Universidade Federal de São João del-Rei
Algoritmo Bioinspirados
Algoritmo Ant System para resolução do PCV

Felipe Francisco Rios de Melo

1. INTRODUÇÃO

Otimização por colônia de formigas (OCF OU ACO, do inglês Ant Colony Optimization) é uma meta-heurística populacional que pode ser usada para encontrar soluções aproximadas para problemas difíceis de otimização.

Em OCF, um conjunto de agentes (formigas artificiais) procuram por soluções para um certo problema de otimização. As formigas artificiais constroem de forma incremental soluções através de agregação de elementos ou conjuntos de elementos do grafo. O processo de construção das soluções é estocástico e guiado pelo modelo de feromônios, que é um conjunto de valores associados com as componentes do grafo, sejam vértices ou arestas. e que sofrem alteração durante o trabalho das formigas artificiais.

O Ant System foi o primeiro algoritmo que surgiu inspirado em colônia de formigas naturais. Os detalhes que caracterizam este algoritmo são:

- Ao tomar um caminho a formiga deixa no mesmo uma certa quantidade de feromônio;
- Uma formiga escolhe determinado caminho de acordo com uma função probabilística envolvendo a distância deste caminho e a quantidade de feromônio presente neste;
- As formigas ”lembram” os pontos por onde já passaram e não retornam a estes pontos até que tenham chegado à fonte de alimento;
- O feromônio evapora.

Este trabalho tem por objetivo a implementação e análise de do Ant System para a resolução do problema do caixeiro viajante.

O problema tem o seguinte enunciado:

O problema PCV consiste na busca por um circuito que possua a menor distância começando numa cidade qualquer, entre várias, e visitando todas cidades, cada uma precisamente uma vez, voltando então para a cidade de origem.

Dado um conjunto $C = \{c_1,..., c_n\}$ de n cidades c_i e uma matriz de distâncias (ρ_{ij}), onde $\rho_{ij} = \rho(c_i, c_j)((i, j) \in \{1, ..., n\}, \rho_{ij} = \rho_{ji}, \rho_{ii}= 0)$, a tarefa passa por encontrar a

permutação $\tilde{\pi} \in S_n = \{s: \{1,...,n\} \rightarrow \{1,...,n\}\}$ que faça com que a função objetivo (distância do circuito) $f: S_n \rightarrow \mathbb{R}$, onde:

$$f(\pi) = \sum_{i=1}^{n-1} \rho(\pi(i),\pi(i+1)) + \rho(\pi(n),\pi(1)),$$

seja mínima.

2. IMPLEMENTAÇÃO

A seguir, será mostrado um resumo das decisões tomadas para a implementação do Ant System:

2.1. Estrutura de dados

Foi implementado uma classe nomeada AntSystem, que é instanciada uma única vez, contendo os atributos e métodos necessários para inicialização e manipulação do algoritmo.

- Atributos: tamanho da população, quantidade de iterações, parâmetro α, parâmetro β, parâmetro Q, matriz de feromônio, vetor de indivíduos, taxa de evaporação, melhor solução encontrada, avaliação da melhor solução encontrada, além das variáveis referentes a instância usada: a matriz de distâncias e a quantidade de nós;
- Métodos: iniciar a população de formigas, iniciar a matriz de feromônios, atualizar a matriz de feromônios, calcular a função objetivo, escolher probabilisticamente o próximo nó e etc.

Além da classe AG, existe também a classe Ant, usada para armazenar a solução de cada indivíduo para o PCV, a avaliação do indivíduo na função objetivo, e o nó corrente do indivíduo.

2.2 População

Se o tamanho da população for maior que a quantidade de vértices, ele deve ser proporcional a quantidade de vértices da instância. Onde geramos a população através da instanciação da classe Ant, iterativamente, alocando cada formiga em um dos *n* vértices para que tenha diversificação nas soluções encontradas através dos diferentes pontos de partida.
Se o tamanho da população for menor que a quantidade de vértices, a alocação de cada formiga é feita por sorteio.

