

Ant Colony Optimization e Particle Swarm Optimization

Felipe Augusto Lima Reis

felipe.reis@ifmg.edu.br



**INSTITUTO
FEDERAL**
Minas Gerais

Sumário



1 Introdução

2 ACO

3 PSO

INTRODUÇÃO

Swarm Intelligence



- **Swarm Intelligence**¹ trata da aplicação de algoritmos computacionais para reprodução de comportamentos biológico-sociais de animais
 - Analisa e reproduz de comportamentos coletivos de indivíduos, quando estes interagem entre si e com seu ambiente;
 - Lida com sistemas naturais (e artificiais) que consistem em um conjunto de indivíduos que se coordenam usando controle descentralizado e auto-organização [Bozorg-Haddad et al., 2017];

¹Tradução literal: Inteligência de Enxame.

Swarm Intelligence



- São exemplos de organizações biológico-sociais que possuem “inteligência de enxame”: [Bozorg-Haddad et al., 2017]
 - Colméias;
 - Colônias de formigas;
 - Cardumes de peixes;
 - Bandos de pássaros;
 - Rebanhos de animais;
 - etc.

OTIMIZAÇÃO POR COLÔNIA DE FORMIGAS

Introdução

- **Ant Colony Optimization (ACO)** é um tipo de metaheurística construtiva inspirada na trilha de feromônios deixados por formigas, permitindo que outras encontrem alimento
 - Feromônio é um sinal químico liberado que dispara uma resposta natural em outros membros da espécie [Coppin, 2004] [Luzia and Rodrigues, 2009];
 - Esse método de comunicação indireta é denominado estigmergia [Bozorg-Haddad et al., 2017];
- O algoritmo ACO foi criado por Marco Dorigo em 1991 e publicado, posteriormente, em 1996 [Souza, 2011] [Bozorg-Haddad et al., 2017].

Visão Geral



- O algoritmo simula o comportamento de um conjunto de agentes simples (formigas artificiais) que cooperam para resolver um problema de otimização [Souza, 2011];
- As formigas são procedimentos estocásticos (a partir de eventos aleatórios) que constroem uma solução por meio da adição iterativa de componentes a uma solução parcial
 - A colônia move-se de forma concorrente e assíncrona construindo caminhos no espaço de busca [Luzia and Rodrigues, 2009].

Visão Geral



- São utilizados os seguintes procedimentos para adição de componentes à solução:
 - 1 Informações heurísticas sobre a instância do problema (se disponível);
 - 2 Trilhas de feromônio artificiais que mudam dinamicamente, em tempo de execução, para refletir a experiência de busca adquirida pelos agentes [Luzia and Rodrigues, 2009] [Souza, 2011].

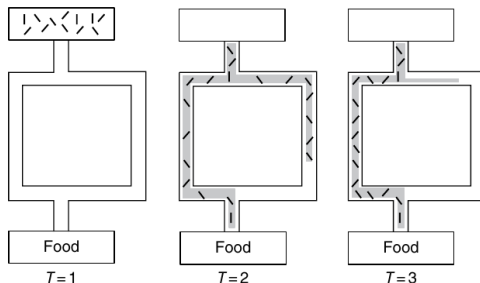
Inspiração Biológica



- Possui correlação com os experimentos de Deneubourg et al. e Goss et al., que estudavam como formigas seguiam trilhas de feromônios;
- No experimento, um ninho de uma colônia de formigas foi conectado a uma fonte de alimento usando duas pontes
 - Na existência de pontes de um mesmo tamanho, as pontes foram selecionadas arbitrariamente;
 - Na existência de pontes de tamanhos distintos, a concentração de feromônio na menor ponte tornou-se superior à ponte de maior tamanho
 - Com isso, as formigas preferiam utilizavam o menor caminho, aumentando ainda mais a concentração de feromônio, e indicando a melhor solução [Bozorg-Haddad et al., 2017].

Inspiração Biológica

- O experimento de Goss et al., com pontes de tamanhos distintos pode ser representado na figura abaixo:



Fonte: [Bozorg-Haddad et al., 2017]

RASTROS DE FEROMÔNIO

Funcionamento dos Rastros de Feromônio

- O ACO é baseado na comunicação indireta, mediada por rastros de feromônios artificiais
 - Os rastros de feromônio funcionam como uma informação numérica distribuída;
 - As formigas artificiais utilizam esse rastro para construir, probabilisticamente, soluções para um dado problema;
 - Esses rastros refletem as experiências de busca das formigas durante a solução e podem ser utilizados como informação por outras formigas [Souza, 2011] [Luzia and Rodrigues, 2009].

Funcionamento dos Rastros de Feromônio

- A medida em que se movem, as formigas constroem novas soluções para o problema de otimização;
- Durante a construção de uma solução, a formiga avalia a solução (parcial ou completa) e deposita uma trilha de feromônio apenas nas componentes ou conexões usadas durante o caminho [Souza, 2011].

