UNIVERSIDADE FEDERAL DE MINAS GERAIS

RAPHAEL RODRIGUES CAMPOS

Heurística para o problema do caminho mais longo baseado em Ant Colony Optimization (ACO)

Segundo trabalho prático da disciplina de Computação Natural do curso de pós-graduação em Ciência da Computação do Departamento de Ciência da Computação da Universidade Federal de Minas Gerais.

Belo Horizonte
6 de Novembro de 2015

Conteúdo

1 Introdução

O problema do CAMINHO MAIS LONGO é definido a seguir. Dado um grafo direcionado e ponderado $G = (V, E)$, uma função $w : E \to \mathbb{R}$ que atribui pesos a cada aresta, um inteiro positivo K, e dois vértices $s, t \in V$. Pergunta-se : Existe um caminho simples[1] $P = \{e_1, ..., e_k\}$ em G de s para t tal que $\sum_{e_i \in P} w(e_i) = K$? Esse problema é NP-completo se o grafo possuir ciclos[2], mas para grafos acíclicos o problema está em P.

Nesse trabalho focaremos no problema de otimização correspondente. Isto é, dado os mesmos parâmetros acima, o objetivo é encontrar o caminho simples $P^* = \{e_1^*, ..., e_k^*\}$ tal que

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

onde $\mathcal{P}$ é o conjunto de caminhos simples no grafo G.

O objetivo desse trabalho é apresentar uma heurística baseada em Ant Colony Optimization(ACO) para resolver o problema supracitado. O restante desse trabalho está dividido da seguinte forma: Seção 2 detalha a implementação da solução, incluindo detalhes de representação, *fitness*, etc; Seção 3 apresenta os experimentos realizados para validação da solução; e Seção 4 apresenta a conclusão do trabalho.

2 Implementação

A inspiração para ACO é o comportamento real das formigas ao buscarem por comida. Quando buscam por comida, formigas inicialmente exploram a área ao redor de forma aleatória. Assim que uma formiga encontra uma fonte de comida, ela avalia a quantidade e qualidade de comida e carrega um pouco de volta para o ninho. Durante a viagem de volta, a formiga deposita feromônio no chão formando uma trilha. A quantidade de feromônio depositado depende da quantidade e qualidade da comida, e isso irá guiar outras formigas para a fonte de comida. A comunicação indireta entre as formigas via a trilha de feromônio permite que elas encontrem o menor caminho entre a ninho e a fonte de comida. Essa característica real das colônias de formigas é explorado em colônias artificiais de formigas, e o algoritmo ACO utilizada a representação de grafo para encontrar(aproximar) soluções para dado problema [3].

O componente central de uma algoritmo de ACO é a o modelo paramétrico probabilístico, que é chamado de modelo de feromônio. Esse modelo é usado para probabilisticamente gerar soluções para o problema em consideração reunindo-os usando um conjunto de finito de componentes da solução. Em tempo de execução, ACO atualiza os valores do feromônio usando a solução previamente gerada. A atualização objetiva concentrar o busca em regiões do espaço de busca que possuam solucoes de alta qualidade. Portanto, nós projetamos um algoritmo básico mostrado no Algoritmo 1, que funciona da seguinte forma[1].

Algorithm 1 *ACO_Longest_Path*()

```
1: InitPheromone()
2: while condição de parada não for satisfeita do
3:    GenerateSolutions()
4:    EvaluateSolutions()
5:    UpdatePheromone()
6: end while
7: return melhor solução
```

O algoritmo primeiro inicializa todos os valores do feromônio de acordo com a função *InitPheoromone*(). Então um processo iterativo inicia, com a função *GenerateSolutions*() sendo usada por todas as formigas para construir soluções probailísticas para o problema baseado em um modelo de feromônio a cada iteração. A função *EvaluateSolutions*() é usada para avaliar a qualidade das soluções construídas, e algumas das soluções são usadas pela função *UpdatePheromone*() para atualizar o feromônio antes do inicio da próxima iteração.