2.3. Feromônio

A matriz de feromônios que tem dimensões n x n, onde n é a quantidade de vértices do problema, é inicializada com uma distribuição igual para todos os elementos da matriz, atribuindo a cada elemento o inteiro 10^{-16}.

2.4. Construção da Solução

A solução construída por uma formiga é dado de maneira iterativa, onde que enquanto não visitar todos os vértices, é selecionado de forma probabilística, o próximo vértice *j* que ainda não foi visitado a partir do vértice atual *i*.

Dada uma formiga *k* no vértice *i*, a probabilidade dela escolher a cidade *j* é dada pela equação abaixo:

$$p_{ij}^{k} = \frac{(\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}}{\sum_{l \in \mathcal{N}_i^k} (\tau_{il})^{\alpha}(\eta_{il})^{\beta}} \qquad (2)$$

Onde τ_{ij} é o feromônio associado à aresta (i,j), $\mathcal{N}_i^k$ é a vizinhança factível da formiga k (o conjunto de cidades ainda não visitadas por ela) e η_{ij} é um fator inversamente proporcional ao custo da aresta (i,j), para o PCV:

$$\eta_{ij} = \frac{1}{d_{ij}}$$

Os valores de α e β são parâmetros ajustáveis, o α pondera a influência do feromônio e o β pondera a influência do custo do caminho no cálculo da probabilidade.

Para todas os vértices não visitados, é calculado a probabilidade do vértice ser escolhido pela equação mostrada acima. A probabilidade para cada vértice é colocado em uma lista, e é sorteado um valor entre 0 e 1, iniciando a partir do primeiro elemento da lista, as probabilidades são somadas em um acumulador, até que se atinja o valor sorteado. O último elemento a ser somado no acumulador quando o valor foi atingido é o selecionado pelo indivíduo. Este processo simula uma escolha por roleta.

Este processo é repetido até que todos os vértices tenham sido visitados. Por fim, repete-se o primeiro vértice da solução no fim da solução, devido a especificação do PCV.

2.5. Avaliação na Função Objetivo

Construído a solução completa, ela é avaliada na função objetivo, que consiste no somatório das distâncias dadas pela sequência de vértices visitados do vetor de solução.

Se a solução corrente for melhor que a melhor solução até o momento S*, esta passa a ser a solução S*.

2.6. Atualização do Feromônio

Em cada posição da matriz de feromônio ocorrem dois eventos: a evaporação que evita que o feromônio acumulado cresça indefinidamente, permitindo esquecer decisões ruins e encontrar soluções diversificadas, e o depósito de feromônio, que varia de acordo com a variante do algoritmo baseado em colônia de formigas.

No Ant System padrão, o depósito de feromônio é feito por todas as formigas que passaram pela aresta (i, j).

Somente depois que todas as formigas construíram suas viagens, o feromônio é atualizado da seguinte forma:

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

onde:

- $\Delta\tau_{ij}^{k} = \frac{Q}{L_k}$ quando a aresta (i, j) pertence S_k;
- $\Delta\tau_{ij}^{k} = 0$, caso contrário
- Q é uma variável definida pelo usuário;
- $0 < \rho \leq 1$ é a taxa de evaporação de feromônio.

A variável Q foi fixada em 100, enquanto que para *p*, que é a taxa de evaporação de feromônio, foi realizado a calibragem de parâmetro para encontrar o melhor para ela.

2.7. Processo Iterativo

Temos um laço de repetição que executa o processo de construção de solução (seção 2.4) e avaliação (seção 2.5) para cada uma das formigas da população.

Englobando este laço que passa por todas as formigas construindo e avaliando a solução encontrada por ela, temos um outro laço que executa um número definido de vezes atualizando, ao fim de cada iteração, o feromônio (dado que o feromônio deve ser atualizado somente quando todas as formigas tiverem construído sem caminho) e criando uma nova população de formigas para encontrarem novas soluções.

3. ANÁLISE DOS RESULTADOS

O funcionamento do algoritmo foi testado com a instância LAU15, fornecida no seguinte endereço: https://people.sc.fsu.edu/~jburkardt/datasets/cities/.

