



UNIVERSIDADE FEDERAL DE MINAS

GERAIS

DEPARTAMENTO DE CIÊNCIA DA
COMPUTAÇÃO

Julio Guimarães

Computação Natural

Belo Horizonte

Contents

1	Introdução	1
2	Implementação	2
3	Experimentos	4
4	Conclusão	6

1 Introdução

Ant Colony Optimization (ACO) é muito famoso no conjunto de algoritmos meta-heurístico que foi introduzido pela primeira vez em 1991 como Ant System. ACO tem aplicação principal (não-determinístico em tempo polinomial) em problemas NP-difíceis, mais frequentemente em aqueles que necessitam de encontrar o caminho mais curto no gráfico. Problema do Caixeiro Viajante (TSP), Problema de Roteamento veicular, Quadratic Assignment Problem, Shortest Supersequência Comum são apenas alguns dos problemas resolvidos usando ACO.

A maior parte dos algoritmos para a procura de caminho mais longo do gráfico são baseados no pressuposto de que o gráfico é acíclico. Algoritmos que são aplicados aos gráficos cíclicos implicam uso da inteligência artificial. Aplicação da ACO para os problemas que buscam o caminho mais longo são muito raros e na maioria das vezes é utilizada apenas com gráficos sem direção. Apesar da falta de sua ampla aplicação em problemas de caminho mais longo, ACO é aplicado com sucesso na busca caminho mais longo nos Gráficos genéricos, Optimal Truss projeto e à procura de Ciclo de Hamilton em segmentos.

Neste trabalho, o ACO foi utilizado para resolver o problema de longest path, como descrito na especificação.

2 Implementação

Para implementar esse algoritmo foi utilizado um grafo, ou seja, toda a entrada é um grafo composto de vertices e arestas, lembrando que o grafo é direcional e possui peso em cada aresta. Dessa maneira o objetivo do algoritmo é aproximar o resultado colocando as formigas para "caminhar" sobre esse grafo. Para isso o grafo não pode ter vertices com self-loops, cada aresta possui um peso, os vertices são sempre conectados ao restante por uma ou mais aresta, entre outras características.

Para começar deve-se definir os valores dos parâmetros de feromônio e taxa de evaporação. Os valores iniciais de feromonas são depositadas em cada aresta do grafo. População de formigas inicial é gerado e é determinada a partir do vértice. Condição final é escolhido para ser o número de iterações, isto é, número de vezes que a colônia foi deixada atravessar o gráfico. Formigas artificiais são então liberados e atravessam o gráfico de acordo com depósitos de feromônios existentes. Todas as formigas estão usando os mesmos depósitos de feromônio durante uma iteração da sua pesquisa. Antes da próxima iteração começa todas as formigas artificiais a depositar os seus feromônios.

O caminhar da formiga é feito de acordo com o feromônio, ou seja, a formiga irá para o vertice cujo o caminho que leva até ele possuir mais feromônio. Além disso cada formiga atualiza o feromônio do caminho que

ela teve que passar, fazendo com que esse caminho seja preferido por outras formigas nas próximas iterações. Dessa maneira, as formigas caminham pelo grafo em busca de uma solução.

A solução é representada pelo valor calculado para aquele caminho, dessa maneira para saber sua fitness basta saber esse valor e comparar com a fitness de outras soluções, para saber qual tem a maior fitness, logo o maior caminho, para encontrar a melhor solução.

3 Experimentos

Para realizar os testes de desempenho, foi utilizado uma máquina Linux FEDORA 64 bits, 8 GB de memória RAM, Intel Core i5, @2.3 GHz.

Para isso foram feitos vários teste variando o valor do feromônio e da taxa de evaporação, para que pudesse ser obtido os melhores resultados. Abaixo segue as tabelas de valores obtidos no final dos testes

Os testes foram feito de última hora, apenas para verificar que o trabalho estava funcionando como especificado. Porém nenhuma analise foi gerada em cima dos resultados por falta de tempo. Porém algumas imagens dos testes e dos resultados obtidos seguem abaixo.

Entrada 1:

Formigas	Feromonio	Taxa de evaporação	Solução
100	0,005	20	896,796
	0,05	20	850,33
	0,5	20	794,635
150	0,005	20	957,124
	0,05	20	907,864
	0,5	20	883,642
200	0,005	20	1.331,560
	0,05	20	1192,654
	0,5	20	1001,42

Entrada 2:

Formigas	Feromonio	Taxa de evaporação	Solução
20	0,005	20	144,989
	0,05	20	135,465
	0,5	20	121,326
30	0,005	20	160,987
	0,05	20	154,863
	0,5	20	138,648
60	0,005	20	194,621
	0,05	20	187,564
	0,5	20	173,687

Dessa maneira é possível analisar que o algoritmo melhora seus resultados de acordo com o aumento da quantidade de formiga, por outro lado, isso faz com que demore muito mais tempo para chegar em um resultado. Além disso foi possível perceber que quanto mais feromônio, mais próximo o resultado fica da solução ótima.

4 Conclusão

No presente trabalho prático, foram implementado o algoritmo ACO para solucionar o problema de longest path. Foram exercitados conceitos adquiridos em sala de aula, como a própria implementação do algoritmo ACO, e como realizar o tratamento de seus parâmetros para conseguir os resultados esperados. Além disso tive que estudar como funcionava o problema de longest path, e como seria feita sua solução utilizando um grafo e as formigas para encontrar seus valores.

O trabalho exige bastante empenho e tempo para que possam ser feitos os testes como pedido na especificação, o que não ocorreu no meu caso, por falta de tempo útil, apenas realizei os testes para as duas primeiras entradas, afim de comparar sua solução com a solução ótima já fornecida. Assim pude analisar esses resultados e chegar a conclusão desejada, que foi na corretude da implementação do algoritmo ACO para solução do problema de longest path.