

Computação Natural

Trabalho prático 2 - Ant Colony Optimization

Universidade Federal de Minas Gerais (UFMG) – Belo Horizonte, MG – Brasil

pedrobrum@dcc.ufmg.br

1. Introdução

Nesse trabalho é apresentado uma solução para o problema conhecido como *longest path problem*¹, utilizando *Ant Colony Optimization* (ACO). Dado um grafo $G(V, E)$, uma função $w : E \rightarrow \mathbb{R}$ que atribui pesos a cada aresta e dois vértices $u, v \in V$, denotaremos como ρ o conjunto de caminhos simples partindo de u e chegando em v . Assim, o problema consiste em encontrar $P^* = e_1^*, e_2^*, \dots, e_k^*$ tal que:

$$P^* = \arg \max_{P \in \rho} \sum_{e_i \in P} w(e_i) \quad (1)$$

Dessa forma, queremos encontrar o caminho simples entre u e v que possui o maior custo. Nesse trabalho, em particular, foi definido que devemos encontrar o caminho de custo máximo entre o vértice de índice 1 e o vértice de índice n .

As decisões de implementação que serão apresentadas e discutidas nesse trabalho são: como representar uma solução, como calcular a *fitness* de uma solução, como calcular a probabilidade de transição e como lidar com soluções inválidas. A solução desenvolvida foi executada e avaliada utilizando 3 grafos diferentes.

Na seção 2 é apresentada uma descrição sobre a implementação do programa, incluindo detalhes da representação, função de probabilidade e estratégias de atualização de ferômonio utilizadas. Na seção 4 é apresentada uma análise do impactos dos parâmetros do algoritmo no resultado.

2. Implementação

A implementação da solução proposta foi realizada utilizando a linguagem *python*. Foram criados objetos que representam os principais envolvidos no processo de otimização: grafo, formiga e a colônia. Cada objeto foi implementado como uma classe e possui atributos e métodos.

A inspiração para o ACO é o comportamento de colônias de formigas. Em uma colônia, cada formiga inicia um caminho de forma aleatória. As formigas enquanto andam, depositam ferômonio ao longo do caminho. Depois de um certo período de tempo, a formiga que segue o caminho mais curto para coletar comida deposita mais ferômonio no seu caminho. Dessa forma, caminhos mais curtos terão uma maior concentração de ferômonio. Quanto maior for a quantidade de ferômonio em um caminho maior é a probabilidade das formigas escolherem aquele caminho. Com o passar do tempo todas a colônia

¹https://en.wikipedia.org/wiki/Longest_path_problem

segue pelo caminho mais curto para coletar comida. A analogia feita com computação é cada caminho pode representar uma possível solução e a quantidade de ferômonio depositada é proporcional a qualidade da solução. Assim, entre duas soluções, a probabilidade da melhor ser escolhida é maior.

Nesse trabalho devemos procurar o caminho mais longo entre dois vértices em um grafo direcionado e ponderado. Assim, podemos associar a cada aresta do grafo um certa quantidade de ferômonio. Ao longo da execução do programa as formigas são capazes de senter e deixar ferômonio. Uma formiga deve escolher uma aresta para seguir o caminho de acordo com uma probabilidade, que depende da concentração de ferômonio existente nela. O ferômonio presente em cada aresta é atualizado em cada formiga que caminha. Dessa forma, uma solução para o problema resolvido é representada por um caminho percorrido por uma formiga. Ou seja, Uma solução é representada por uma lista de arestas ($P^* = e_1^*, e_2^*, \dots, e_k^*$).

Para o problema do caminho mais longo a qualidade de cada aresta que pode ser adicionado a uma solução candidata parcial pode ser medida pelo seu peso. Assim, a *fitness* de cada componente (arestas) para o problema do caminho mais longo é o peso da aresta/caminho. Vamos representar isso pela função η (desirability).

