

# Algoritmos de Inteligência de Enxames por Colônia de Formigas na Análise de Indicadores Ambientais de Bacias Hidrográficas

Samuel Ghellere, Gabriel Felipe, Maicon B. Palhano, Ruano M. Pereira, Kristian Madeira,  
Carlyle T. B. de Menezes, Merisandra C. de Mattos Garcia

Curso de Ciência da Computação  
Universidade do Extremo Sul Catarinense  
Criciúma, Brasil  
{mem, maiconpalhano, ruano}@unesc.net

**Resumo**—Tecnologias atuais de armazenamento permitem gerar grandes bases de dados, as quais podem possuir informações desconhecidas. Dessa forma, são necessárias tecnologias voltadas a explorar essas bases, tendo-se o *data mining* como uma dessas tecnologias, que utiliza algoritmos para extrair conhecimento e padrões presentes nos dados. Este artigo demonstra a aplicação do Standard Ant Clustering Algorithm, Ant Based Clustering e Adaptative Ant-Clustering Algorithm, pertencentes à tarefa de agrupamento, para análise de dados referente aos indicadores do monitoramento de bacias hidrográficas da região carbonífera do Sul de Santa Catarina. A tarefa de agrupamento objetiva reunir uma base de dados em grupos de dados semelhantes, sendo que os algoritmos aplicados nessa pesquisa simulam o comportamento da organização de cemitérios observados em espécies de formigas.

**Palavras Chave**—*Data mining; Agrupamento; Colônia de Formigas; Standard Ant Clustering Algorithm; Ant Based Clustering; Adaptative Ant-Clustering Algorithm.*

## I. INTRODUÇÃO

O avanço da tecnologia facilita o armazenamento de dados e permite gerar grandes bases de dados. Devido ao tamanho dessas bases de dados a análise das informações tornou-se complexa para a capacidade humana. Em vista disso, técnicas capazes de procurar e extrair informações significativas dos dados foram criadas. Essa procura de relações entre os dados ficou conhecida como *Knowledge Discovery in Databases* (KDD), sendo o *data mining* a principal etapa desse processo.

Data mining é a exploração e análise, por meio automático ou semi-automático, de grandes quantidades de dados a fim de descobrir padrões [1]. Para a execução do *data mining* são necessários métodos que são disponibilizados em ferramentas computacionais específicas, sendo que dentre estas existe a Shell Orion Data Mining Engine, que consiste em um projeto acadêmico que encontra-se em desenvolvimento. A Shell Orion possui implementadas, atualmente, as tarefas de classificação, associação e agrupamento por meio de diferentes algoritmos.

A tarefa de agrupamento tem como objetivo reunir dados de um conjunto de elementos de forma que os grupos

formados, denominados *clusters*, apresentem a maior similaridade possível, com os dados do mesmo *cluster*. Entre os algoritmos de agrupamento, os baseados em inteligência de enxame são inspirados no comportamento coletivo de insetos sociais, e possuem, como ponto positivo, flexibilidade, robustez e auto-organização.

O comportamento coletivo desses insetos sociais, inspirou a ciência a fazer simulações computacionais. Há pelo menos dois motivos para isso, o primeiro é que os mecanismos responsáveis pelo comportamento são ainda desconhecidos e pela simulação podem ser estudados a fim de dar melhor entendimento à natureza e o segundo motivo é que os comportamentos coletivos possuem características de robustez e confiabilidade o que os tornam atraentes para pesquisas científicas [2]. Um estudo específico da formiga *Pheidole pallidula* demonstrou que elas organizam seus corpos mortos em grupos semelhantes o que resultou no desenvolvimento do algoritmo de agrupamento Standard Ant Clustering Algorithm (SACA), o qual consiste em uma simulação de colônia de agentes de formigas que se movem aleatoriamente em uma grade toroidal (sem extremidade) bidimensional, onde os objetos estão posicionados. A principal vantagem desse método é o fato de que não necessita nenhuma informação inicial a respeito da massa de dados que será particionada [3].

Nesse artigo apresenta-se a implementação no módulo de agrupamento da Shell Orion e aplicação do algoritmo baseado em inteligência de enxame SACA proposto por Lumer e Faieta (1994) [3] e aos aperfeiçoamentos propostos por Handl (2003) [4], Ant Based Clustering, e ao algoritmo adaptativo, Adaptative Ant-Clustering Algorithm (A<sup>2</sup>CA), proposto por Vizine et al (2005) [5].

## II. A TAREFA DE AGRUPAMENTO EM DATA MINING

Esta tarefa tem como principal objetivo dividir o conjunto de dados em grupos, de forma que os elementos de cada *cluster* compartilhem propriedades semelhantes os diferenciando dos outros subconjuntos [1].

