

Fundamentos de Inteligência Artificial [5COP099]



Dr. Sylvio Barbon Junior

Departamento de Computação - UEL

Disciplina Anual

Aula 19

ACO - Ant Colony Optimization

Sumário

- Introdução
- Problema do Caixeiro Viajante

Introdução

- Problemas de Otimização são de grande importância para o mundo atual:
 - Agendamento e escalas de veículos coletivos;
 - Alocação de recursos humanos;
 - Telecomunicações;
- A maioria das abordagens desenvolvidas usam como modelo de teste o problema do caixeiro viajante (*Traveling Salesman Problema - TSP*). Este problema de otimização está caracterizado na área da otimização combinatória (*Combinatorial Optimization - CO*)
- Os problemas de CO são NP-HARD

Introdução

- ACO - Ant Colony Optimization é uma das mais recentes técnicas de otimização;
- O ACO foi criado inspirado no mecanismo de funcionamento real das colônias de formigas. Especificamente, reproduz o comportamento de busca por alimentos.
- O núcleo da ideia é reproduzir o comportamento de comunicação entre as formigas por meio dos rastros de feromônio. Estes rastros permitem encontrar os menores caminhos entre o formigueiro e a fonte de alimento.
- Estas características são exploradas pelo algoritmo ACO na solução de problemas de otimização;

Introdução

- Dependendo do ponto de vista, o ACO pode ser classificado como:
 1. Algoritmos de solução por aproximação;
 2. Para a IA um algoritmos de Inteligência por Enxames (Swarm Intelligence);
 3. Para a Pesquisa Operacional (PO) é uma meta-heurística;
- Em biologia, o rastro de feromônios é explicado por um fenômeno chamado **estimergia**.

Introdução

- O ACO foi criado por Marco Dorigo na década de 90 e segue as seguintes premissas:
 1. Formigas são insetos sociais, ou seja, trocam mensagens (relacionamento) via feromônios;
 2. Observando seu comportamento em busca de alimentos, temos:
 - 2.1 Inicialmente saem do formigueiro de forma aleatória;
 - 2.2 Quando se movem deixam um rastro de feromônio por onde passam;
 - 2.3 Cada formiga tende a escolher um caminho onde a concentração de feromônio é alta;
 - 2.4 Quando encontram uma fonte de alimento, elas avaliam a qualidade e quantidade. Pegam o que podem carregar e retornam ao formigueiro deixando feromônio proporcional a característica da fonte de alimentos;

Introdução

- Para a computação, podemos pensar no ACO como uma modelagem via Grafo: $G = (V, E)$, onde :
 - V consiste de dois nós (vértices), v_s representando o formigueiro e v_d representando a fonte de alimento.
 - E consiste de duas arestas (edge), e_1 é o menor caminho entre v_d e v_s e e_2 representa o maior caminho.
- A modelagem do feromônio é realizada como τ_i para os dois links e_i ($i = 1, 2$), este representa a "força" do feromônio para um dado caminho.
- Cada formiga, sendo n_a o total, cada formiga escolhe um caminho e_1 ou e_2 partindo de v_s para v_d com a seguinte probabilidade:

