

# UM ALGORITMO DE COLÔNIA DE FORMIGAS COM TÉCNICAS DE DISPERSÃO DE FEROMÔNIOS E MINERAÇÃO DE DADOS

**Rodolfo Ranck Jr.**

**José Carlos Becceneri**

Instituto Nacional de Pesquisas Espaciais – INPE  
Caixa Postal 515 – 12.227-010 – São José dos Campos – SP – Brasil  
rodolfoforanck@gmail.com, becce@lac.inpe.br

## RESUMO

Neste artigo, apresenta-se um método de solução baseado na meta-heurística Colônia de Formigas (ACO - *Ant Colony Optimization*) e nas técnicas de Dispersão de Feromônios e Mineração de Dados. Neste método, inicia-se uma busca utilizando um algoritmo ACO e em cada iteração, acrescenta-se feromônio nos arcos que compõem a solução e, em menores quantidades, nos arcos vizinhos. Após um determinado número de iterações, esta busca é interrompida e o conjunto de soluções gerado até o momento é analisado em busca de padrões que possam caracterizar soluções boas e ruins para o problema sendo resolvido. Com base nestes padrões a busca do ACO é intensificada nas regiões do espaço de busca em que há maiores chances de encontrar uma boa solução, e penalizada nas regiões em que há maiores chances de encontrar uma solução ruim. O método aqui proposto é comparado com o algoritmo ACO sem empregar essas duas técnicas e também empregando apenas a técnica de mineração de dados. Para o conjunto de instâncias de teste avaliado, em média, o método proposto mostrou-se capaz de obter soluções com menores valores de distância total e, na grande maioria dos casos, em um menor número de iterações.

**PALAVRAS CHAVE.** Otimização por Colônia de Formigas, Mineração de Dados, Problema do Caixeiro Viajante, Dispersão de Feromônio.

## ABSTRACT

In this paper, we present a solution method based on Ant Colony (ACO) meta-heuristic and on the techniques of Pheromones Dispersion and Data Mining. In this method a search is started using an ACO algorithm and in each iteration, it adds pheromone in the edges that compose the solution and, in smaller quantities, in neighboring edges. After a certain number of iterations, the search stops and the set of solutions generated so far is analyzed looking for patterns that might characterize good and bad solutions for the problem being solved. Based on these patterns the ACO search is intensified in the search space regions where there are greater chances of finding a good solution, and penalized in the regions where there are greater chances of finding a bad solution. The method here proposed is compared with the ACO algorithm without using both of these techniques and yet using only the data mining technique. For all evaluated testing instances, on average, the proposed method proved to be able to obtain solutions with smaller values of total distance and, in most cases, in a lesser number of iterations.

**KEYWORDS.** Ant Colony Optimization, Data Mining, Traveling Salesman Problem, Pheromone Dispersion.

## 1. Introdução

De maneira diferente do proposto tradicionalmente para algoritmos de ACO, a técnica de dispersão de feromônios baseia-se na ideia de adicionar uma quantidade de feromônio nas arestas que compõem a melhor solução encontrada e, em uma menor quantidade, nas arestas adjacentes com base em algum critério de proximidade. Esta técnica permite diversificar a busca nas vizinhanças das melhores soluções encontradas em cada iteração do ACO (veja mais em Becceneri *et al.*, 2008).

A técnica de mineração de dados aplicada ao ACO baseia-se na busca por padrões que possam caracterizar soluções boas e ruins a partir de um conjunto de soluções gerado. Essa técnica permite intensificar a busca nas vizinhanças das melhores soluções encontradas e penalizar a busca nas vizinhanças das piores soluções encontradas em cada iteração do ACO (veja mais em Becceneri *et al.*, 2009).

As técnicas dispersão de feromônios e mineração de dados para o ACO são recentes na literatura e apresentaram sucesso na melhora das soluções obtidas a partir de um algoritmo comum de ACO para o Problema do Caixeiro Viajante (c.f. Becceneri *et al.*, 2008, Becceneri *et al.*, 2009). Neste trabalho, em busca de melhores soluções para este problema, propõe-se, uma meta-heurística ACO híbrida que une estas duas técnicas.