Funcionamento dos Rastros de Feromônio

- O algoritmo ACO contém ainda dois componentes adicionais: a evaporação da trilha de feromônio e ações *daemon*.
- A **evaporação de feromônio** é o processo pelo qual o feromônio decresce ao longo do tempo.
 - A evaporação evita uma convergência prematura do algoritmo em uma região subótima;
 - Também permite que rotas velhas desapareçam gradualmente, permitindo a manutenção da rota de melhor valor [Souza, 2011] [Luzia and Rodrigues, 2009] [Coppin, 2004].

Funcionamento dos Rastros de Feromônio

- *Ações daemon*² podem ser usadas para implementar ações centralizadas, as quais não seriam realizadas pelas formigas tomadas isoladamente .
 - Um exemplo prático, citado por [Souza, 2011], é a adição de uma quantidade extra de feromônio apenas nas componentes usadas pela formiga que construiu a melhor solução;
 - Atualizações de feromônio realizadas por *daemon* são chamadas atualizações de feromônio off-line [Souza, 2011]

²Este procedimento é pouco utilizado.

ALGORITMO

Algoritmo



- O pseudo-algoritmo do ACO pode ser visto abaixo.

```
01: C ← {C1, ..., Cn} componentes
02: t ← número de trilhas para construir de uma só vez
03: f ← <f1,...,fn> feromônios dos componentes
04: Melhor ← nulo
05: repita
06:   P ← t trilhas, construídas por seleção iterativa de componentes baseada nos
   feromônios e nas informações heurísticas
07:   para cada Pi em P faça
08:     se Melhor = nulo ou Qualidade(Pi) > Qualidade (Melhor) então
09:       Melhor ← Pi
10:   atualize f para os componentes baseado na qualidade para cada Pi em P em que
   eles participaram
11:   EvaporarFeromônio()
12:   AçõesDeSegundoPlano()
13: até que Melhor seja a solução ideal ou o tempo tenha se esgotado
14: devolva Melhor
```

Ant Colony Optimization

Fonte: [Luzia and Rodrigues, 2009]

VANTAGENS E DESVANTAGENS

Vantagens e Desvantagens

- Vantagens:
 - A característica estocástica permite que as formigas construam uma grande variedade de soluções diferentes;
 - Flexibilidade: o ACO pode ser adaptado a diversas situações;
 - Capacidade de uso em problemas dinâmicos, cujos valores se alteram durante a ciclo de execução do programa [Luzia and Rodrigues, 2009].
- Desvantagens:
 - Possibilidade de estagnação;
 - Possibilidade de convergência prematura;
 - Alto custo computacional [Luzia and Rodrigues, 2009];

PARTICLE SWARM OPTIMIZATION

Introdução

- **Particle Swarm Optimization (PSO)** é uma metaheurística inspirada no comportamento social exibido por pássaros ou peixes quando se esforçam para chegar a um destino
 - Desenvolvido para simulação de um modelo social simplificado: a simulação de um bando ou revoada de pássaros em torno de um ponto contendo comida ou local para descanso;
- O algoritmo foi proposto por Kennedy e Eberhart em 1995, como método para otimização de funções não lineares [Luzia and Rodrigues, 2009] [Bozorg-Haddad et al., 2017].

Introdução

- O algoritmo PSO é baseado no comportamento social dos pássaros (bando de pássaros)
 - Suponha que um grupo de pássaros esteja procurando comida aleatoriamente em uma área;
 - Considere que os pássaros não sabem onde está a comida;
 - Uma estratégia a ser adotada pelo bando é tentar seguir o pássaro que está mais próximo do alimento;
 - O algoritmo aplica uma estratégia semelhante para solução de problemas de otimização [Bozorg-Haddad et al., 2017].

Particle Swarm Optimization

- No PSO, cada solução única no espaço de decisão é definida como um pássaro e denominada partícula
 - Todas as partículas possuem valores de aptidão, que representam a distância até o alimento;
 - Partículas também possuem uma velocidade, que direcionam o vôo;
 - A melhor partícula é o líder, e outras partículas seguem o líder;
 - As partículas voam pelo espaço de decisão do problema seguindo o líder [Bozorg-Haddad et al., 2017].

Particle Swarm Optimization

- Diferentemente de outros métodos baseados em população (ex. algoritmos genéticos), o PSO não cria novos indivíduos (e descarta indivíduos menos aptos) durante as iterações;
- Uma única população estática é mantida, cujos membros possuem propriedades modificadas (como posição e velocidade) em resposta a novas descobertas sobre o espaço [Luzia and Rodrigues, 2009].

Particle Swarm Optimization

- Cada partícula define sua próxima posição com base em:
 - ① Sua melhor posição individual ocupada até o momento, e
 - ② A melhor posição alcançada no grupo (melhor posição global);
- O algoritmo inicializa com posições e velocidades aleatórias
 - Os valores de aptidão das partículas são calculados;
 - Posições e velocidades são atualizadas para gerar novas soluções a cada iteração [Bozorg-Haddad et al., 2017].