Um dos principais elementos de um algoritmo de ACO é a regra de transição probabilística, que é baseada no valor da função η e na quantidade de ferômonio τ associada a cada componente de uma solução candidata. Essa regra decide qual componente será inserido na solução parcial e é proporcional ao produto $\eta_i \times \tau_i$. A probabilidade de escolher a aresta ij foi definida como:

$$p_{ij} = \frac{(\tau_{ij})^\alpha (n_{ij})^\beta}{\sum_{k \in N(i)} (\tau_{ik})^\alpha (n_{ik})^\beta} \quad (2)$$

onde τ_{ij} é a quantidade de ferômonio na aresta ij , η_{ij} é o valor da função de *desirability* e $N(i)$ é o conjunto de vizinhos elegíveis de i . Nesse trabalho, em particular, devemos procurar o caminho mais longo entre o vértice de número 1 e o vértice de número n . Dessa forma, caminhos que não chegam em n são soluções inválidas. No programa implementado, essas soluções recebem custo igual a -1. Dessa forma, qualquer solução entre os vértices 1 e n é considerada melhor que uma solução inválida.

Outro elemento muito importante de um algoritmo de ACO é a regra para atualização do ferômonio. Essa regra especifica como atualizar a quantidade de ferômonio (τ) associada a cada aresta inserido no caminho seguido por uma formiga. O ferômonio deve aumentar de forma proporcional a qualidade do caminho. A estratégia de atualização de ferômonio escolhida para esse trabalho foi a *Max - min* e ocorre da seguinte forma:

1. Aplica-se um regra de evaporação de ferômonio, diminuindo o seu valor em cada aresta ij do grafo por uma constante:

$$\tau_{ij} = (1 - \sigma) \times \tau_{ij}, \forall ij \in E \quad (3)$$

onde $\sigma \in (0, 1]$ é a taxa de evaporação de ferômonio.

2. **Apenas a melhor formiga da iteração** atualiza o ferômonio, de acordo com os limites $[\tau_{min}, \tau_{max}]$. Essa formiga deposita uma quantidade $\Delta\tau_{ij}$ em cada aresta presente no caminho realizado por ela:

$$\tau_{ij} = \tau_{ij} + \Delta\tau_{ij} \quad (4)$$

$$\Delta\tau_{ij} = Q \times \left(1 - \frac{1}{\eta_{ij}}\right) \quad (5)$$

onde $Q \in (0, 1]$. Vamos chamar Q de taxa de intensidade de ferômonio.

Portanto, foi necessário dedidir os valores de ferômonio inicial em cada aresta (τ_{ij}), intensidade de ferômonio (Q), alpha (α), beta (β), τ_{min} e τ_{max} .

Alguns testes foram realizados para se decidir os valores de cada um desses parâmetros e os melhores valores encontrados foram:

- Ferômonio inicial: 1.0
- Intensidade de ferômonio: 0.5
- Valor de *Alpha*: 1.0
- Valor de *Beta*: 4.0
- Valor de τ_{min} : 10^{-4}
- Valor de τ_{max} : 10.0

Assim, para todos os experimentos esses parâmetros recebem esses valores como padrão. Além desses parâmetros, temos o número de formigas, número de iterações e a taxa de evaporação de ferômonio. O método principal para solução do problema é mostrado no Algoritmo 1. Para cada iteração temos: para cada formiga, escolhemos o vértice 1 como o início e utilizamos valores de ferômonio e da função η para construir incrementamente um caminho (Algoritmo 2).

Algorithm 1: Solve

input : Grafo $G = (V, E)$
output: Melhor solução encontrada

```

1 begin
2    $k \leftarrow$  Número de iterações a serem realizadas
3    $a \leftarrow$  Número de formigas
4    $best\_solution \leftarrow$  Melhor solução inicial (custo 0)
5   for  $i = 0$  to  $k$  do
6      $A \leftarrow$  Conjunto de formigas de tamanho  $n$ 
7      $BuildSolutions(G, A)$  // Gera as soluções
8      $best\_solution \leftarrow$  Melhor solução gerada
9      $MaxMin(G, best\_solution)$  // Atualiza o ferômonio
10 return  $best\_solution$ 
```

Algorithm 2: BuildSolutions

input : Grafo $G = (V, E)$; Conjunto de formigas A

output: Soluções para cada formiga

```
1 begin
2    $n \leftarrow$  número de vértices
3   foreach  $a_i \in A$  do
4     for  $i = 0$  to  $n$  do
5       SelectNext() // Seleciona uma aresta
6        $r \leftarrow$  Aresta selecionada
7       if  $|N(r)| = 0$  and  $r \neq n$  then
8         // Custo do caminho construído é igual a -1
9         // break
10      if  $r = n$  then
11        // break
12 return Soluções para cada formiga
```

3. Arquivos, compilação e execução

3.1. Arquivos

São partes do código-fonte do programa os seguintes arquivos:

- *main.py* : arquivo principal de execução. Nele estão implementados os métodos de leitura do grafo e leitura de parâmetros.
- *aco.py* : arquivo onde estão implementadas as classes *Graph*, *ACO* e *Ant*, e seus respectivos métodos e componentes principais.