A seguir, apresenta-se nas Subseções 1.1, 1.2, 1.3, respectivamente, uma introdução sobre o Problema do Caixeiro Viajante, Mineração de Dados e sobre a meta-heurística ACO. Nas Seções 2 e 3, apresentam-se respectivamente as técnicas de dispersão de feromônios e de mineração de dados empregadas no ACO para o desenvolvimento do método proposto. Este método será mais bem descrito na Seção 4. Na Seção 5 apresenta-se uma avaliação computacional comparando este método com um método (i) idêntico, senão por não empregar a técnica de mineração de dados aqui utilizada e também com outro método (ii) idêntico a (i), senão por não empregar a técnica de dispersão de feromônios aqui utilizada. Na Seção 6 apresenta-se uma análise dos resultados obtidos e as conclusões deste trabalho.

### 1.1. O problema do Caixeiro Viajante

O Problema do Caixeiro Viajante (TSP - *Travelling Salesman Problem*) consiste em, a partir de uma cidade qualquer, visitar todas as outras e retornar a cidade de origem, sendo que com exceção desta, todas as cidades devem ser visitadas apenas uma vez (veja Nilsson, 1982).

O TSP é um dos problemas mais investigados em otimização e atrai a atenção de diversos pesquisadores principalmente pela dificuldade em resolvê-lo, já que seu espaço de busca cresce fatorialmente com o aumento do número de cidades. De fato, em sua versão de otimização, este problema pertence à categoria NP-difícil (mais detalhes sobre complexidade podem ser encontrados em Garey e Johnson, 1978) e para resolvê-lo, é comum o uso de heurísticas como a meta-heurística ACO.

Este problema pode ser representado em um grafo, onde cidades são os nós e os caminhos entre elas são os arcos. Cada arco é valorado com um custo e a resolução deste problema consiste em encontrar neste grafo um circuito Hamiltoniano de menor custo. Neste trabalho, este custo é dado pela distância total deste circuito.

Para mais detalhes deste problema e de suas diversas aplicações, consulte, por exemplo, Lawler *et al.* (1985) e Applegate *et al.* (2006).

### 1.2. Meta-heurística Colônia de Formigas

O ACO é um método de solução heurístico para problemas de otimização inspirado no comportamento de algumas espécies de formigas na busca por alimento. O ACO foi originalmente proposto para resolver o TSP (vide Dorigo *et al.*, 1991).

Formigas são capazes de navegar em ambientes complexos em busca de comida. Nessa busca, enquanto caminham, deixam no solo certa quantidade de uma substância química, denominada de feromônio. Este feromônio é captado por outras formigas e serve

como um elo de comunicação entre elas. Na possibilidade de decidir por um conjunto de caminhos, admite-se que uma formiga escolhe com maior probabilidade caminhos que tenham maior quantidade de feromônio, depositado por outras formigas que por ali passaram. Com o decorrer do tempo o feromônio sofre evaporação e observa-se na natureza que após certo tempo a maioria delas decide pelo caminho de menor custo, possivelmente o mais curto, entre o seu ninho e a fonte de alimento (veja Figura 1). Esta forma de comunicação sinérgica utilizando o meio é denominada de stigmergia (Dorigo *et al.*, 2000).