[1] Um caminho simples é tal que não passa pelo mesmo vértice mais de uma vez

Algorithm 2 $InitPheoromone(G = (V, E), s, t, A)$

for all $e_i \in E$ **do**
  $\tau_{e_i} = \epsilon$
end for
for all $a_i \in A$ **do**
  $n \leftarrow s$
  $a_i.P \leftarrow \{s\}$
  $Z \leftarrow V - \{s\}$
  while $n \neq t$ **and** $Z \neq \emptyset$ **do**
    $c \leftarrow n$
    Selecione aleatoriamente o nó n de acordo com a distribuição de probabilidade $P_{(c,n)}$
    $a_i.P \leftarrow a_i.P \cup \{n\}$
    $Z \leftarrow Z - \{n\}$
  end while
  if $a_i.P_1 \neq s$ **or** $a_i.P_{|a_i.P|} \neq t$ **then**
    $a_i.P \leftarrow \emptyset$
  end if
end for
Seja T o conjunto com as k formigas com os caminhos mais longos
for all $a_i \in T$ **do**
  for $j = 1$ **to** $|a_i.P| - 1$ **do**
    $e \leftarrow (v_j, v_{j+1})$
    $\tau_e \leftarrow \tau_e + \Delta\tau_e^{a_i}$
  end for
end for

A função $InitPheoromone()$ é usada para inicializar o feromônio de todas as arestas do grafo $G = (V, E)$. Inicialmente, é atribuído a cada aresta um valor pequeno de feromônio $\epsilon \neq 0$. Então, cada formiga $a_i \in A$ faz um caminhamento aleatório pelo grafo a partir do nó s. O nó inicial n é adicionado a solução, e um de seus vizinhos v aleatoriamente selecionado com probabilidade P; então um dos vizinhos de v é selecionado aleatoriamente e assim por diante, até que ela chegue ao nó t. As formigas/soluções são avaliadas de acordo com o tamanho do caminho que ela percorreu pelo grafo. As k formigas que percorreram os caminhos mais longos entre s e t são usadas para depositar o feromônio em todas arestas que compõe suas respectivas soluções. Soluções distintas podem depositar feromônio em arestas iguais, no qual todo montante de feromônio é somado. O algoritmo detalhado é mostrado no Algoritmo 2.

No processo iterativo, todas as formigas constroem probabilisticamente soluções para o problema. Na função $GenerateSolutions()$, cada formiga artificial a_i gera um solução selecionando cada aresta do caminho de acordo com uma regra probabilística de transição de estado (caminhamento aleatório em um grafo): a probabilidade de uma formiga no ponto c mover-se para o ponto n é dada por:

$$P_{(c,n)} = \begin{cases} \frac{\tau_{(c,n)}\eta_{(c,n)}}{\sum_{z \in Z} \tau_{(c,z)}\eta_{(c,z)}} & \text{Se } n \in Z \\ 0 & \text{caso contrário} \end{cases} \quad (2)$$

onde $\tau_{(c,n)}$ é a quantidade de feromônio depositado na aresta $(c, n) \in E$, $\eta_{(c,n)}$ é a "atratividade"do movimento (É usado $\eta_{(c,n)} = w(c, n)$) e Z é o conjunto de vértices para quais a formiga pode se mover partindo do ponto c.

A regra claramente favorece transições por arestas com maior quantidade de feromônio depositado e peso. O algoritmo detalhado da solução $GenerateSolutions()$ é mostrado no Algoritmo 3.

Assim, a função $EvaluateSolutions()$ é usada para avaliar o desempenho de cada formiga. O desempenho da formiga a_i é avaliado pelo tamanho L_{a_i} do caminho percorrido por ela, que é dado por:

Algorithm 3 *GeneratesSolutions*$(G = (V, E), s, t, A)$

for all $a_i \in A$ **do**
  $n = s$
  $a_i.P \leftarrow \{s\}$
  $Z \leftarrow V - \{s\}$
  while $n \neq t$ **and** $Z \neq \emptyset$ **do**
    $c \leftarrow n$
    Selecione aleatoriamente o nó n de acordo com a distribuição de probabilidade $P_{(c,n)}$
    $a_i.P \leftarrow a_i.P \cup \{n\}$
    $Z \leftarrow Z - \{n\}$
  end while
  if $a_i.P_1 \neq s$ **or** $a_i.P_{|a_i.P|} \neq t$ **then**
    $a_i.P \leftarrow \emptyset$
  end if
end for

$$L_{a_i} = \sum_{j=1}^{|a_i.P|-1} w(v_j, v_{j+1}) \tag{3}$$

que é, simplesmente, a soma dos pesos de todas as arestas do caminho simples $a_i.P$.

