UNIVERSIDADE FEDERAL DE MINAS GERAIS

Amanda Cristina Rodrigues Tavares

Trabalho Prático 2: Ant Colony Optimization

Belo Horizonte

Outubro de 2020

Sumário

1 Introdução

Este documento possui o objetivo de detalhar a implementação, experimentos e conclusões a partir da proposta do Trabalho Prático 2, que consiste na elaboração de soluções para o *longest path problem* fazendo uso de *Ant Colony Optmization* (ACO). Os dados utilizados no trabalho se tratam de três grafos de diferentes tamanhos, nos quais dois deles já sabemos a solução ótima.

A implementação e metodologia foram baseadas nos conceitos apresentados nas aulas e também apresentados no artigo *Ant system: optimization by a colony of cooperating agents* (Dorigo; Maniezzo; Colorni, 1996).

2 Implementação

A implementação do programa foi feita em Python 3.8. Não é necessário instalar nenhum pacote para sua execução. Para rodar o algoritmo, é necessário passar o caminho do dataset como argumento ao executar o programa por linha de terminal (ex: python main.py dataset.txt).

O algoritmo se baseou na implementação do *Ant Colony Optmization* (ACO) para o problema do Caixeiro Viajante, já que é desejável encontrar uma solução que passe por todos os pontos do grafo sem visitar um ponto mais de uma vez. Foi necessário fazer algumas adaptações, pois o peso de cada aresta já é dado no arquivo de entrada do programa e os pontos do grafo não estão totalmente conectados, fazendo com que tenha poucas opções para montar o caminho. E o objetivo é encontrar o caminho com maior quantidade de peso, ao contrário do problema do Caixeiro Viajante que busca encontrar o caminho mais curto que passa por todas as cidades.

O grafo dentro do algoritmo é representado como uma matriz em que cada linha é o primeiro ponto e cada coluna representa para qual outro ponto pode visitar, sendo que a informação da coluna armazena tanto o ponto quanto o peso formado pela aresta entre os dois pontos. Uma classe foi criada para armazenar tanto a matriz com o grafo quanto outras informações como a quantidade de pontos e a taxa de feromônio para todas as arestas do grafo (que inicialmente contém o mesmo valor para todas).

Para representar cada formiga foi criada uma classe que armazena informações sobre a colônia, grafo, o peso do caminho construído pela formiga, o caminho traçado, uma lista com a quantidade de feromônio que a formiga vai deixar, quais pontos a formiga pode visitar, onde a formiga se encontra e uma lista com a *desirability* de cada aresta que pode compor o caminho.

A *desirability* é calculada como o peso representado pela aresta dividido por 100, para manter o cálculo com valores não muito grandes.

A colônia também foi representada como uma classe, contendo informações como a quantidade de formigas, quantidade de gerações, os valores de α e β que compõem os pesos no cálculo da probabilidade de escolha para cada aresta, taxa de evaporação do feromônio (um valor multiplicado ao feromônio no momento da atualização) e uma constante que é multiplicada para cada valor de feromônio no momento da atualização.

O início da caminhada da formiga é escolhido aleatoriamente. Para a formiga decidir o próximo ponto primeiro é verificado quais pontos ela pode visitar a partir do ponto em que está no momento. O cálculo da probabilidade segue a mesma lógica apresentada nos

conceitos vistos em aula:

$$p_{ij} = \frac{(\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}}{\Sigma_{k \in N(i)}(\tau_{ik})^{\alpha}(\eta_{ik})^{\beta}}$$

Figura 1 – Fórmula de probabilidade de escolha para cada aresta

sendo que η é a *desirability* e τ é a quantidade de feromônio presente na aresta.

A atualização da variação de feromônio a ser adicionada na aresta é adaptado a partir da técnica *ant-cycle* abordada no artigo *Ant system: optimization by a colony of cooperating agents* (Dorigo; Maniezzo; Colorni, 1996), em que nesta implementação é utilizada a constante definida pela colônia multiplicada pelo peso total do caminho dividido por 100. O método *ant-cycle* foi escolhido por apresentar resultados melhores do que *ant-density* e *ant-quantity*, que fazem uso de informação local, ou sejam, não olham o caminho como um todo.

A cada geração da colônia são criadas formigas a partir da quantidade decidida por parâmetro. Em seguida, é calculado o peso do caminho gerado por cada formiga e é comparado com o melhor peso já encontrado para atualizar esta informação (o melhor peso será aquele que é maior). A taxa de feromônio da formiga é recalculada. Após gerar todas as formigas, o feromônio de cada aresta é recalculado. E após rodar todas as gerações o melhor caminho encontrado é retornado.

Da forma como o algoritmo foi implementado não houveram soluções inválidas, já que sempre é levado em conta quais pontos já foram visitados e quais são os permitidos dependendo da estrutura do grafo. O caminho é construído enquanto existir próximos pontos válidos a serem escolhidos.

3 Experimentos

Pela natureza estocástica do algoritmo foi necessário repetir os testes algumas vezes para então poder analisar e tirar conclusões mais concretas. Como os resultados de um mesmo teste não variavam muito entre si, foram realizadas 10 repetições e a média foi considerada como o resultado definitivo. Para cada grafo foi testado variar os parâmetros referentes ao número de formigas, número de gerações, taxa de evaporação do feromônio e pesos (α e β) do cálculo da probabilidade.