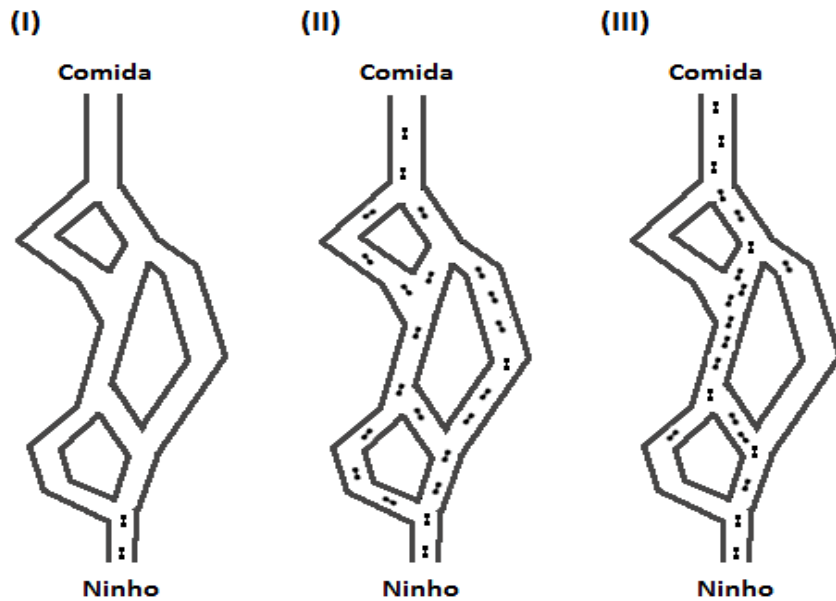


Figura 1 - (I) Formigas iniciando a busca por comida. (II) Distribuição das formigas logo após terem encontrado comida. (III) Distribuição das formigas após algum tempo movimentando-se entre o local onde está a comida e o ninho.

Diversos trabalhos têm utilizado o ACO e variações para diferentes aplicações como nos Problemas da Designação Quadrática, de Roteamento de Veículos, de Coloração de Grafos, de Programação de Operações em Sistemas de Produção, e outros (c.f. Dorigo *et al.* 1999). Para aplicar a meta-heurística ACO, é usual que o problema tratado seja modelado como um problema de percorrimento em grafos, em analogia a busca das formigas por alimento.

Seja um grafo completo e direcionado  $G(E,V)$  em que  $V$  é o conjunto dos nós e  $E$  o conjunto de arcos, para definir o ACO para o TCP, considera-se uma formiga sendo um agente que:

1. ao mover-se do nó  $i$  para o nó  $j$ , deixa uma quantidade de feromônio no caminho sendo percorrido;
2. possui uma tabela  $J$  com as cidades que pode visitar. Isto é necessário para evitar que uma formiga vá para uma cidade por onde ela já passou;
3. para selecionar um nó a ser visitado, usa uma certa função de probabilidade. Uma formiga  $k$ , estando no nó  $i$ , calcula sua probabilidade  $p_{ij}^k(t)$  de ir para o nó  $j$  na iteração  $t$  através da expressão (1).

$$p_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in J_t^k} [\tau_{il}(t)]^\alpha [\eta_{il}]^\beta} \quad (1)$$

em que:

$\tau_{ij}(t)$  é a quantidade de feromônio entre os nós  $i$  e  $j$  no instante  $t$ ;

$\eta_{ij}$  é o inverso da distância entre os nós  $i$  e  $j$ . Esse parâmetro é chamado de “desejabilidade”. No caso do TSP, quanto maior a distância entre os nós, menor é o desejo de ir de um destes nós ao outro;  $\alpha$  e  $\beta$  são parâmetros para controlar a importância, das distâncias e das quantidades de feromônio na escolha feita pelas formigas.

As quantidades de feromônio precisam ser atualizadas a cada iteração do ACO:

1. o feromônio depositado em cada arco sofre uma evaporação. Neste trabalho, seu valor é decrementado por algum percentual da quantidade de feromônio existente;
2. as quantidades de feromônio precisam ser atualizadas. Neste trabalho, acrescenta-se feromônio no melhor caminho encontrado. O valor acrescido é um percentual da quantidade de feromônio existente em cada aresta deste caminho.

Uma introdução mais detalhada ao ACO pode ser encontrada em Dorigo *et al.* (1991) e para um estudo mais aprofundado, veja, por exemplo, Bonabeau *et al.* (1999).

### 1.3. Mineração de Dados

Quando grandes quantidades de dados estão envolvidas, pode ser difícil encontrar um certo conjunto de informações, ou ainda, verificar quais informações relevantes este volume de dados guarda. Nesse contexto, Mineração de dados (DM - *Data Mining*) pode ser definida como uma caixa de ferramentas destinadas a extrair conhecimentos que estão ocultos em uma grande quantidade de dados. Tais conhecimentos podem ser úteis para entender os processos participantes na geração destes dados. Dentre estas ferramentas, há técnicas como: Classificação; Predição; Agrupamento; Sumarização; Detecção de Desvios; e Identificação de Associações.