Após todas as formigas terem percorrido o grafo, o feromônio é atualizado. Em nosso sistema foi implementado a seguinte regra de atualização:

$$\tau_{e_i} = (1 - \rho) * \tau_{e_i} + \sum_{a_i \in T} \Delta\tau_{e_i}^{a_i}, \forall e_i \in E \tag{4}$$

Similar a função *InitPheromone*(), as k formigas que percorreram os caminhos mais longos entre s e t são usadas para depositar o feromônio em todas arestas que compõe suas respectivas soluções e todos os valores de feromônio $\Delta\tau_{e_i}^{a_i}$ da mesma aresta são somados. O parâmetro ρ é a taxa de evaporação e é implementado para evitar que o algoritmo convirja rapidamente em direção a uma região sub-ótima e quantidade de feromônio depositado é defino como:

$$\Delta\tau_{e_i}^{a_i} = \begin{cases} 1 - \frac{1}{L_{a_i}} & \text{Se a aresta } e_i \text{ faz parte da solução de } a_i \\ 0 & \text{caso contrário} \end{cases} \tag{5}$$

O algoritmo *UpdatePheromone*() detalhado é descrito no Algoritmo 4.

Algorithm 4 *UpdatePheromone*$(G = (V, E), s, t, A)$

Seja T o conjunto com as k formigas com os caminhos mais longos
$\Delta\tau[1..|E|] \leftarrow \{0, .., 0\}$
for all $a_i \in T$ **do**
  for $j = 1$ **to** $|a_i.P| - 1$ **do**
    $e \leftarrow (v_j, v_{j+1})$
    $\Delta\tau_e \leftarrow \Delta\tau_e + (1 - \frac{1}{L_{a_i}})$
  end for
end for
for all $e_i \in E$ **do**
  $\tau_{e_i} \leftarrow (1 - \rho) \times \tau_{e_i} + \Delta\tau_{e_i}$
end for

3 Experimentos

3.1 Bases de dados

Os experimentos foram rodados para três entradas distintas. O primeiro grafo possui 100 vértices e 8020 arestas. Já o segundo grafo possui 10 vértices e 190 arestas. Por fim, o terceiro grafo possui 1000 vértices e 499500 arestas. Todos os grafos são direcionados e ponderados, com pesos inteiros entre 1 e 10. Os vértices estão numerados de 1 à N.

Para os dois primeiro grafos, sabe-se a solução ótima. Para o primeiro grafo, a solução ótima é 990. Já para o segundo grafo, a solução ótima possui peso 168.

3.2 Método

A Tabela 1 mostra os parâmetros para utilizados no algoritmo ACO. Estes valores foram escolhidos a partir das conclusões da análise dos parâmetros apresentadas na seção 3.3.

	Base de dados		
Parâmetros	Entrada 1	Entrada 2	Entrada 3
Número de formigas	1000	200	1000
Número de iterações	5000	1000	5000
Taxa de evaporação (ρ)	0.1	0.01	0.1
Tam. torneio (k)	10	10	10
Elitismo	Sim	Sim	Sim

Tabela 1: Parâmetros dos experimentos

Para calibração dos parâmetros foram rodados simulações para coletar dados estatístico para decidir quais seriam os parâmetros mais apropriados a serem usados.

Os parâmetros considerados aqui são aqueles que afetam diretamente ou indiretamente a computação da probabilidade na equação (2):

- ρ : Taxa de evaporação de feromônio, $0 < \rho < 1$
- k : Quantidade de formigas que irão depositar feromônio, $k \geq 0$ se $k = 0$ então todas a formigas irão depositar;
- m : Número de formigas, $m > 0$;

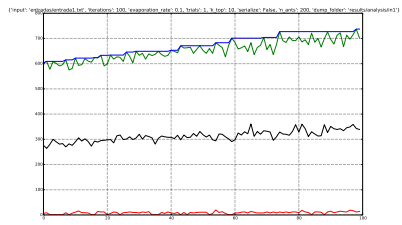
3.3 Análise dos parâmetros

Vários valores foram testados para cada parâmetro enquanto os outros eram mantidos constantes (cada configuração foi simulada 10 vezes para alcançar alguma informação estatística). Os valores padrões dos parâmetros são $m = 200$, $\rho = 0.1$ e $k = 10$. Em cada experimento apenas um dos valores era mudado. Os valores testados foram: $m = \{50, 200, 500, 1000\}$, $\rho = \{0.01, 0.1, 0.3, 0.5, 0.7\}$ e $k = \{0, 2, 10, 100\}$.

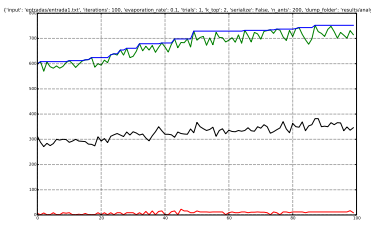
Todos os teste foram rodados usando no máximo 100 iterações e são as médias sobre 10 tentativas.