As instância descreve a distancia entre 15 cidades, e é dado também a solução ótima, que é 291.

Para calibrar os parâmetros do algoritmo aplicado ao problema do caixeiro viajante, foi realizado um experimento fatorial completo variando o tamanho da população, o valor de α e β, e a taxa de evaporação do feromônio, nos seguintes valores:

Tamanho da população	25% da quantidade de vértices	50% da quantidade de vértices	100% da quantidade de vértices	
α	0	1	2	
β	0	3	5	
Taxa de evaporação	0.25	0.50	0.75	1

Variando estes parâmetros temos 108 combinações possíveis, e realizado 20 execuções do algoritmo, para obter um resultado confiável, resulta num total de 2160 execuções, estas foram efetuadas de maneira automatizada pelo arquivo python *analysis.py*.

O número de iterações foi fixado em 50, para reduzir o tempo de processamento do experimento, e Q foi fixado em 100, pois nos testes iniciais, se mostrou um bom valor para esta instância usada.

A identificação da combinação de parâmetros referente aos dados mostrados nesta análise estarão no padrão (*escalar que é multiplicado pela quantidade de vértices para se obter o tamanho da população - alpha - beta - taxa de evaporação*). Por exemplo: 0.5-1-5-0.25.

Para iniciar a calibragem de parâmetros, vamos pela análise de quantas vezes cada combinação atingiu a solução ótima, dada pelo gráfico abaixo:

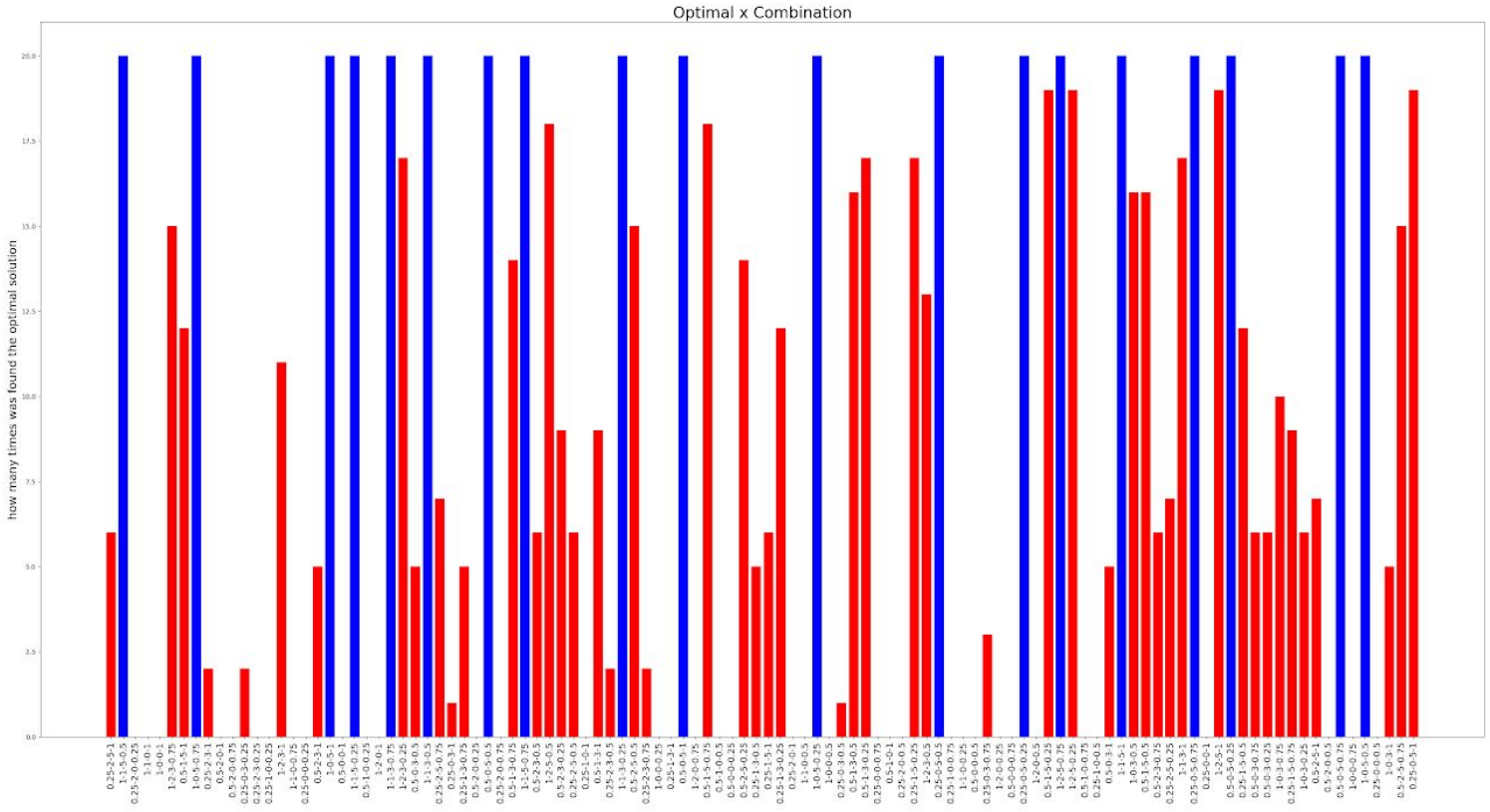

Gráfico 1 - Quantidade de vezes que cada combinação atingiu a solução ótima.

As combinações com barras azuis são as que conseguiram alcançar a solução ótima em todas as 20 execuções. Devido a visibilidade do gráfico não estar tão clara, abaixo estão as combinações que atingiram o ótimo:

> ['1-1-5-0.5', '1-0-5-0.75', '1-0-5-1', '1-1-5-0.25', '1-1-3-0.75', '1-1-3-0.5', '0.5-0-5-0.5', '1-1-5-0.75', '1-1-3-0.25', '0.5-0-5-1', '1-0-5-0.25', '0.25-0-5-0.5', '0.25-0-5-0.25', '1-2-5-0.75', '1-1-5-1', '0.25-0-5-0.75', '0.5-0-5-0.25', '0.5-0-5-0.75', '1-0-5-0.5']

Observamos que, das 19 combinações que obtiveram a solução ótima, 16 delas possuem β = 5, as outras 3 possuem β = 3 e nenhuma possui β = 0. Quando β = 0, as cidades associadas a arestas com maior quantidade de feromônio tenderão a ser escolhidas, além de que, devido a política de atualização de feromônio implementada, em que todas as formigas deixam feromônio por onde passam, tanto formigas boas quanto formigas ruins irão deixar feromônios nas arestas que

passaram. Então como a função de probabilidade tende a escolher cidades com arestas com maior quantidade de feromônio (β= 0) a escolha do nó se torna, de certa forma, aleatória, pois não tem nenhum critério específico para o depósito de feromônio. Tendo um valor maior para β, passamos a considerar também o custo do caminho da aresta, e não só o feromônio, permitindo que a escolha da próxima cidade passe a ser não aleatória, sendo, então, 5 um bom valor para β.

Quanto ao valor ideal para α, 11 combinações ótimas tem como α = 0, 7 combinações tem α = 1 e apenas 1 tem α = 2. Quando α = 0 as cidades mais próximas tendem a ser escolhidas, desconsiderando totalmente o feromônio na aresta. Porém, este comportamento observado, em que α = 0 prevaleceu sobre os outros valores não significa que a matriz de feromônio não estabelece relação nenhuma sobre a convergência do algoritmo, pelo contrário, como será visto a seguir.

Abaixo temos uma comparação entre α = 1 e α = 0, para combinações os mesmos parâmetros (com exceção do α).