Neste trabalho, focaliza-se a Identificação de Associações (veja mais sobre esta técnica em, por exemplo, Goldschmidt e Passos, 2005 e Larose, 2006). Emprega-se esta ferramenta na busca de padrões que possam caracterizar as soluções do problema sendo resolvido, neste caso particular, o TSP.

## 2. ACO com Dispersão de Feromônios

Uma possível maneira de modelar um algoritmo ACO é considerar que o feromônio deixado no solo pelas formigas exala um odor que se propaga pelos caminhos vizinhos influenciando a decisão das formigas que passam por estes caminhos. Denomina-se este algoritmo de ACO com Dispersão de Feromônios (ACO-DF). No ACO-DF quanto mais próximo uma formiga estiver do caminho onde o feromônio foi deixado, maior será sua percepção deste hormônio. Neste caso, entende-se que a probabilidade desta formiga ser influenciada na escolha de seu caminho, cresce com o aumento da quantidade percebida. O objetivo da dispersão de feromônios é permitir maior exploração do espaço de estados deste problema.

Do nosso conhecimento, o ACO-DF foi proposto pela primeira vez em Becceneri *et al.* (2008). Nesse trabalho, os autores apresentaram um estudo computacional comparando, entre si, um mesmo algoritmo ACO sem e com a técnica de dispersão de feromônios. As soluções obtidas quando a técnica de dispersão de feromônios foi utilizada apresentaram, em todos os casos avaliados, um menor custo médio em comparação aos casos em que ela não foi empregada.

Para modelar a influência da dispersão de feromônios em um algoritmo ACO, ao adicionar feromônio a um caminho, pode-se impor a adição de quantidades menores de feromônio nos seus arcos vizinhos. Para isto, utiliza-se uma regra *fuzzy* gradual em que, quanto mais próximo um arco vizinho está de outro onde uma quantidade de feromônio foi

depositada, maior a quantidade de feromônio recebido por ele. A seguir, apresentam-se dois conceitos fundamentais neste método:

**i) vizinhança de um arco:**

Neste trabalho, considera-se que os vizinhos de um arco direcionado A, são os arcos que contém o nó do qual este arco A parte.

**ii) distância entre dois arcos:**

Neste trabalho, assim como em Becceneri *et al.* (2008), a distância entre dois arcos vizinhos é dada pela distância Euclidiana entre o par de nós destes arcos que não pertencem a eles simultaneamente. Outras medidas poderiam também ser utilizadas como a separação angular entre estes arcos vizinhos (Becceneri *et al.*, 2008).

Em um problema grande pode existir uma quantidade muito grande de arcos vizinhos dificultando sua resolução computacional. Para evitar esta dificuldade, impõe-se que dois arcos somente podem ser vizinhos se a distância entre eles é menor que um parâmetro  $dLim$ . Para calcular este parâmetro, realiza-se o seguinte procedimento: dado uma lista com as distâncias entre todos os nós do problema, ordena-se esta lista de maneira decrescente de acordo com o valor destas distâncias. Desta lista exclui-se metade das maiores distâncias. O parâmetro  $dLim$  é dado pela média dos valores das distâncias remanescentes nesta lista multiplicado por um fator de ajuste  $pDlim$

Seja  $X$  o conjunto dos arcos que pertencem ao melhor caminho encontrado em uma iteração do ACO e que terão suas quantidades de feromônio atualizadas. A quantidade de feromônio adicionada a estes arcos e aos arcos vizinhos é dada pela relação (2).

$$\tau_{ij} = \begin{cases} \tau_{ij} + \left[ \tau_{ij} * fInc * \lambda * \left( 1 - \frac{d_{jk}}{dLim} \right) \right], & \forall i, j, k | (i, j) \notin X; (i, k) \in X; d_{jk} < dLim \\ \tau_{ij} + (\tau_{ij} * fInc), & \forall i, j | (i, j) \in X \\ \tau_{ij}, & \text{Caso contrário} \end{cases} \quad (2)$$

em que:

$fInc$  é um fator de incremento de feromônio na solução,  $fInc \geq 0$ ;

$\lambda$  é um fator de ajuste,  $\lambda > 0$ ;

$d_{jk}$  é a distância entre os nós  $j$  e  $k$ .