O agrupamento frequentemente é usado como a primeira tarefa no processo de extração de padrões, podendo o seu resultado servir de entrada para alguma outra tarefa. Mediante isso, o agrupamento pode facilitar e melhorar o desempenho de outros algoritmos de *data mining* no processo de descoberta de conhecimento [6].

### III. INTELIGÊNCIA DE ENXAME

Denomina-se inteligência de enxame a tentativa de desenvolvimento de algoritmos inspirados no comportamento coletivo de colônia de insetos sociais e outras sociedades de animais. Esses algoritmos são caracterizados pela interação com um grande número de agentes que percebem e modificam seu ambiente localmente [7].

Pode-se exemplificar a inteligência de enxame por meio dos processos naturais de construção de ninhos pelos cupins ou sociedade de abelhas, atividade forrageadora das formigas, divisão de trabalho e alocação de tarefas nas colônias de formiga, transporte cooperativo de comida, seleção de fontes com melhor néctar pelas abelhas, entre outros.

### IV. STANDARD ANT CLUSTERING ALGORITHM

O algoritmo SACA foi apresentado por Lumer e Faieta, segundo eles, no algoritmo SACA as formigas e os objetivos (conjunto de dados) são espalhados em uma grade bidimensional [3]. A cada iteração uma formiga é selecionada aleatoriamente e pode pegar ou deixar um objeto no seu local atual uma vez que, respectivamente: há um objeto naquele local, ou que a formiga esteja carregando um objeto e o lugar esteja vazio. Supondo que o agente encontrou um objeto, a probabilidade da formiga pegar esse objeto diminui com a similaridade dos demais objetos posicionados ao seu redor.

Além da função da vizinhança que permitiu que as formigas identificassem objetos ao seu redor, Lumer e Faieta introduziram no algoritmo SACA os seguintes conceitos: memória de curta duração e população heterogênea de agentes [3].

A memória de curta duração possibilita a formiga lembrar os últimos itens que foram movimentados, assim como os respectivos locais em que foram deixados. Isto permite que a formiga movimente o objeto atual para o local onde foi deixado o objeto com o qual mais se assemelha. Essa alteração acelera significativamente o agrupamento, pois a posição *best matching* memorizada, que é a posição de mínima dissimilaridade, será usada para indicar o sentido da direção aleatória da formiga [3].

A população homogênea de agentes limita o tamanho do passo que uma formiga pode dar no seu movimento aleatório, sendo que o passo corresponde a quantidade de células da grade bidimensional que a formiga é capaz de caminhar em um movimento. Com esse ajuste tem-se formigas mais rápidas e que podem realizar passos maiores na grade, o que otimiza a velocidade de agrupamento do algoritmo [3].

### V. ANT BASED CLUSTERING

A existência de dois problemas no algoritmo SACA foram verificadas por Handl [4]. Um destes problemas era devido ao

resultado apresentado pelo algoritmo: ele não gera partições, mas sim uma distribuição dos elementos em uma representação gráfica. Outro problema identificado foi a dificuldade de se ajustar os diversos parâmetros do algoritmo para diferentes tipos de dados.

Visando aperfeiçoar o algoritmo, foram introduzidas as seguintes alterações: adaptação da função vizinhança, raio de percepção crescente, separação espacial, vizinhança ponderada, modificação na probabilidade de pegar e deixar um item e adaptação de  $\alpha$ .

### VI. ADAPTIVE ANT-CLUSTERING ALGORITHM

Uma das dificuldades de se aplicar o algoritmo SACA em problemas complexos é devido ao fato do algoritmo gerar um número excessivo de grupos que não correspondem ao número efetivo [7].

A fim de superar a dificuldade citada, foi proposto o algoritmo adaptativo, Adaptive Ant-Clustering Algorithm (A<sup>2</sup>CA). Com isso, foram introduzidas no algoritmo original, SACA, três alterações principais: uma rotina de adaptação do valor do parâmetro  $k$ , um campo de visão que se adapta aos grupos formados durante o processo do algoritmo, e utilização de uma heurística de feromônio.

### VII. ALGORITMOS DE INTELIGÊNCIA DE ENXAMES POR COLÔNIA DE FORMIGAS NA TAREFA DE AGRUPAMENTO DA SHELL ORION DATA MINING ENGINE

A implementação dos algoritmos SACA, Ant Based Clustering e A<sup>2</sup>CA iniciaram-se com a construção dos diagramas de caso de uso, atividades e sequência utilizando padrões UML. Posteriormente foi desenvolvida a demonstração matemática dos funcionamentos dos algoritmos com a finalidade de facilitar o entendimento e a sua implementação.