3.2. Compilação

A compilação do programa pode ser realizada da seguinte forma:

```
$ python main.py -g <grafo> -a <numero de formigas>
-i <numero de iteracoes> -e <taxa de evaporacao>
-o <arquivo de saida>
```

3.3. Execução

A passagem de parâmetros é opcional, e eles assumirão seus respectivos valores padrões se não forem informados:

1. Número de formigas (inteiro). Padrão: 200
2. Número de iterações (inteiro). Padrão: 100
3. Taxa de evaporação de feromônio (float). Padrão: 0.2
4. Intensidade de feromônio (float). Padrão: 0.5
5. Valor do feromônia inicial (float). Padrão: 1.0
6. Valor de *Alpha* (inteiro). Padrão: 1
7. Valor de *Beta* (inteiro). Padrão: 4
8. Número de repetições (inteiro). Padrão: 10

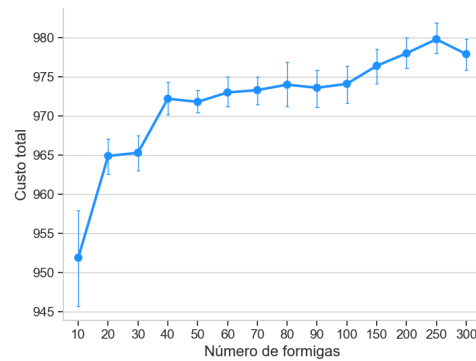
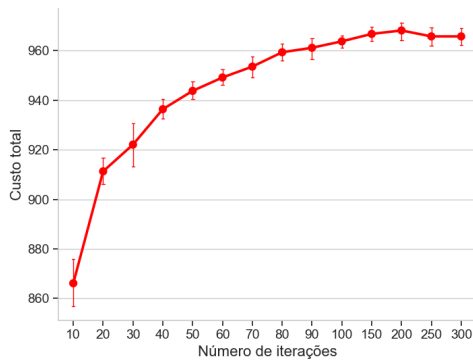


Figura 1: Valores médios do custo encontrado pelo ACO para diferentes números de iterações considerando o primeiro grafo. Figura 2: Valores médios do custo encontrado pelo ACO para diferentes números de formigas considerando o primeiro grafo.

4. Experimentos

4.1. Bases de dados

Para realizar a avaliação experimental do programa implementado foram utilizados três grafos direcionados e ponderados, com pesos entre 1 e 10. O primeiro grafo possui 100 vértices e 8020 arestas. O segundo grafo possui 20 vértices e 190 arestas. Por fim, o terceiro possui 1000 vértices e 499500 arestas. Sabemos que a solução ótima para o primeiro grafo é igual a 168 e que para o segundo grafo é igual a 990. Como o primeiro e o segundo grafo são menores, as análises foram feitas de forma mais minuciosa neles.

4.2. Método

Foi realizada uma análise de sensibilidade dos principais parâmetros do ACO: número de formigas n_f , número de iterações (n_i) e taxa de evaporação de feromônio e . Ao mexer em um dos parâmetros, todos os outros foram mantidos constantes. Como dito anteriormente, o algoritmo procura o caminho mais longo entre o primeiro vértice (número 1) e o último vértice (número n). Como o algoritmo baseado em colônia de formigas é um método estocástico, a avaliação experimental foi realizada com repetições. No primeiro e no segundo grafo, para cada escolha de parâmetros o algoritmo foi executado 10 vezes. No terceiro grafo, para cada escolha de parâmetros o algoritmo foi executado 5 vezes. Abaixo, temos uma lista não-exaustiva de parâmetros a se avaliar:

- Número de formigas
- Número de iterações
- Taxa de evaporação de feromônio

4.3. Análise dos parâmetros

4.3.1. Grafo 1

Abaixo estão descritos os experimentos realizados para o primeiro grafo, bem como discussões sobre os resultados.