Na relação anterior, a primeira linha da equação é utilizada para o cálculo das quantidades de feromônio adicionadas nos arcos vizinhos da solução. A segunda linha calcula a quantidade de feromônio utilizada nos arcos que compõem a melhor solução do problema em uma dada iteração.

### 3. ACO com mineração de dados

Como observado em Becceneri *et al.* (2009), a meta-heurística ACO comum considera informações da melhor solução, ou caminho, gerado em cada iteração para determinar o local onde haverá acréscimo de feromônio. No entanto, esse método carece de uma análise dos padrões existentes nas soluções geradas. Estes padrões são sequências em caminhos que surgem com alguma frequência nas soluções do problema tratado. Neste trabalho, consideramos que padrões são sequências de dois nós em um caminho, isto é, um arco.

Estamos interessados em analisar os padrões que surgem nas soluções boas do problema (padrões bons), uma vez que eles podem indicar como construir novas soluções boas ao longo de novas iterações do algoritmo. Estamos interessados também nos padrões que

aparecem nas soluções ruins do problema (padrões ruins) para tentar evitar o surgimento deste tipo de solução em novas iterações do algoritmo.

Na meta-heurística ACO com técnica de mineração de dados (ACO-DM), toda solução gerada em uma iteração é armazenada. Após certo número de iterações ( $nItDM$ ), realiza-se uma análise no conjunto de soluções obtido até o momento em busca de padrões bons e ruins. Para isto, soma-se o número de vezes que cada aresta do problema é percorrida nas  $nSB$  melhores soluções deste conjunto (soluções boas) e armazena-se esta informação em uma estrutura de dados. O mesmo procedimento é utilizado para o caso das  $nSR$  piores soluções deste conjunto (soluções ruins) e esta informação é armazenada em outra estrutura de dados. Neste conjunto de soluções, toda aresta percorrida mais do que  $\gamma$  vezes é considerada um padrão, em que:  $\gamma$  é o número médio de vezes que uma mesma aresta pode ser percorrida em uma mesma solução considerando uma distribuição uniforme. Para as soluções boas ou ruins, quanto mais vezes um padrão é repetido, maior o fator utilizado para, respectivamente, incrementar ou decrementar o feromônio na aresta correspondente. O valor máximo incrementado ou decrementado ( $fIncDM$  ou  $fDecDM$ ) são utilizados no caso em que um padrão surge em, respectivamente, todas as soluções boas ou ruins.

A aplicação de metaheurísticas em conjunto com técnicas de mineração de dados já foi explorada anteriormente na literatura, veja, por exemplo, Ribeiro *et al.* (2004).

#### 4. ACO com Mineração de Dados e Dispersão de Feromônios

O método proposto neste artigo (ACO-DM-DF) consiste em um algoritmo de ACO que emprega os procedimentos de mineração de dados e dispersão de feromônios da maneira descrita nas Seções 3 e 4. No Anexo I deste artigo, apresenta-se o pseudocódigo deste método.

#### 5. Avaliação Computacional

Nesta seção, apresentam-se resultados computacionais comparando a meta-heurística proposta ACO-DF-DM com as meta-heurísticas ACO-DM e ACO na resolução do TSP. Essas três meta-heurísticas utilizam um mesmo algoritmo ACO, porém, adicionalmente, na primeira são empregados procedimentos de mineração de dados e dispersão de feromônios, na segunda apenas emprega-se o procedimento de mineração descrito na Seção 4 (ver Becceneri *et al.*, 2009).

Todo o código desenvolvido neste trabalho foi escrito em linguagem C++. Para a execução dos testes, utilizou-se um computador de processador Intel Celeron M, 1,73 Ghz com 2GB de memória RAM do tipo DDR2. O Compilador utilizado foi o do Microsoft Visual C++ 2008 Express Edition e o Sistema Operacional utilizado foi o Microsoft Windows 7. Para gerar números pseudoaleatórios utilizou-se a função `rand()` do ANSI C.