Os algoritmos necessitam dos seguintes parâmetros:

- Número de formigas: parâmetro que determina quantas formigas irão trabalhar para agrupar os objetos na grade, esse parâmetro é informado nos algoritmos SACA, A<sup>2</sup>CA e Ant Based Clustering.
- Total de iterações: quantidade de ciclos do algoritmo, informado no SACA, A<sup>2</sup>CA e Ant Based Clustering.
- Memória: quantidade de posições que a formiga consegue lembrar, informado nos algoritmos SACA, A<sup>2</sup>CA e Ant Based Clustering.
- Alfa  $\alpha$ : parâmetro que determina a dissimilaridade na formação dos grupos, informado nos algoritmos SACA, A<sup>2</sup>CA.
- $kp$  e  $kd$ : constantes utilizadas no cálculo das probabilidades, informadas nos algoritmos SACA e A<sup>2</sup>CA.
- Campo de visão: número de células que uma formiga consegue perceber ao seu redor, informado no algoritmo SACA.

- Número de passos: esse valor é aleatório em todos os algoritmos e representa a quantidade de células que uma formiga consegue se movimentar.

Após a parametrização os algoritmos constroem a matriz de dissimilaridade, que é montada calculando a distância euclidiana n-dimensional. Essa matriz é quadrada e de tamanho  $n \times n$  que é definido pela quantidade de objetos a serem agrupados. A seguir apresenta-se uma matriz de dissimilaridade [3]:

$$\begin{bmatrix} 0 & & & & & & & \\ dist(2,1) & 0 & & & & & & \\ dist(3,1) & dist(3,2) & 0 & & & & & \\ \vdots & \vdots & \vdots & 0 & & & & \\ dist(n,1) & dist(n,2) & \dots & \dots & 0 & & & \end{bmatrix} \quad (1)$$

A matriz de dissimilaridade é montada calculando a distância euclidiana n-dimensional conforme a fórmula:

$$\delta(i, j) = \sqrt{|x_{il} - x_{jl}|^2 + |x_{i2} - x_{j2}|^2 + \dots + |x_{in} - x_{jn}|^2} \quad (2)$$

Com todas as distâncias calculadas é realizada a normalização para que os valores fiquem entre 0 e 1. Para isso usou-se a normalização MIN-MAX:

$$Z_i = \frac{V_i - \min_{v_i}}{\max_{v_i} - \min_{v_i}} (n_{\max} - n_{\min}) + n_{\min} \quad (3)$$

Cada linha dessa matriz representa um objeto que será carregado pelas formigas durante o processo do algoritmo e sempre que a função vizinhança for calculada essa matriz deverá ser consultada.

Com os parâmetros de entrada definidos, a primeira fase dos algoritmos é iniciada, eles espalham os dados na grade bidimensional que corresponde a uma matriz  $n \times n$ . Com todos os objetos espalhados na grade, os algoritmos aleatoriamente carregam cada formiga com um objeto, posicionando-a no mesmo lugar do objeto que ela irá carregar. Dessa forma a segunda fase, que corresponde ao agrupamento é iniciada.

Nessa fase uma formiga é escolhida aleatoriamente para iniciar o processo. Verifica-se se ela está carregando um objeto e calcula-se a probabilidade de deixar o objeto carregado na vizinhança atual da formiga, caso for verdadeiro a formiga deixa o objeto e procura um outro livre na grade, dessa forma, calcula-se a probabilidade de pegar esse objeto. Quando uma formiga não consegue deixar o objeto na sua posição atual, ela se movimenta aleatoriamente na grade considerando o número de passos que a mesma foi parametrizada e outra formiga é selecionada para continuar o processo de agrupamento. Quando uma formiga não consegue pegar um objeto ela seleciona outro aleatoriamente até conseguir pegar algum objeto para prosseguir na execução do algoritmo.

A função que verifica a vizinhança da formiga é determinada como:

$$f(i) = \begin{cases} \frac{1}{\sigma^2} \sum_j \left( 1 - \frac{\delta(i, j)}{\alpha} \right) & \text{se } f(i) > 0 \\ 0 & \text{em outros casos} \end{cases} \quad (4)$$

Essa função é utilizada nos três algoritmos sendo que no *Ant Based Clustering* ela sofre modificações. Em determinado momento ela deixa de considerar o valor de  $\sigma$  e passa a utilizar no lugar do mesmo, o número de objetos encontrados na vizinhança. Além disso, também possui uma restrição adicional que penaliza altas dissimilaridades entre os objetos. Essa função passa a ser determinada como:

$$f^*(i) = \begin{cases} \frac{1}{N_{occ}} \sum_j \left( 1 - \frac{\delta(i, j)}{\alpha} \right) & \text{se } f(i) > 0 \wedge \left( 1 - \frac{\delta(i, j)}{\alpha} \right) > 0 \\ 0 & \text{em outros casos} \end{cases} \quad (5)$$