As Figuras 1, 2 e 3 mostram a evolução das formigas durante 100 iterações para as entradas 1,2 e 3 respectivamente. As curvas representam:

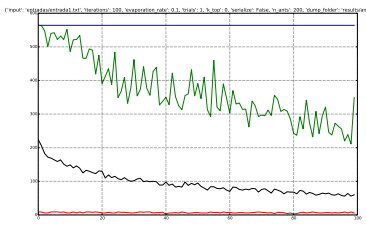
- Curva azul : *Fitness* da melhor formiga até a i-ésima iteração;
- Curva verde : *Fitness* da melhor formiga na i-ésima iteração;
- Curva preta : *Fitness* médio das formigas na i-ésima iteração;
- Curva vermelha : *Fitness* da pior formiga na i-ésima iteração;



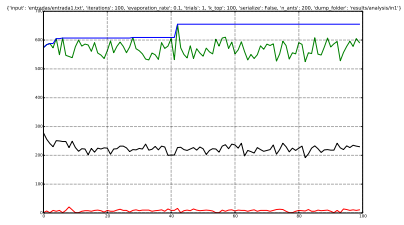
(a) m: 200/ ρ: 0.1/ k: 10



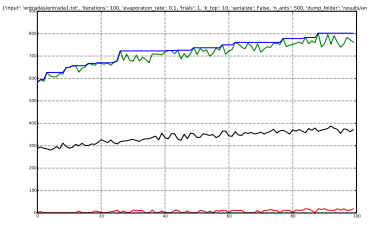
(b) m: 200/ ρ: 0.1/ k: 2



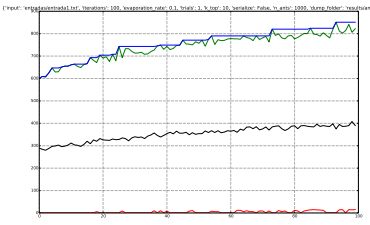
(c) m: 200/ ρ: 0.1/ k: 0



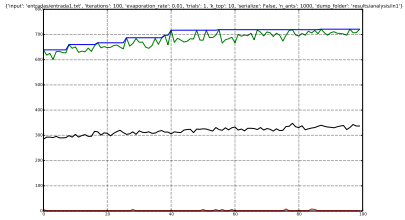
(d) m: 200/ ρ: 0.1/ k: 100



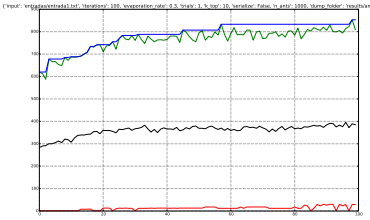
(e) m: 500/ ρ: 0.1/ k: 10



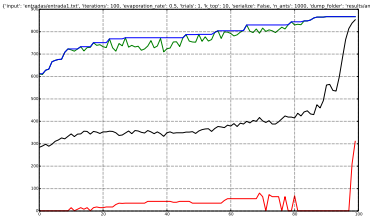
(f) m: 1000/ ρ: 0.1/ k: 10



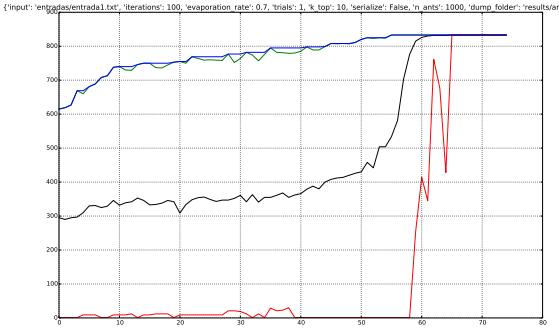
(g) Formigas: 1000/ρ: 0.01/k: 10



(h) m: 1000/ρ: 0.3/k: 10



(i) m: 1000/ ρ: 0.5/ k: 10



(j) m: 1000/ ρ: 0.7/ k: 10

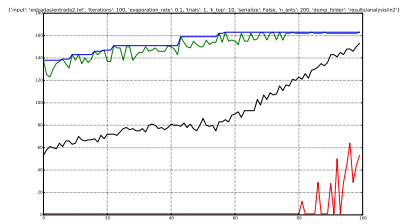
Figura 1: Análise do efeito dos parâmetros para Entrada 1. Significado das curvas no início da seção 3.3.