Para realizar os testes com o método de solução proposto utilizou-se as seguintes instâncias do TSP encontradas na literatura (veja Siqueira, 2005): u16 (16 cidades), u22 (22 cidades), u48 (48 cidades), pr76 (76 cidades) e ch150 (150 cidades). Esse conjunto de instâncias contém todas as instâncias utilizadas em Becceneri *et al.* (2009).

As Tabelas 1, 2 e 3 a seguir, apresentam respectivamente as configurações utilizadas para a meta-heurística ACO, para o procedimento de mineração de dados utilizado nas meta-heurísticas ACO-DM e ACO-DM-DF e para o procedimento de dispersão de feromônios utilizado na meta-heurística ACO-DM-DF. Os parâmetros da Tabela 1 valem para todas as heurísticas avaliadas.

Tabela 1 – Parâmetros e valores para o ACO

$nIt$	$nAnt$	$fInc$	$fDec$	$\alpha$	$\beta$
100	$2*nCidades$	0,07	0,08	1	5

Tabela 2 – Parâmetros e valores exclusivos para o procedimento de mineração de dados

<i>nItDM</i>	<i>nSB</i>	<i>nSR</i>	<i>fIncDM</i>	<i>fDecDM</i>
2	$\lceil 0,15 * nSol \rceil$	$\lceil 0,15 * nSol \rceil$	0,8	0,4

Tabela 3 – Parâmetros e valores exclusivos para o procedimento de dispersão de feromônios

$\lambda$	<i>pDlim</i>
0,722	0,5

em que:

*fDec* é o fator de decremento de feromônio na solução;

*nIt* é o número de iterações utilizadas pelo ACO;

*nAnt* é o número de formigas utilizado pelo ACO;

e os outros parâmetros conforme apresentados anteriormente.

Para os testes computacionais desta Seção, utilizou-se 25 sementes distintas para a geração de números aleatórios. Os valores destas sementes foram gerados aleatoriamente no intervalo [0,1000] com outra semente cujo valor foi 2010. Cada uma das instâncias foi resolvida 25 vezes por cada meta-heurística avaliada, e em cada uma destas 25 resoluções utilizou-se uma única semente distinta. Cada execução fornece a melhor solução nela encontrada.

Nas Tabelas (4-8), apresentam-se informações de média, desvio padrão, o maior e o menor valor dos custos das soluções fornecidas e também do número de iterações necessário para encontrar cada uma delas. Essas informações são apresentadas para cada uma das 3 meta-heurísticas comparadas. Os resultados dessas tabelas referem-se, respectivamente, às instâncias u16, u22, u48, pr76, ch150.

Tabela 4 – Resultados para a instância u16

Instância:	u16					
	ACO-DF-DM		ACO-DM		ACO	
	Iterações	Custo	Iterações	Custo	Iterações	Custo
<b>MÉDIA:</b>	45,2	74,10260	46,92	74,18866	71,28	74,23958
<b>DESVIO P.:</b>	18,46844	0,12780	23,08304	0,21820	24,95248	0,28822
<b>MAIOR:</b>	81	74,6008	96	74,6148	99	74,6148
<b>MENOR:</b>	17	73,9876	15	73,9876	26	73,9876

Tabela 5 – Resultados para a instância u22

Instância:	u22					
	ACO-DF-DM		ACO-DM		ACO	
	Iterações	Custo	Iterações	Custo	Iterações	Custo
<b>MÉDIA:</b>	59,24	75,39270	60	75,40725	87,76	75,93964
<b>DESVIO P.:</b>	13,4483	0,18190	16,86713	0,18506	11,47926	0,22306
<b>MAIOR:</b>	92	76,0319	93	75,8944	100	76,1971
<b>MENOR:</b>	37	75,3097	27	75,3097	51	75,3097

Tabela 6 – Resultados para a instância u48

Instância:	u48					
	ACO-DF-DM		ACO-DM		ACO	
	Iterações	Custo	Iterações	Custo	Iterações	Custo
MÉDIA:	64,92	34173,52	64,24	34208,032	88,92	34217,02
DESVIO P.:	16,67313	396,0909	13,59926	332,67587	8,57282	327,6311
MAIOR:	95	35362,7	88	34907,8	100	34905,4
MENOR:	43	33548,8	33	33548,8	66	33632,7

