Algoritmo de colônia de formigas aplicado ao problema do caixeiro viajante

Matheus Henrique[1] , **Antônio Carlos**[2]
Universidade Federal de São João del-Rei
matheusrickbatista@aluno.ufsj.edu.br[1], juniormanhaesfilho@aluno.ufsj.edu.br[2]

Abstract

O algoritmo de colônia de formigas (ACO, sigla em inglês) é um algoritmo bioinspirado no comportamento de formigas reais, frequentemente utilizado para resolver problemas de otimização combinatória. Esse algoritmo se baseia na observação de como as formigas encontram o caminho mais curto entre a fonte de alimento e o formigueiro. Nesse trabalho, apresentamos a aplicação do ACO no problema de otimização combinatória mais famoso no campo da ciência da computação, o problema do caixeiro viajante (TSP, sigla em inglês).

1 Introdução

O problema do caixeiro viajante, na sua versão mais simples, consiste em determinar a menor rota para percorrer uma série de cidades (visitando-as uma única vez), retornando à cidade de origem. Ainda não foi possível provar que o problema possui solução ótima em tempo polinomial. Deste modo, observa-se uma tendência na utilização de métodos aproximados para resolve-lo. Este trabalho apresenta a aplicação de um desses métodos, conhecido como algoritmo de colônia de formigas (ACO). O ACO é uma heurística baseada em probabilidades, que encontra boas aproximações e, por vezes, resultados ótimos para o TSP.

A metáfora natural na qual os algoritmos de formigas são baseados nos diz que formigas são capazes de encontrar o caminho mais curto de uma fonte de alimento para seu formigueiro sem utilizar elementos visuais, explorando apenas a quantidade de feromônio que é deixado por outras formigas ao passar por uma rota.

Ao caminhar, as formigas depositam feromônio no chão, esse feromônio é o que guia outras formigas para utilizar a mesma rota. Inicialmente, quando não há feromônio para guiar as formigas decidem aleatoriamente quais caminhos seguir, mas após algumas idas e voltas entre o alimento e o formigueiro elas se alinham em um mesmo caminho, esse caminho é o menor caminho.

2 Fundamentação teórica

Vamos apresentar os aspectos teóricos do problema e do metódo utilizado.

2.1 TSP

A formulação do TSP é simples, é informado um conjunto de cidades e uma custo $c_{i,j}$ associado a cada par i, j de cidades, o caixeiro deve iniciar iniciar a jornada em uma cidade inicial, passar por todas as demais cidades do conjunto apenas uma vez e retornar a cidade de origem. O problema consiste em encontrar a rota com menor custo associado total, ou seja, o menor caminho possível.

Nesse trabalho, modelamos as cidades e suas conexões através de um grafo ($G = (V, A)$, tal que V é o conjunto que representa os vértices do grafo, as cidades que devem ser visitadas, e A é o conjunto que representa as arestas, conexões entre as cidades.

Uma aresta $a \in A$ representa a conexão entre duas cidades i, j tal que $i \in V$ e $j \in V$. A ponderação da aresta é a distância entre as duas cidades. O grafo utilizado é um grafo completo de forma que $\forall(i, j)\ \exists a \in A$, isto é, existe caminho entre qualquer par de vértices. A distância entre as cidades foi armazenada em uma matriz de distâncias.

2.2 ACO

A formulação do ACO aplicado ao TSP pode ser expressa através de passos:

- Passo 1: colocar cada formiga em uma cidade aleatória
- Passo 2: criar as rotas das formigas utilizando a regra da probabilidade
- Passo 3: calcular a distância de cada rota
- Passo 4: salvar a melhor rota
- Passo 5: atualizar os ferormônios de acordo com o *fitness* de cada rota
- Passo 6: repetir até o número máximo de iterações

Alguns aspectos são relevantes durante esse processo. O número de formigas recomendado é um número igual ao número de cidade do problema. Inicialmente, a quantidade de ferormônio é igual pra todas as rotas.

A regra de probabilidade é uma definida pela função de probabilidade, calculada pela distância entre os vértices e o feromônio depositado na aresta, tal que:

$$P_{ij} = \frac{[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta}{\sum\limits_{k \in \text{vizinhos viáveis de i}} [\tau_{ik}]^\alpha \cdot [\eta_{ik}]^\beta} \tag{1}$$

Tal que:

- P_{ij} é a probabilidade da formiga ir da cidade i para a cidade j.
- τ_{ij} é a quantidade de feromônio depositada na trilha entre as cidades i e j.
- α é um parâmetro que controla a influência do feromônio na escolha do caminho.
- η_{ij} é um fator que mede a atratividade do movimento da cidade i para a cidade j, ele é inversamente proporcional ao custo da aresta (i, j).
- β é um parâmetro que pondera a influência do custo do caminho no cálculo da probabilidade.
- A soma é feita apenas sobre as cidades vizinhas viáveis de i.

Os parâmetros α e β são bastante importantes para o calculo da probabilidade, eles devem ser ajustados empiricamente de acordo com a necessidade do problema.

A atualização do feromônio também é outro ponto que merece atenção. Em cada posição da matriz de feromônio τ_{ij} associado a aresta (i, j) ocorrem dois eventos:

- Evaporação: evita que o feromônio acumulado cresça indefinidamente, permitindo esquecer decisões ruins e encontrar soluções diversificadas.
- Depósito: feromônio acrescentado na rota, que varia de acordo com um parâmetro pré-definido.

Optamos por utilizar o método Ant System em que o depósito de feromônio é feito por todas as formigas que passam por uma determinar aresta.