3.3.1 Entrada 1

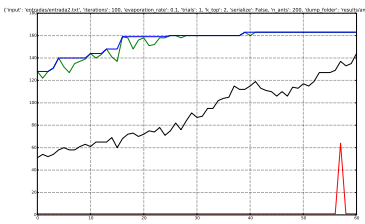
A sequência utilizada de configurações de parâmetros é mostrada na Figura 1 da direita para esquerda de cima para baixo. Inicialmente foi escolhido a configuração padrão e após ser rodado as simulações para tal, foi alterado o parâmetro k. Note-se que $k = 0$, ou seja, todas a formigas depositando feromônio, gera um resultado muito ruim. Com passar do tempo a *fitness* decresce de forma linear. Para o $k = 100$ a *fitness* do melhor individuo e a média é praticamente constante. Já o $k = 10$ e $k = 2$ produzem curvas com o comportamento mais desejável. Apesar delas serem parecidas, a configuração da Figura 1(b) tem um crescimento mais acentuado, por isso vamos fixar $k = 10$ daqui em diante. Agora que k já foi fixado, o m

será variado. Nota-se que não ha variação visível na inclinação das curvas nas Figuras 1(a),(e) e (f), desse modo $m = 1000$ foi escolhido de forma uniformemente aleatória. Finalmente, o ρ será variado e os efeitos dessa variação podem ser averiguados nas Figuras 1(f),(g), (i) e (j). Pode-se verificar, claramente, que as curvas das Figuras 1(i) e (j) convergiram precocemente e, portanto, essas configurações serão descartadas. A curva azul e verde da Figura 1(h) está chegando ao limite assintótico muito depressa se comparada com as Figuras 1(f) e (g). Desse modo, a configuração Figura 1(f) foi escolhida para os encontrar a solução do problema, já que o crescimento da curva com $\rho = 0.01$ é muito lenta.

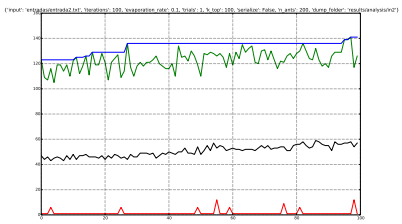
3.3.2 Entrada 2



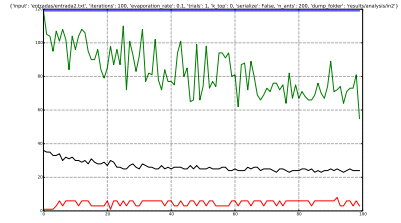
(a) m: 200/ ρ: 0.1/ k: 10



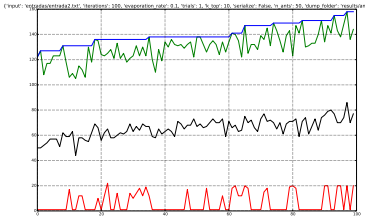
(b) m: 200/ ρ: 0.1/ k: 2



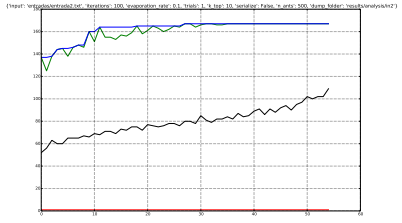
(c) m: 200/ ρ: 0.1/ k: 100



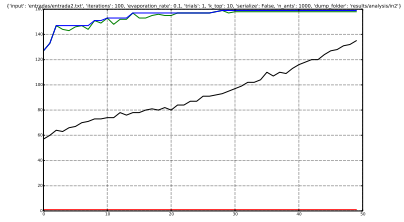
(d) m: 200/ ρ: 0.1/ k: 0



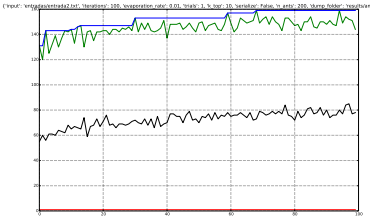
(e) m: 50/ ρ: 0.1/ k: 10



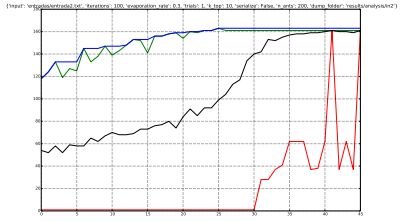
(f) m: 500/ ρ: 0.1/ k: 10



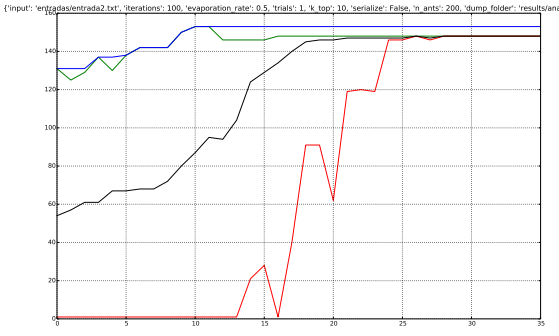
(g) m: 1000/ ρ: 0.1/ k: 10



(h) m: 200/ρ: 0.01/k: 10



(i) m: 200/ ρ: 0.3/ k: 10



(j) m: 200/ ρ: 0.5/ k: 10

Figura 2: Análise do efeito dos parâmetros para Entrada 2. Significado das curvas no início da seção 3.3.