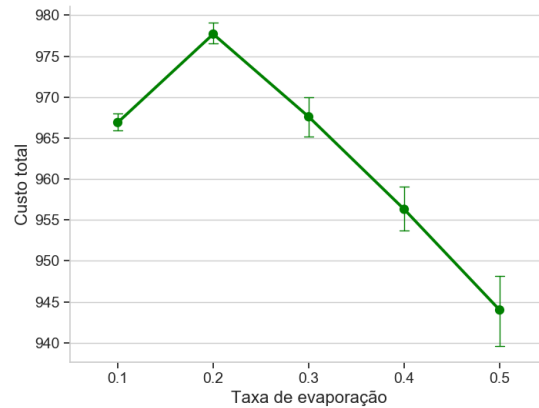


Figura 3: Valores médios do custo encontrado pelo ACO para diferentes taxas de evaporação de feromônio considerando o primeiro grafo.

Para gerar os resultados para diferentes números de iterações, foram mantidos o número de formigas $n_f = 20$ e taxa de evaporação de feromônio $e = 0.2$. Foram gerados resultados utilizando de 10 a 100 iterações e utilizando 150, 200, 250 e 300 iterações.

Na figura 1 temos o valor médio do custo encontrado pelo ACO para diferentes números de iterações, bem como o desvio padrão. Podemos concluir que quanto maior o número de iterações melhor é o resultado final obtido e menor é o desvio padrão. Além disso, temos que, em um certo ponto, não é possível melhorar os resultados apenas aumentando o número de iterações. Apesar dos resultados utilizando 100 ou mais iterações apresentarem resultados melhores, o tempo de execução do programa aumenta consideravelmente. Mesmo utilizando menos iterações podemos obter resultados razoáveis em relação ao ótimo.

Quanto maior o número de iterações maior é a quantidade de feromônio depositada em arestas de caminhos melhores e menor é a quantidade de feromônio depositada em arestas de caminhos piores em relação ao ótimo. Assim, quanto maior o número de iterações maior é a probabilidade de se encontrar o caminho de custo ótimo. Porém, não conseguimos melhorar o resultado apenas aumentando o número de iterações. Provavelmente, isso ocorre porque 20 formigas não conseguem buscar soluções por todo grafo, devido o seu tamanho de 100 vértices e 8020 arestas.

Para gerar os resultados para diferentes números de formigas, foram mantidos $n_i = 100$ e $e = 0.2$. Foram gerados resultados utilizando de 10 a 100 formigas e utilizando 150, 200, 250 e 300 formigas.

Na figura 2 temos o valor médio do custo encontrado pelo ACO para diferentes número de formigas, bem como o desvio padrão. Podemos concluir que os melhores resultados são obtidos quando utilizamos 150 ou mais formigas. Quanto maior o número de formigas maior é o número de caminhos distintos encontrados a cada iteração, ou seja, maior é o espaço de busca. Assim, a probabilidade de se encontrar o caminho de custo ótimo é maior. Isso comprova a nossa tese de que utilizar poucas formigas e muitas iterações não é o suficiente para encontrar uma boa solução.

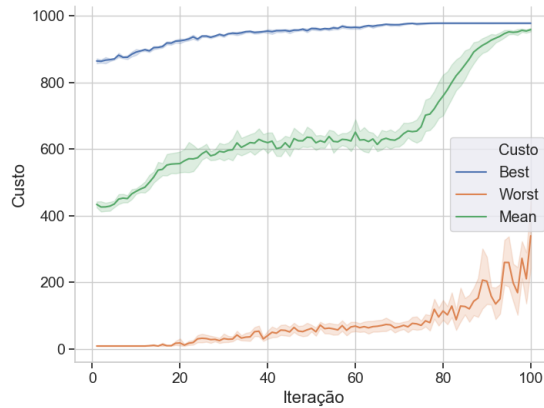


Figura 4: Valores do custo médio, melhor custo e pior custo ao longo das iterações.