Tabela 7 – Resultados para a instância pr76

Instância:	pr76					
	ACO-DF-DM		ACO-DM		ACO	
	Iterações	Custo	Iterações	Custo	Iterações	Custo
MÉDIA:	86,52	112171,6	88,12	112299,28	92,32	115788,96
DESVIO P.:	14,19542	1127,07427	11,67376	1203,75494	9,25436	2493,47624
MAIOR:	100	115586	100	116294	100	119832
MENOR:	50	110853	49	110690	59	111236

Tabela 8 – Resultados para a instância ch150

Instância:	ch150					
	ACO-DF-DM		ACO-DM		ACO	
	Iterações	Custo	Iterações	Custo	Iterações	Custo
MÉDIA:	72,4	6720,135	78,08	6721,974	93,68	7101,542
DESVIO P.:	13,83534	23,57584	13,946087	25,62058	6,89033	149,3958
MAIOR:	99	6771,82	100	6784,93	100	7359,99
MENOR:	43	6677,55	51	6682,4	78	6895,97

## 6. Conclusão

Em todos os casos avaliados nas Tabelas (4-8), observa-se que o método proposto ACO-DM-DF obteve, em média, soluções com menor custo em comparação aos casos em que as técnicas de mineração de dados e de dispersão de feromônios não foram empregadas e aos casos em que apenas a técnica de mineração de dados foi empregada no ACO. Além disso, com exceção da instância u48, observa-se que o método proposto utiliza, em média, um menor número de iterações para encontrar a melhor solução em comparação com os demais métodos avaliados.

Para a maioria das instâncias avaliadas: o método ACO-DF-DM obteve, em média, os menores valores para o maior custo e também para o maior número de iterações; os menores valores de custo obtido pelo ACO-DF-DM não são maiores do que os respectivos valores obtidos pelos demais métodos comparados; o desvio padrão do custo das soluções avaliadas é menor para as meta-heurísticas ACO-DF-DM e ACO-DM em comparação com a meta-heurística ACO.

A união das técnicas de mineração de dados e de dispersão de feromônios como proposto mostrou poder gerar, em média, soluções melhores em comparação aos métodos aqui avaliados. Como trabalho futuro pretende-se aprimorar estas técnicas em conjunto com outras meta-heurísticas e buscar aplicá-las a outros problemas de Otimização Combinatória.



**Agradecimentos:** Aos revisores deste artigo que contribuíram para a melhoria deste trabalho e para futuros desenvolvimentos a ele relacionados. Este estudo contou com apoio financeiro do CNPq.