Assim como feito para Entrada 1, inicialmente foi escolhido a configuração padrão e após ser rodado as simulações para tal, foi alterado o parâmetro k. Note-se que $k = 0$, ou seja, todas a formigas depositando feromônio, gera um resultado muito ruim. Com passar do tempo a *fitness* decresce de forma linear. Para o $k = 100$ a *fitness* do melhor individuo e a média é praticamente constante. Já o $k = 10$ e $k = 2$ produzem curvas com o comportamento mais desejável, porém é notável que há um convergência célere. Espera-se que com a alteração de outros parâmetros esse comportamento seja estabilizado. Apesar delas serem parecidas, iremos fixar $k = 10$ daqui em diante. Agora que k já foi fixado, o m será variado. Nota-se que não há variação visível na inclinação das curvas nas Figuras 2(a),(f) e (g), porém a curva da Figura 2(a) foi a única que não convergiu, portanto, $m = 200$ é fixado. Finalmente, o ρ será variado e os efeitos dessa variação podem ser averiguados nas Figuras 2(a),(h), (i) e (j). Pode-se verificar, claramente, que as curvas das Figuras 2(i) e (j) convergiram precocemente e, portanto, essas configurações serão descartadas. As curvas da Figura 2(h) estão crescendo lentamente e ,portanto, tendem a evitar convergência a um máximo local. Desse modo, a configuração Figura 2(h) foi escolhida para os encontrar a solução do problema, como mostrado na Tabela 1.

3.3.3 Entrada 3

Assim como feito para Entrada 1 e 2, inicialmente foi escolhido a configuração padrão e após ser rodado as simulações para tal, foi alterado o parâmetro k. Note-se que $k = 0$, ou seja, todas a formigas depositando feromônio, gera um resultado muito ruim. Com o passar das iterações a *fitness* decresce de forma linear. Para o $k = 100$ a *fitness* do melhor individuo e a média é decaem ligeiramente. Já o $k = 10$ e $k = 2$ tem um crescimento muito pequeno, quase imperceptível. Espera-se que com a alteração de outros parâmetros esse comportamento seja melhorado. Apesar das Figuras 3(a) e (b) serem parecidas, iremos fixar $k = 10$ daqui pra frente. Agora que k já foi fixado, o m será variado. Nota-se que não há variação visível na inclinação das curvas nas Figuras 3(a), (e), (f) e (g), porém a curva da Figura 3(g) aparenta ser a única que está crescendo, portanto, $m = 1000$ é fixado. Finalmente, o ρ será variado e os efeitos dessa variação podem ser averiguados nas Figuras 3(g),(h), (i) e (j). Pode-se verificar, claramente, que as curvas em verde e preto das Figuras 3(i) e (j) estão decaindo a partir de um determinado e, portanto, essas configurações serão descartadas. As curvas da Figura 3(g) estão crescendo lentamente e ,portanto, tendem a evitar convergência a um máximo local. Desse modo, a configuração Figura 3(g) foi escolhida para os encontrar a solução do problema, como mostrado na Tabela 1.

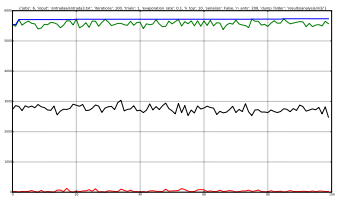
3.4 Resultados

O experimento foi rodado 30 vezes para cada uma das entradas e foi utilizado os parâmetros mostrado na Tabela 1.