Para gerar os resultados para diferentes taxas de evaporação de ferômonio, foram mantidos $n_i = 100$ e $n_f = 200$. Foram gerados resultados utilizando taxas iguais a 0.1, 0.2, 0.3, 0.4 e 0.5.

Na figura 3 temos o valor médio do custo encontrado pelo ACO para diferentes taxas de evaporação de ferômonio, bem como o desvio padrão. Podemos concluir que os melhores resultados são obtidos quando utilizamos $e = 0.2$. A taxa de evaporação de ferômonio é utilizada para evitar a rápida convergência. Dessa forma, a taxa de evaporação de ferômonio não pode ser nem muito baixa nem muito alta. Se e for muito baixa ou muito alta, o algoritmo converge para uma solução rapidamente e não explora de forma satisfatória o espaço de busca.

No gráfico da figura 4 podemos observar os valores do custo médio, melhor custo e pior custo encontrados ao longo das iterações quando utilizamos 200 formigas, 100 iterações e taxa de evaporação de ferômonio igual a 0.2. A partir da observação do gráfico podemos concluir que os valores de custo médio e melhor custo convergem para o mesmo valor na iteração de número 100. Além disso, temos que o pior custo não possui um valor alto ao fim do ACO e possui valores baixos ao longo das iterações. Isso comprova que a abordagem de atualização **max-min** é a melhor para resolver esse problema. Se utilizássemos a abordagem clássica, usaríamos todas as formigas para atualizar o ferômonio das arestas, inclusive as formiga que geram as soluções que possuem o menor custo. Como utilizamos apenas a formiga que gera a melhor solução a cada iteração, a probabilidade de conseguirmos uma boa solução no fim do processo é maior.

4.3.2. Grafo 2

Abaixo estão descritos os experimentos realizados para o segundo grafo, bem como discussões sobre o resultado.

Para gerar os resultados para diferentes números de iterações, foram mantidos o número de formigas $n_f = 20$ e taxa de evaporação de ferômonio $e = 0.2$. Foram gerados resultados utilizando de 10 a 100 iterações e utilizando 150, 200, 250 e 300 iterações. Observando o gráfico da figura 5 podemos concluir que, para o segundo grafo, aumentar

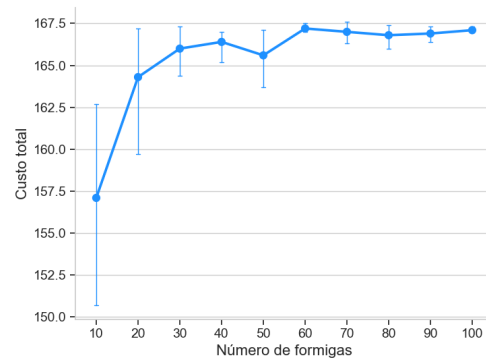
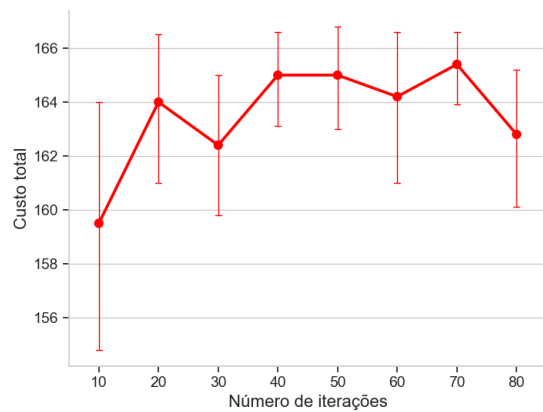


Figura 5: Valores médios do custo encontrado pelo ACO para diferentes números de iterações considerando o segundo grafo.

Figura 6: Valores médios do custo encontrado pelo ACO para diferentes números de formigas considerando o segundo grafo.



Figura 7: Valores médios do custo encontrado pelo ACO para diferentes taxas de evaporação de ferômonio considerando o terceiro grafo.