O depósito de feromônio é dado por:

$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \tag{2}$$

Tal que:

- τ_{ij} é o valor do depósito de feromônio na trilha entre a cidade i e a cidade j na próxima iteração.
- τ_{ij} é o valor do feromônio na trilha entre a cidade i e a cidade j.
- ρ é a taxa de evaporação do feromônio ($0 < \rho < 1$) que controla a taxa de redução do feromônio ao longo do tempo.
- m é o número total de formigas.
- $\Delta\tau_{ij}^{k}$ é o valor do feromônio depositado pela formiga k na trilha entre a cidade i e a cidade j na iteração t.

No Ant system, $\Delta\tau_{ij}^{k}$ é escolhido como um valor fixo para todas as formigas, pois como falamos anteriormente haverá depósito por todas as formigas que passarem pela aresta.

Podemos definir $\Delta\tau_{ij}^{k}$ equivalente a $\frac{Q}{L^k}$, onde Q é uma constante que controla a quantidade total de feromônio depositada e L^k é o custo do caminho percorrido pela formiga k.

Q e ρ são definidos previamente e empiricamente de acordo com a necessidade do problema.

Existem outras formas de atualização de feromônio como elitismo e rank-based mas não utilizaremos elas nesse trabalho.

O último critério que merece atenção é o critério de parada. Utilizamos o critério do número máximo de iterações que determina o fim da execução do algoritmo.

3 Implementação

Para realizar a implementação utilizamos a matriz de distância "lau15.txt" disponibilizada para realização do trabalho. Essa matriz descreve a distância entre 15 cidades e tem valor ótimo conhecido de 291.

Os parâmetros utilizados na implementação foram:

α	β	Q	evaporação	iterações	formigas
1	2	50	0.5	10	$\|V\|$

Inicialmente realizamos a leitura do arquivo e montamos a matriz de distância com ajuda da biblioteca numpy. Com os parâmetros definidos e a matriz de distância criada podemos iniciar o algoritmo.

Os feromônios são armazenados em uma matriz semelhante a matriz de distâncias, inicialmente todos eles tem valor igual a 1.

Feito isso, podemos inicializar os dois loops principais do algoritmo. O primeiro percorre a quantidade de iterações e o segundo percorre a quantidade de formigas.

Dentro do segundo loop, para cada formiga, calculamos a função de probabilidade e construímos uma solução para o problema utilizando a regra. Ainda dentro do segundo loop selecionamos a melhor rota.

Após a ocorrência completa do segundo loop, o algoritmo atualiza os feromônios e parte para a próxima iteração.

3.1 Resultados

Nessa implementação conseguimos alcançar o resultado ótimo de 291.

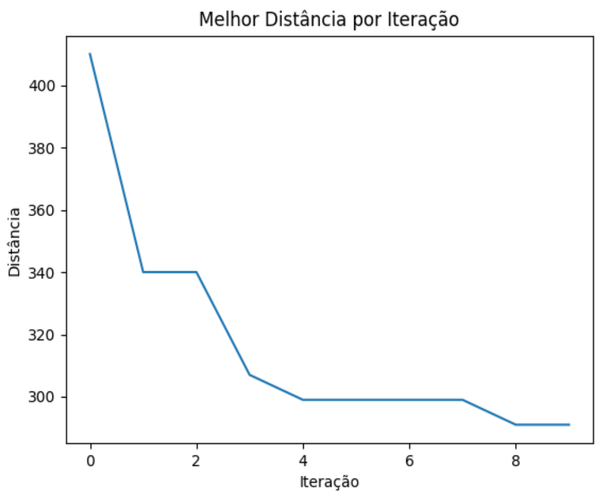
Na primeira iteração o algoritmo obteve os seguintes valores:

Rota Inicial: [6, 4, 9, 7, 5, 3, 13, 11, 2, 8, 14, 1, 12, 0, 10]
Distancia Inicial: 410.0

Na última iteração o algoritmo obteve os seguintes resultados:

Rota Final: [11, 2, 6, 4, 8, 14, 1, 12, 0, 10, 3, 5, 7, 9, 13]
Distancia Final: 291.0

A seguir apresentamos o gráfico de distância por iterações em que é possível ver a convergência do algoritmo. É possível perceber que ao chegar na geração 8 o algoritmo encontrar o valor ótimo. Os parâmetros α, β e número de iterações foram definidos em função das poucas cidades do grafo.



3.2 Conclusão

Existe um interesse crescente na aplicação de algoritmos de colônia de formigas a problemas combinatórios difíceis [2]. Esse trabalho mostrou uma aplicação introdutória da técnica na ultilização de um problema famoso. Alguns aspectos mostraram-se relevantes durante o processo:

- O algoritmo de colônia de formigas é um método flexível e, em função disso, é propício para adaptações mais simples a problemas reais [3].
- É possível atingir uma solução ótima para problemas difíceis em um tempo aceitável e, para aqueles problemas que não necessariamente precisam de um ótimo global, mas de um bom resultado aproximado, em pouco tempo, o algoritmo continua tendo grande aplicabilidade.
- O ponto sensível do algoritmo, sem dúvidas, é a definição dos parâmetros, a implementação é simples e em problemas simples os parâmetros podem ser testados diversas vezes, mas, em aplicações maiores que levam um tempo maior de execução encontrar os parâmetros ideais pode ser uma tarefa dificíl.

4 Referências

[1] GAREY, M. R.; JOHNSON, D. S. Computers and Intractability: a Guide to the Theory Of NP_Completness. 1. ed. New York: W. H. Freeman, 1979.

[2] M. Dorigo L. M. Gambardella, 1997. ”Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem”.

[3] ALOISE, D.; NORONHA, T. F.; MAIA, R. S.; BITTENCOURT, V. G.; ALOISE, D. J. Heurísticas de colônia de formigas com path-relinking para o problema de otimização da alocação de sondas de produção terrestre–SPT.