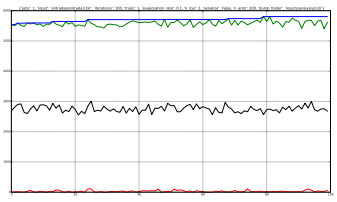
Base de dados	*Fitness* médio	Melhor *Fitness*	Pior *Fitness*	Convergiu	Num. iterações médio
Entrada 1	986.40 ± 1.02	988	985	Sim	1872.80 ± 174.41
Entrada 2	167.77 ± 0.42	168	167	Sim	98.40 ± 64.77
Entrada 3	9323.00 ± 2.45	9326	9320	Não	5000.00 ± 0.00

Tabela 2: Resultados para cada conjunto de dados

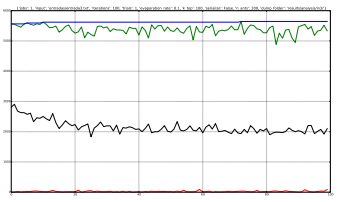
A Tabela 2 mostra o resultado do experimento supracitado utilizando os parâmetros da Tabela 1. Nota-se que do parâmetros selecionados a partir das análises feitas na seção 3.3 são satisfatórios, apesar de não ser a configuração ótima. Pode-se notar que para a Entrada 1 foi obtido um resultado bem próximo ao ótimo. A melhor *fitness* encontrada foi 988 sendo que a solução ótima do problema para Entrada 1 é 990 e foi obtido em média uma *fitness* de 986.40 que é muito próxima a solução ótima. Já para o Entrada 2 foi obtido resultados ainda melhores, o que era esperado já que o grafo é menor. A melhor *fitness* encontrada foi 168 que é exatamente o valor da solução ótima. Além disso, a *fitness* média ficou muito próxima a 168 e com uma variância pequena. Esses resultados mostram que a abordagem apresentada no trabalho tem grande efetividade na resolução do problema. Para Entrada 3, não é sabido o valor da solução ótima, porém pode-se ter uma noção da qualidade da solução dada pelo algoritmo se for levado em conta o limite superior do grafo. Por exemplo, suponha que haja um caminho simples P de s para t tal que todos os vértices em V pertençam



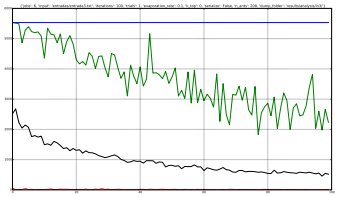
(a) m: 200/ ρ: 0.1/ k: 10



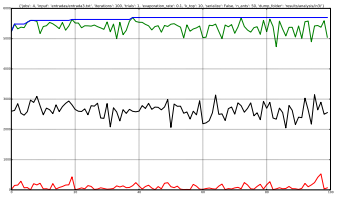
(b) m: 200/ ρ: 0.1/ k: 2



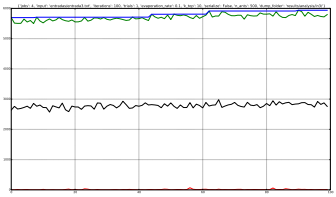
(c) m: 200/ ρ: 0.1/ k: 100



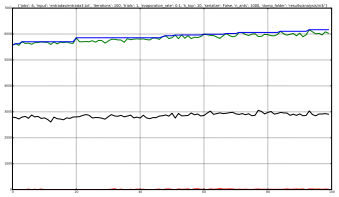
(d) m: 200/ ρ: 0.1/ k: 0



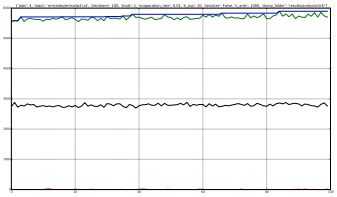
(e) m: 50/ ρ: 0.1/ k: 10



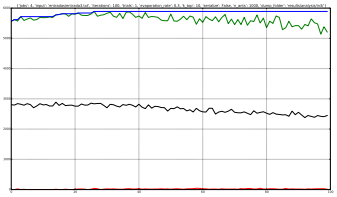
(f) m: 500/ ρ: 0.1/ k: 10



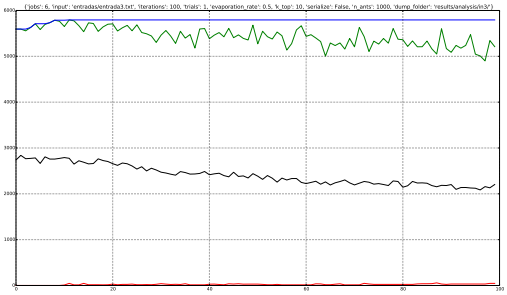
(g) m: 1000/ ρ: 0.1/ k: 10



(h) m: 1000/ρ: 0.01/k: 10



(i) m: 1000/ ρ: 0.3/ k: 10



(j) m: 1000/ ρ: 0.5/ k: 10

Figura 3: Análise do efeito dos parâmetros para Entrada 3. Significado das curvas no início da seção 3.3.