## Referências

- Applegate, D. L., Bixby, R. E., Chvátal, V., Cook, W. J.** The Traveling Salesman Problem: A Computational Study, Princeton University Press, 2006.
- Becceneri, J. C., Sandri, S., Pacheco da Luz, E. F.** Using ant colony systems with pheromone dispersion in the traveling salesman problem. In Proceeding of the 2008 Conference on Artificial intelligence Research and Development: Proceedings of the 11th international Conference of the Catalan Association For Artificial intelligence T. Alsinet, J. Puyol-Gruart, and C. Torras, Eds. Frontiers in Artificial Intelligence and Applications, vol. 184. IOS Press, Amsterdam, The Netherlands, 333-341, 2008.
- Becceneri, J. C., Ranck Jr., R., Coelho dos Santos, R. D.** Colônia de formigas com técnicas de Mineração de Dados. Anais do III Encontro Regional de Pesquisa Operacional. Fortaleza, Novembro de 2009.
- Bonabeau, E., Dorigo, M., and Theraulaz, G.** Swarm Intelligence: from natural to artificial systems. Oxford University Press, Inc., 1999.
- Dorigo, M., Maniezzo, V., & Colorni, A.** The Ant System: an autocatalytic optimizing process (Technical report 91-016 revised). Milan, Italy: Politecnico di Milano, 1981.
- Dorigo M., Di Caroand, G., Gambardella L.M.** Ant algorithms for discrete optimization Artificial Life, Vol.5, n.3, p.137-172, 1999.
- Dorigo M., Bonabeau E., Theraulaz G.** Ant algorithms and stigmergy. Future Generation Comp. Syst. 16(8): 851-871, 2000.
- Garey, M. R. and Johnson, D. S.** Computers and Intractability: a Guide to the Theory of Np-Completeness. W. H. Freeman & Co, 1979.
- Goldschmidt, R. and Passos, E.** Data Mining: um Guia Prático. Campus, 2005.
- Larose, D. T.** Data Mining Methods and Models. Wiley-Interscience, 2006.
- Lawler, E. L.; Lenstra, J. K.; Kan, a. h. g. r.; Shmoys, D. B.** The Traveling Salesman Problem: a guided tour of combinatorial optimization. Chichester, UK. A Wiley interscience Publication, John Wiley & Sons, p. 251-305, 1985.
- Nilsson, N. J.** Principles of Artificial Intelligence. Morgan Kaufmann Publishers Inc, 1980.
- Ribeiro, M.H.F.; Trindade, V. F.; Plastino, A.; Martins S.L.** Hybridization of GRASP metaheuristic with data mining techniques, Proceedings of ECAI Workshop on Hybrid Metaheuristic, pp. 69-78, 2004.

## Anexo I

A seguir, apresenta-se o pseudocódigo do método aqui proposto e dos procedimentos de mineração de dados e de dispersão de feromônios utilizados, vide Tabelas 9, 10 e 11.

Tabela 9 – Pseudocódigo do método principal para o ACO

```
Início_Método principalACO-DF-DM()  
  Leia os parâmetros do problema e inicialize as variáveis utilizadas neste método;  
  Inicialize a matriz de feromônios;  
  Distribua as formigas entre as cidades;  
  para  $t=0$  até  $nIt-1$  Faça  
    para  $i=0$  até  $nCidades-1$  Faça  
      para  $k=0$  até  $Formiga_i-1$  Faça  
         $geraCaminho(i, solucao)$ ; //gera um circuito Hamiltoniano a partir de  $i$   
         $cSolucao=verificaCusto(solucao)$ ; //obtem o custo do caminho gerado;  
        Se  $cSolucao < menorCusto$   
           $menorCusto=cSolucao$ ;  
        Fim_Se  
        Armazene a solução em uma estrutura de dados  
      Fim_Para  
    Fim_Para  
    Incremente a matriz de feromônio nos arcos pertencentes à melhor solução,  
    por um fator ( $fInc$ ) definido como um parâmetro de entrada;  
    Decremente a matriz de feromônio nos arcos não pertencentes à melhor  
    solução, por um fator ( $fDec$ ) definido como um parâmetro de entrada;  
    Execute o procedimento algoritmoDF();  
    Se o resto da divisão  $t/nItDm$  for zero Faça  
      Execute o procedimento algoritmoDM();  
    Fim_Se  
  Fim_Para  
Fim_Método
```

Tabela 10 – Pseudocódigo do procedimento de mineração de dados utilizado

```
Início_Método algoritmoDM ()  
  Ordene as soluções armazenadas de forma decrescente de acordo com seu custo  
  associado;  
  Defina os grupos de soluções boas e ruins sendo, respectivamente, as  $nSB$  primeiras  
  e as  $nSR$  últimas do grupo de soluções ordenado;  
  Para toda aresta pertencente ao grupo de soluções boas e para toda aresta  
  pertencente ao grupo de soluções ruins armazene, em estruturas de dados distintas, o  
  número de vezes que cada uma delas é repetida nos respectivos grupos de soluções;  
  Atualize a matriz de feromônio com base nos dados armazenados no passo anterior  
  de acordo com o procedimento descrito na Seção 3 deste artigo;  
Fim_Método
```

Tabela 11 – Pseudocódigo do procedimento de dispersão de feromônios utilizado

```
Início_Método algoritmoDF ()  
  Atualize os feromônios dos arcos conforme relação (2) da Seção 2.  
Fim_Método
```