Para o início dos testes foi escolhido o primeiro grafo por ser o mais simples, inicialmente variado o número de formigas e de gerações:

nº formigas	nº gerações	média da melhor solução
10	200	97.6
10	50	146.5
10	100	149.7
20	100	148.8

O melhor resultado encontrado foi aquele que manteve um número pequeno de formigas para um número de gerações que não fosse muito grande. Tal fato se dá pelo grafo conter 20 vértices, logo não irá necessariamente se beneficiar por um número muito grande de formigas ou gerações. Como a atração de cada aresta varia de acordo com as gerações, a tendência é fortalecer um caminho em específico, resultando na convergência. Um número elevado de gerações teve o efeito de reforçar um caminho que não chega perto do ótimo no caso deste grafo.

O próximo teste foi feito variando o valor de α e de β, que impacta diretamente o cálculo da probabilidade por favorecer ou prejudicar a importância do valor de feromônio presente na aresta ou da *desirability*:

α	β	média da melhor solução
1	10	149.7
10	10	31.4
10	1	32.1
10	20	31.2
0.5	10	149.2
10	0.5	32.3

O peso da *desirability* deve ser maior que o do feromônio para favorecer os resultados do algoritmo, sendo que aumentar muito o peso para o feromônio chega a ser extremamente prejudicial, já que tem como consequência a convergência prematura.

E por último foi testado variar os valores para a taxa de evaporação do feromônio:

taxa de evaporação	média da melhor solução
0.2	83.7
0.5	149.7
2	154.8
3	111.6
5	83.8

Uma taxa muito baixa prejudica a qualidade da solução por diminuir a importância do trabalho feito pelas formigas ao longo do algoritmo. Já um valor alto resultará numa convergência prematura, por diminuir a *exploration* do algoritmo.

O segundo grafo possui 5 vezes o número de vértices do primeiro. Consequentemente, os testes com o mesmo demoram mais tempo e os parâmetros precisam ser ajustados para melhorar a qualidade dos resultados. A análise começa pelo número de formigas e de gerações:

nº formigas	nº gerações	média da melhor solução
10	100	663.9
20	100	664.9
50	100	675.4
100	50	675.4
100	100	683.4

Pelo tamanho do grafo ser maior, houve um benefício considerável em aumentar tanto o número de formigas quanto o de gerações, ao contrário do primeiro grafo. Porém, tal aumento resulta numa performance cada vez pior do algoritmo. Portanto, para facilitar os próximos testes foi limitado um certo valor. Levando em consideração o resultado com o grafo anterior, em algum momento aumentar tais parâmetros terá efeito negativo. A relação entre número de formigas e número de gerações é proporcional entre si, por manter resultados iguais ou muito parecidos ao trocar os valores entre si.

O próximo parâmetro variado foi o da taxa de evaporação:

taxa de evaporação	média da melhor solução
0.2	681.4
0.5	683.4
1	678.2
10	681.8

Tal parâmetro não afetou fortemente os resultados do teste em comparação ao grafo anterior, implicando que quanto maior o tamanho do grafo, menos controle é possível ter dos fatores de *exploitation* e *exploration* se buscar variar a evaporação do feromônio.

Por último, foi variado os pesos de α e de β:

α	β	média da melhor solução
1	10	683.4
1	2	680.9
1	20	673.8
0.5	10	683.0
10	1	86.7
10	20	86.7

Os efeitos no ajuste de parâmetros foram semelhantes ao do primeiro grafo, reforçando a importância da *desirability* independentemente do tamanho do grafo.

No caso do terceiro grafo não foram realizados muitos testes por ele ser cerca de 10 vezes maior que o segundo e demorar ainda mais tempo para testar. Para o número de gerações e número de formigas os seguintes testes foram realizados:

nº formigas	nº gerações	média da melhor solução
100	2	5780.5
20	5	5739.4
40	5	5774.0
20	8	5784.0

Um número maior de formigas é mais importante do que um número maior de gerações para gerar soluções melhores, já que um implica em mais caminhos variados gerados (diversidade), consequentemente favorecendo o *exploration*. Uma quantidade maior de gerações também melhora a qualidade no geral por reforçar arestas que são mais escolhidas pelas formigas, ou seja, mais desejáveis como um todo e melhores para a solução.

Em seguida, foi variada a taxa de evaporação do feromônio:

taxa de evaporação	média da melhor solução
0.2	5753.5
0.5	5739.4
4	5765.4

Os resultados são um pouco parecidos com o do segundo grafo. Existe um intervalo em que o valor da taxa de evaporação resulta em qualidade pior comparado a valores que vem antes ou depois. E um valor muito grande possivelmente terá impacto negativo nos resultados.

Não foram realizados testes com os valores de α e de β por terem tido o mesmo comportamento nos grafos anteriores.

4 Conclusões

Após os testes é possível concluir que o tamanho do grafo impacta diretamente na influência que o número de formigas e o de gerações terá nos resultados. Grafos maiores precisam de mais formigas e mais gerações para ter resultados melhores. No entanto, a partir de um certo valor não existe benefício em aumentar o número de gerações ou de formigas.

A *desirability* deve ter peso maior do que a taxa de feromônio, sendo que a taxa de feromônio preferencialmente não deve ter um peso muito grande. Já a taxa de evaporação costuma ter uma faixa de valores ideal dependendo do tamanho do grafo, sendo essa faixa maior para grafos maiores.

A média dos resultados para o primeiro e segundo grafo não chegou perto do ótimo que foi informado, sendo assim existem melhorias que devem ser feitas para que o algoritmo tenha resultados melhores. Possivelmente se a *exploitation* fosse favorecida seria benéfico. Para tanto, seria necessário introduzir um sistema de elitismo ou outras alterações que visam filtrar melhor as soluções.

Referências

Dorigo, M.; Maniezzo, V.; Colorni, A. Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, v. 26, n. 1, p. 29–41, 1996. Citado 2 vezes nas páginas 2 e 4.