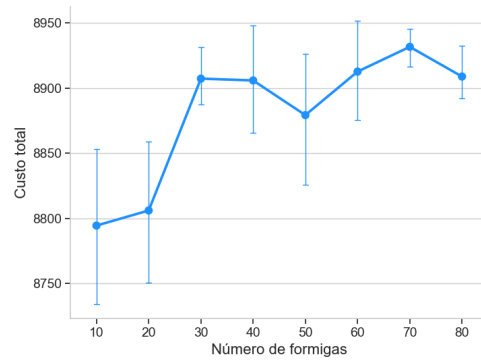
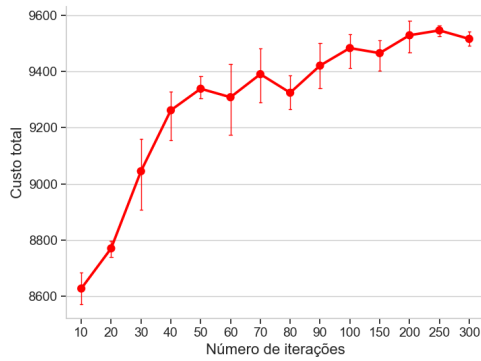


Figura 8: Valores médios do custo encontrado pelo ACO para diferentes números de iterações considerando o terceiro grafo. Figura 9: Valores médios do custo encontrado pelo ACO para diferentes números de formigas considerando o terceiro grafo.

o número de iterações não melhora de forma significativa os resultados. Além disso, podemos observar que para todos números de iterações o desvio padrão é muito alto em relação ao valor médio. Por outro lado, para o primeiro grafo quanto maior o número de iterações menor é o desvio padrão das soluções. Provavelmente, isso ocorre devido ao número de formigas utilizado ($n_f = 20$) e a distribuição de pesos das arestas.

Para gerar os resultados para diferentes números de formigas, foram mantidos $n_i = 50$ e $e = 0.2$. Foram gerados resultados utilizando de 10 a 100 formigas. Na figura 6 temos o valor médio do custo encontrado pelo ACO para diferentes número de formigas, bem como o desvio padrão. Podemos concluir que os melhores resultados são obtidos quando utilizamos 60 ou mais formigas.

Para gerar os resultados para diferentes taxas de evaporação de ferômonio, foram mantidos $n_i = 50$ e $n_f = 60$. Foram gerados resultados utilizando taxa iguais a 0.1, 0.2, 0.3, 0.4 e 0.5.

Na figura 7 temos o valor médio do custo encontrado pelo ACO para diferentes taxas de evaporação de ferômonio, bem como o desvio padrão. Diferente do que ocorre com o primeiro grafo, os resultados obtidos quando utilizamos taxa de evaporação de ferômonio iguais a 0.1, 0.2 e 0.3 são muito próximos. Ou seja, pode-se afirmar que, que o segundo grafo é menos sensível a taxa de ferômonio em relação ao primeiro grafo. Além disso, pode-se concluir que para o segundo grafo, os resultados finais obtidos quando utilizamos taxa de evaporação iguais a 0.1, 0.2 e 0.3 geram bons resultados. Novamente, isso deve ocorrer devido a distribuição de pesos das arestas.

4.3.3. Grafo 3

Abaixo estão descritos os experimentos realizados para o terceiro grafo, bem como discussões sobre o resultado.

Para gerar os resultados para diferentes números de iterações, foram mantidos o número de formigas $n_f = 10$ e taxa de evaporação de ferômonio $e = 0.2$. Foram gerados resultados utilizando de 10 a 100 iterações e utilizando 150, 200, 250 e 300 iterações.

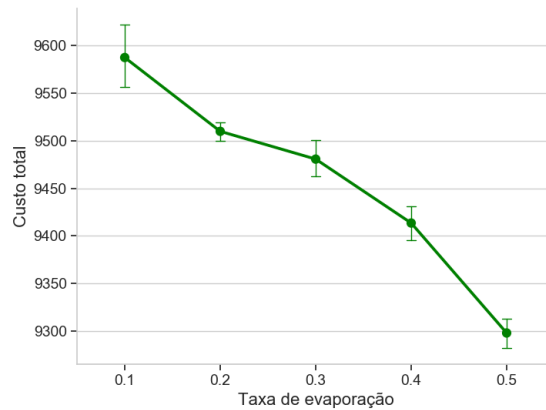


Figura 10: Valores médios do custo encontrado pelo ACO para diferentes taxas de evaporação de feromônio considerando o terceiro grafo.