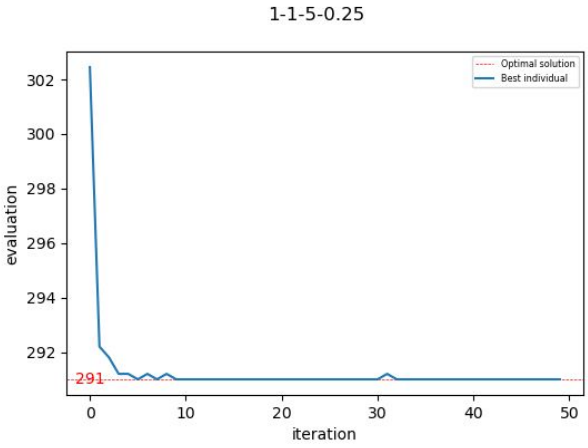

Gráfico 2 - Média dos melhores indivíduos em combinação com α = 1

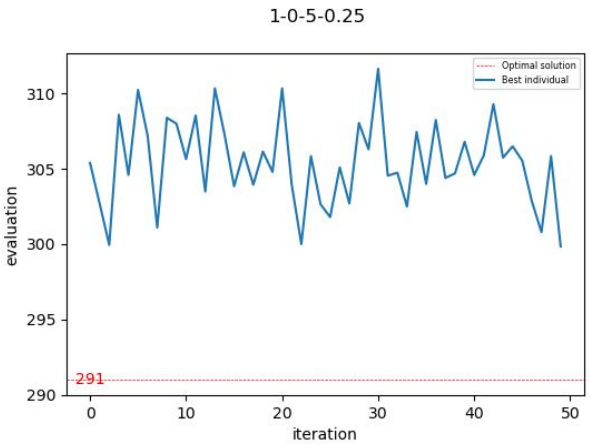

Gráfico 3 - Média dos melhores indivíduos em combinação com α = 0

O gráfico 1, mostra apenas se em algum momento da execução, o algoritmo chegou ao ótimo global, ele não tem nenhuma relação com a convergência do algoritmo para uma dada combinação. Já os gráficos 2 e 3, cumprem este papel, neles temos a média dos melhores indivíduos em cada geração para cada combinação.

Observamos claramente que a combinação com α = 1, convergiu para a solução ótima, enquanto que com α = 0, em alguma iteração algum indivíduo chegou a solução ótima, mas nas iterações seguintes, os próximos indivíduos não seguiram seus caminhos pois, como dito, com α = 0 é desconsiderado totalmente o feromônio na aresta. Mesmo com α = 0, indivíduos esporadicamente chegam ao mínimo global, devido ao pequeno número de cidades (15) da instância e ao número considerável de iterações (50). Porém, dadas todas as justificativas acima, o valor ideal para α nesta instância é α = 1.

É possível notar também, que combinações com um maior número de populações tende a encontrar a melhor solução mais vezes, das 19 combinações que chegaram ao mínimo global, 12 tem a população equivalente a quantidade de cidades. O que é de se esperar, pois quanto maior a solução mais candidatos a encontrar o caminho de custo mínimo.

Em relação a taxa de evaporação do feromônio, considerando as combinações com tamanho da população igual a 100% dos número de vértices, α = 1 e β = 5, que são os valores que apresentaram melhor qualidade nas soluções, temos os seguintes resultados:

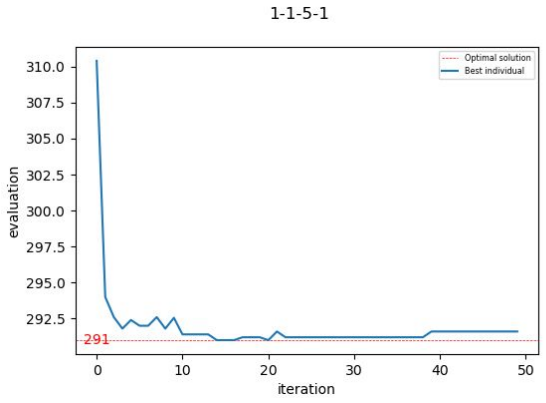

Gráfico 4 - Combinação com taxa de evaporação = 1

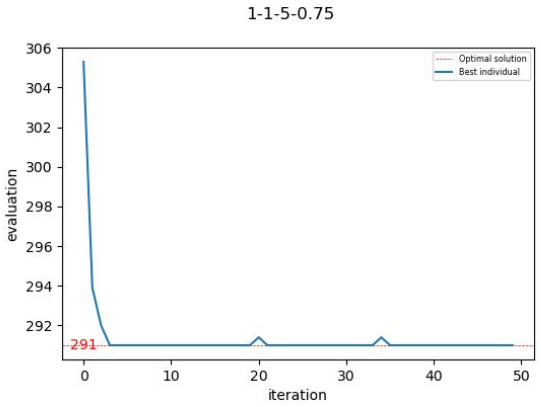

Gráfico 5 - Combinação com taxa de evaporação = 0.75

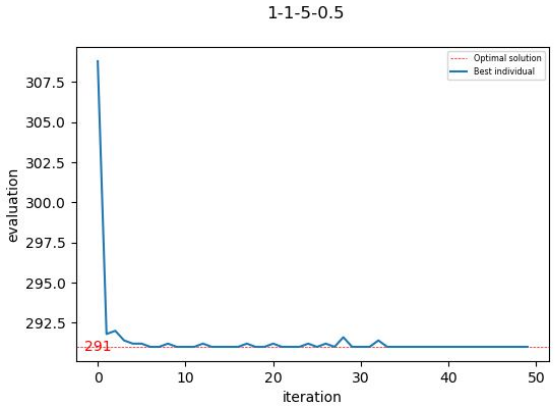

Gráfico 6 - Combinação com taxa de evaporação = 0.5

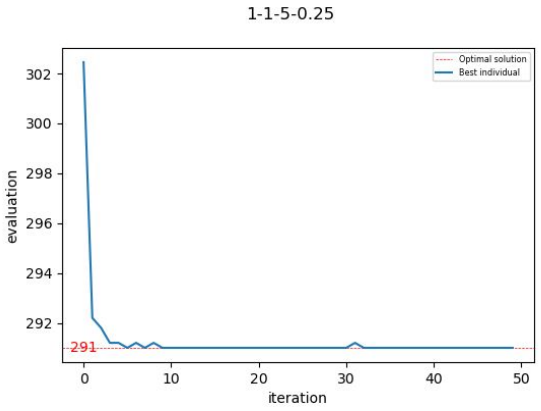

Gráfico 7 - Combinação com taxa de evaporação = 0.25

Notamos que, uma de evaporação de 100% não oferece soluções tão boas quanto as soluções geradas pelas outras variações de taxas. O que é esperado, pois com uma evaporação de 100% a única informação que as formigas da iteração corrente terão será o feromônio liberado pelas formigas da iteração anterior. Já com uma taxa de evaporação diferente de 100%, as formigas da iteração corrente terão não só o feromônio liberado pelas formigas da iteração anterior como também uma parte não evaporada do feromônio liberado pelas formigas das gerações anteriores à geração imediatamente anterior.

Pela observação do gráfico, podemos definir a taxa de evaporação igual a 25%, como a taxa ideal para esta instância.

Tendo por fim, como resultado do experimento fatorial completo para a calibragem dos parâmetros, os seguintes parâmetros para obter uma solução de qualidade:

- Tamanho da população = 100% da quantidade de vértices
- $\beta = 5$
- $\alpha = 1$
- Taxa de evaporação (ρ) = 25% (0.25)
- Quantidade de iterações = 50
- Q = 100

4. CONCLUSÃO

Com este exercício de implementação foi possível aplicar o conceito de um algoritmo baseado em colônia de formigas em um problema conhecido na ciência da computação que é o problema do caixeiro viajante (PCV).

E em relação ao experimento fatorial completo, os parâmetros foram bem calibrados para esta instância em específico, executando para uma instância maior (como a SGB128.txt, contida na pasta ‘instance’ do diretório do trabalho), seria necessário uma recalibragem. Por exemplo uma alteração no tamanho da população, pois uma população de 128 indivíduos aumentaria consideravelmente o custo computacional do algoritmo.