$$p_i = \frac{\tau_i}{\tau_1 + \tau_2}, i = 1, 2$$

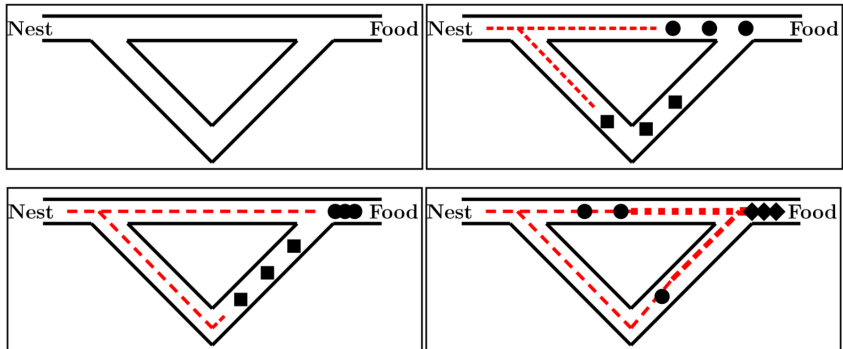
Introdução

- Sendo $\tau_1 > \tau_2$, a probabilidade de τ_1 é maior.
- Considerando o caminho escolhido até v_d , a formiga modifica o feromônio τ_i com:
$$\tau_i = \tau_i + \frac{Q}{l_i}, \text{ sendo:}$$
 - Q um parâmetro do modelo, relacionado ao índice de feromônio depositado.
 - l_i seria uma possível solução para o problema.
- A cada iteração (movimento de todas as formigas para v_s), tem-se a implementação do fenômeno natural da evaporação dos feromônio, por: $\tau_i = (1 - \rho) \cdot \tau_i$, $i = 1, 2$
 - O parâmetro $\rho \in [0, 1]$ regula o fenômeno da evaporação.

Introdução

- Mesmo inspirado, existem algumas diferenças entre o comportamento real das formigas e o implementado no ACO.
 1. As formigas reais se movem de maneira não sincronizada, diferentemente da implementação do ACO;
 2. As formigas reais depositam feromônios durante sua movimentação, no algoritmo o depósito ocorre somente no retorno a v_s ;
 3. O reforço de feromônios nas formigas reais é maior e a descoberta do menor caminho é rápido. Já as formigas implementadas no ACO tem um poder de avaliação de qualidade do alimento mais precisa.

Introdução



Caixeiro Viajante

- Para a solução do problema do caixeiro viajante, usa-se normalmente a implementação chamada Ant System (AS) que é uma implementação do ACO.
- Lembrando que a ideia é reconhecer o menor caminho para um conjunto de cidades, onde cada cidade tem um custo associado dada uma origem.
- Assim,
 1. Para cada $e_{i,j}$ será criado um $\tau_{i,j}$
 2. $\tau_{i,j}$ deve ser atualizado de acordo com a qualidade da fonte de alimentos.

Caixeiro Viajante

- Probabilidade de Seleção de Caminho:

$$p_{i,j}^k = \frac{[\tau_{i,j}(t)]^\alpha \cdot [\eta_{i,j}]^\beta}{\sum_{l \in N_i^k} [\tau_{i,l}(t)]^\alpha \cdot [\eta_{i,l}]^\beta}$$

- Onde $\eta_{i,j}$ é a função que descreve o problema. No caso do TSP seria $1/\text{distancia}$;
- Se $\alpha = 0$, só há influência da distância, assim seria algo como a busca gulosa;
- Se $\beta = 0$ existe somente a dependência do feromônio, encontrando rotinas fortemente sub-ótimas;

Caixeiro Viajante

- Evaporação:

$$\tau_{i,l}(t+1) = (1 - \text{evap}) \cdot \tau_{i,l} + \sum_{k=1}^m \Delta\tau_{i,l}^k$$

- Onde evap é uma constante de evaporação $0 > \text{evap} > 1$;
- A ideia da evaporação é fazer com que sejam "esquecidos" os caminhos menos utilizados para não induzir rotas não ótimas.

Lista de Referências

1. Henneth A. de Jong, Evolutionary Computation: A Unified Approach, MIT, 2006
2. M. Dorigo, Optimization, Learning and Natural Algorithms, PhD thesis, Politecnico di Milano, Italie, 1992
3. Kennedy, J.; Eberhart, R. (1995). "Particle Swarm Optimization". Proceedings of IEEE International Conference on Neural Networks. IV. pp. 1942–1948. doi:10.1109/ICNN.1995.488968.
4. Shi, Y.; Eberhart, R.C. (1998). "A modified particle swarm optimizer". Proceedings of IEEE International Conference on Evolutionary Computation. pp. 69–73.
5. Kennedy, J.; Eberhart, R.C. (2001). Swarm Intelligence. Morgan Kaufmann. ISBN 1-55860-595-9.