ao caminho P e que todas as arestas que formam o caminho tenham o máximo de peso possível que é 10. Assim, temos que P possui $|V| - 1$ arestas e cada aresta tem peso igual a 10. Portanto, o comprimento do caminho mais longo que grafo pode ter é no máximo igual a $10 \times (|V| - 1)$, ou seja, 9990 no caso do grafo da Entrada 3. Além disso, tomando as informações do histograma da Figura 4 pode-se notar que os pesos são distribuídos de modo uniforme entre as arestas. Desse modo, pode-se estimar um limite inferior do caminho mais longo. Suponha que as andemos de forma aleatória no grafo onde todas as arestas tem chances iguais de serem escolhidas para o caminho e que a distribuição mostrada na Figura 4 igualmente distribuída sobre as arestas de saída de todos os vértices, então podemos calcular o valor esperado para o caminho aleatório contendo todos os vértices em V como $(|V|-1)*\sum_{i=1}^{|W|} w_i * P(w_i)$, onde w_i é o i-ésimo peso dentre os possíveis $W = \{1, ..., 10\}$ e $P(w_i)$ é a probabilidade de uma dada aresta do caminho ter o peso w_i. Assim, temos que o valor esperado para o comprimento do caminhamento aleatório no grafo é 6109,53. É notório que o resultado apresentado na Tabela 2 para Entrada 3 está limitado pelo o limite inferior e superior mencionado,

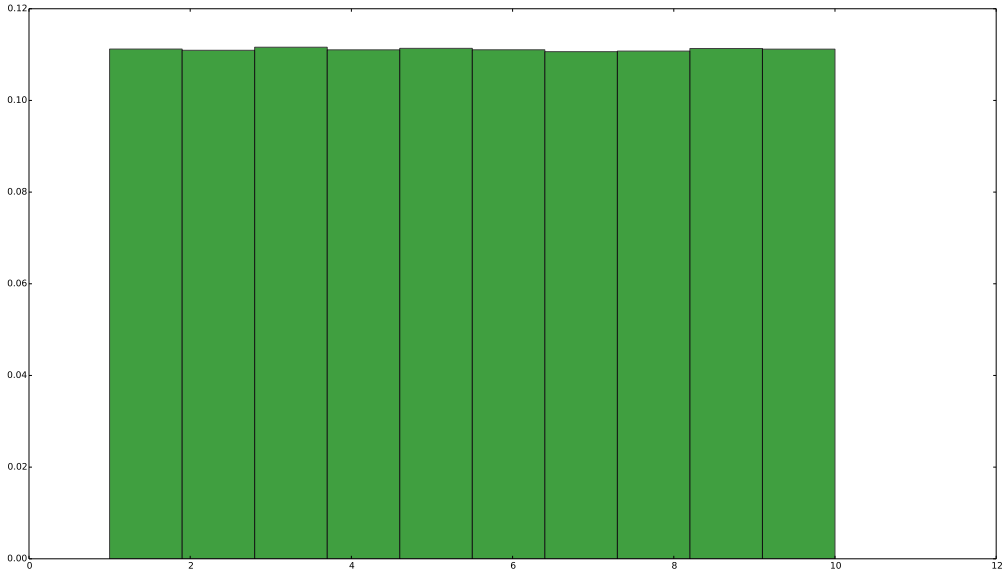


Figura 4: Distribuição dos pesos nas arestas do grafo da Entrada 3

ou seja, $6109.53 \leq 9326 \leq 9990$. Dessa forma, pode-se concluir que o resultado para Entrada 3 é satisfatório e, portanto, valida a abordagem apresentada nesse trabalho para solução do problema do CAMINHO MAIS LONGO.

4 Conclusões

Esse trabalho utilizou a capacidade de busca do algoritmo ACO para resolver o problema de CAMINHO MAIS LONGO em um grafo direcionado e ponderado. Experimentos revelaram que a abordagem proposta eficazmente encontra soluções próximas a ótima, como discutido na seção anterior. Estes resultados mostram quão poderoso pode ser o trabalho em conjunto de agentes que desempenham atividades muito simples.

Referências

[1] M. Dorigo, V. Maniezzo, and A. Colorni. The ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics–Part*, 26:1–13, 1996.

[2] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.

[3] W.-S. Yang and S.-X. Weng. Application of the ant colony optimization algorithm to the influence-maximization problem. *International Journal of Swarm Intelligence and Evolutionary Computation*, 2012.