ALGORITMO

Algoritmo



- O PSO pode utilizar a seguinte nomenclatura, para velocidade e posição das partículas: [Luzia and Rodrigues, 2009]
 - $\vec{x} = \langle x_1, x_2, \dots \rangle$: localização da partícula no espaço;
 - $\vec{v} = \langle v_1, v_2, \dots \rangle$: velocidade da partícula³;
 - \vec{x}^* : melhor posição da partícula até o momento;
 - \vec{x}^+ : melhor posição do conjunto de informantes da partícula⁴;
 - $\vec{x}^!$: melhor posição de qualquer partícula até o momento.

³A velocidade da partícula pode ser calculada ainda por $\vec{v}_i = \vec{x}_i^{(t)} - \vec{x}_i^{(t-1)}$.

⁴Pequeno conjunto de partículas escolhidas aleatoriamente a cada iteração, incluindo a própria partícula.

Algoritmo



- O PSO contém diferentes parâmetros que devem ser analisados individualmente: [Luzia and Rodrigues, 2009]
 - α : proporção de \vec{v} a ser mantida;
 - β : proporção de \vec{x}^* a ser mantida;
 - γ : proporção de \vec{x}^+ a ser mantida;
 - δ : proporção de $\vec{x}^!$ a ser mantida;
 - ϵ : fator multiplicativo para a aplicação da velocidade.

Algoritmo

- O pseudo-algoritmo do PSO pode ser visto abaixo.

```
01:  $n \leftarrow$  tamanho do enxame
02:  $\alpha \leftarrow$  proporção de  $\vec{v}$  a ser mantida
03:  $\beta \leftarrow$  proporção de  $\vec{x}^*$  a ser mantida
04:  $\gamma \leftarrow$  proporção de  $\vec{x}^+$  a ser mantida
05:  $\delta \leftarrow$  proporção de  $\vec{x}^!$  a ser mantida
06:  $\epsilon \leftarrow$  fator multiplicativo para a aplicação da velocidade
07:  $P \leftarrow$  conjunto de  $n$  partículas, inicializadas com velocidade e posição aleatórias
08: Melhor  $\leftarrow$  nulo
09: repita
10:   para cada  $p$  em  $P$  faça
11:     se Melhor = nulo ou Qualidade( $p$ ) > Qualidade(Melhor) então
12:       Melhor  $\leftarrow p$ 
13:   para cada  $p$  em  $P$  faça
14:     para cada dimensão  $i$  de  $\vec{x}$  faça
15:        $b \leftarrow$  número aleatório no intervalo  $[0, \beta]$ 
16:        $c \leftarrow$  número aleatório no intervalo  $[0, \gamma]$ 
17:        $d \leftarrow$  número aleatório no intervalo  $[0, \delta]$ 
18:        $v_i \leftarrow \alpha v_i + b(x_i^* - x_i) + c(x_i^+ - x_i) + d(x_i^! - x_i)$ 
19:   para cada  $p$  em  $P$  faça
20:      $\vec{x} \leftarrow \vec{x} + \epsilon \vec{v}$ 
21: até que Melhor seja ideal ou o tempo tenha esgotado
22: devolva Melhor
```

Particle Swarm Optimization

Fonte: [Luzia and Rodrigues, 2009]

VANTAGENS E DESVANTAGENS

Vantagens e Desvantagens

- Vantagens:
 - Fácil de ser implementado;
 - Possui desempenho similar aos algoritmos genéticos;
 - Não requer o cálculo de derivadas na sua implementação;
 - Pode ser implementado usando paralelismo [Luzia and Rodrigues, 2009].
- Desvantagens:
 - Possibilidade de estagnação;
 - Possibilidade de convergência prematura;
 - Alto custo computacional [Luzia and Rodrigues, 2009].

Referências I



Bozorg-Haddad, O., Solgi, M., and Loáiciga, H. A. (2017).
Meta-heuristic and Evolutionary Algorithms for Engineering Optimization.
Wiley Series in Operations Research and Management Science. Wiley, 1 edition.



Coppin, B. (2004).
Artificial intelligence illuminated.
Jones and Bartlett illuminated series. Jones and Bartlett Publishers, 1 edition.



Luzia, L. F. and Rodrigues, M. C. (2009).
Estudo sobre as metaheurísticas.
[Online]; acessado em 22 de Setembro de 2020. Disponível em:
<https://www.ime.usp.br/~gold/cursos/2009/mac5758/LeandroMauricioHeuristica.pdf>.



Souza, M. J. F. (2011).
Inteligência computacional para otimização.
[Online]; acessado em 12 de Maio de 2021. Disponível em: <http://www.decom.ufop.br/prof/marcone/ Disciplinas/InteligenciaComputacional/InteligenciaComputacional.pdf>.