Grafo	Iterações	Formigas	e	runs	Média	Max	Min	Std
1	100	200	0.2	10	977.7	983	976	2.21
2	50	60	0.1	10	167.2	168	167	0.42
3	100	20	0.1	5	9587.6	9651	9547	43.44

Tabela 1: Resultados finais para cada grafo.

Assim como observado no grafo 1, quanto maior o número de iterações maior é a chance de se obter um bom resultado final.

Para gerar os resultados para diferentes números de formigas, foram mantidos $n_i = 20$ e $e = 0.2$. Foram gerados resultados utilizando de 10 a 80 formigas. Diferente do que foi observado no grafo 1 e 2, os resultados variam muito com o número de formigas utilizadas. Dessa forma, temos que o terceiro grafo é mais sensível ao número de formigas utilizado. Pode-se observar também que para o custo final aumenta consideravelmente de 20 formigas para 30 formigas. Além disso, temos que o valor de custo encontrado quando utilizamos 30 formigas possui um baixo desvio padrão e possui valores muito próximos aos obtidos quando utilizamos mais formigas.

Para gerar os resultados para diferentes taxas de evaporação de feromônio, foram mantidos $n_i = 100$ e $n_f = 20$. Foram gerados resultados utilizando taxa iguais a 0.1, 0.2, 0.3, 0.4 e 0.5. Pode-se concluir que os melhores resultados são gerados quando utilizamos taxa de evaporação de feromônio igual a 0.1, diferente do que observamos para o primeiro e segundo grafo.

5. Resultados finais

Para cada um dos grafos foram gerados os resultados finais considerando os valores de parâmetros escolhidos para cada um deles.

A tabela 1 mostra os resultados dos experimentos para cada grafo e conjunto de parâmetros escolhido. Pode-se notar que a escolha dos parâmetros é satisfatória, uma vez que, apesar de não produzirem a solução ótima, geram um valor muito próximo de ótimo. Para o primeiro grafo, a melhor solução encontra possui custo igual a 983, 7 unidades a

menos que o da solução ótima. Por outro lado, para o segundo grafo foi possível encontrar a solução ótima que possui custo 168, o que era esperado, uma vez que o grafo é menor. Além disso, o valor médio do custo para o segundo grafo ficou muito perto do ótimo e o desvio padrão ficou muito pequena. Esses resultados mostram que a abordagem de ACO é muito efetiva para resolver o problema do caminho mais longo.

Não se conhece o custo da solução ótima para o terceiro grafo. Porém, a melhor solução encontrada utilizando o conjunto de parâmetros apresentado na tabela 1 já é melhor que os valores encontrados nos experimentos realizados. Pode-se estimar o custo do caminho mais longo para o terceiro grafo. Sabe-se que o peso das arestas dos grafos são inteiros entre 1 e 10. Assim, temos que o maior caminho mais longo para o grafo é $10 \times (|V| - 1)$, onde V é o conjunto de arestas do grafo. Como o número de arestas é 1000, temos que o custo máximo de um caminho simples é igual a $10(1000 - 1) = 9900$.

6. Conclusão

Neste trabalho implementamos um algoritmo de ACO para resolver o problema do caminho mais longo. Os parâmetros do algoritmo implementado são número de formigas, número de iterações, taxa de evaporação de feromônio, intensidade de feromônio, valor de feromônio inicial, *Alpha* e *Beta*. Para escolher os melhores parâmetros para o programa foi realizado um estudo considerando cada um deles individualmente. A escolha desses parâmetros não é uma tarefa fácil, visto que a modificação de apenas um deles pode alterar drasticamente o resultado. A realização de testes foi fundamental para entender o efeito de cada um dos parâmetros nos valores finais encontrados. Com a escolha dos parâmetros foram gerados os resultados para cada uma das três grafos disponíveis. Os melhores resultados foram obtidos para o segundo grafo e resultados satisfatórios foram alcançados para o primeiro e terceiro grafo.

Referências

- [1] E. Bonabeau, G. Théraulaz. *Swarm Smarts*. (2000).
- [2] M. Dorigo, G. Di Caro, L. M. Gambardella. *Ant Algorithms for Discrete Optimization*. (1999) 5(2): 137-72.
- [2] M. Dorigo, V. Maniezzo, A. Coloni. *The Ant System: Optimization by a colony of cooperating agents*. (1996) 26(1): 1-13.