As probabilidades de pegar e deixar um objeto também variam entre os algoritmos. Para o SACA as equações são as seguintes, respectivamente:

$$P_d(i) = \begin{cases} 1 & \text{se } f(i) \geq k_d \\ 2 * f(i) & \text{em outros casos} \end{cases} \quad (6)$$

$$P_p(i) = \left( \frac{k_p}{k_p + f(i)} \right)^2 \quad (7)$$

No algoritmo *Ant Based Clustering*, as constantes  $k_p$  e  $k_d$  são descartadas utilizando apenas o valor da função vizinhança para calcular as duas probabilidades. As equações das probabilidades de pegar e deixar um objeto são determinadas da seguinte forma, respectivamente:

$$p_p(i) = \begin{cases} 1 & \text{se } f(i) \leq 1 \\ \frac{1}{f(i)^2} & \text{em outros casos} \end{cases} \quad (8)$$

$$p_d(i) = \begin{cases} 1 & \text{se } f(i) \geq 1 \\ f(i)^4 & \text{em outros casos} \end{cases} \quad (9)$$

Já o algoritmo A<sup>2</sup>CA possui uma heurística de feromônio que acrescenta o valor na célula onde está sendo deixado um objeto. Esse valor é considerado nas equações das probabilidades. Essas equações são determinadas das seguintes formas no A<sup>2</sup>CA:

$$p_d = \frac{1}{f(i)\phi(i)} \left( \frac{f(i)}{f(i) + k_d} \right)^2 \quad (10)$$

$$p_p = f(i)\phi(i) \left( \frac{k_p}{f(i) + k_p} \right)^2 \quad (11)$$

Onde o valor de feromônio é representado por  $\phi(i)$ .

O parâmetro  $\alpha$  no algoritmo *Ant Based Clustering* possui uma adaptação, esse valor é modificado sempre que a formiga falhar ao tentar deixar um objeto, essa adaptação é dada por:

$$a \leftarrow \begin{cases} a + 0.01 & \text{se } r_{falhas} > 0.99 \\ a - 0.01 & \text{se } r_{falhas} \leq 0.99 \end{cases} \quad (12)$$

Onde o  $r_{falhas}$  consiste no número de vezes que a formiga falhou em deixar o objeto dividido por 100. Já o campo de

visão inicia com o valor 3 e no final das iterações esse valor passa a ser 5, conforme sugerido na literatura [4] [8].

No algoritmo A<sup>2</sup>CA o campo de visão é baseado na função vizinhança, quando o valor desta é maior que um valor determinado  $\theta$ , o campo de visão da formiga é acrescentado 2 unidades até um valor máximo de 7 e a função vizinhança é recalculada, conforme sugerido na literatura [5]. Já o parâmetro  $k_p$  possui um resfriamento que acontece a cada 10000 ciclos de formigas, esse resfriamento é dado por:

$$\begin{cases} k_p \leftarrow k_p * 0.98 \\ k_{p_{min}} = 0.001 \end{cases} \quad (13)$$

Onde  $k_{p_{min}}$  é o máximo de resfriamento do parâmetro  $k_p$ .

### VIII. IMPLEMENTAÇÃO

Os algoritmos dessa pesquisa foram desenvolvidos no módulo de agrupamento da Shell Orion Data Mining Engine, por meio da linguagem de programação Java e do ambiente de programação integrado NetBeans 7.0.1.

A validação dos resultados do agrupamento gerado pelos algoritmos foi realizada pelo uso de índices estatísticos. Foi utilizado o índice de Dunn proposto por J.C. Dunn em 1974 e o C-Index proposto por L.J. Hubert e J. R. Levin em 1974.

O índice Dunn tem como objetivo avaliar as partições para identificar o quanto os *clusters* encontrados são compactados e bem separados [9].

$$Dunn = \min_{l < i < k} \left( \min_{i+l \leq j \leq k} \left( \frac{dist(c_i, c_j)}{\max diam(c_i)} \right) \right) \quad (14)$$

O índice C-Index apresenta valores para cada *cluster* encontrado separadamente, seus resultados estão no intervalo de [0,1], onde quanto mais próximo de zero for o resultado melhor é considerado o agrupamento [10].

$$C - Index = \frac{d_w(c_i) - \min(n_w)}{\max(n_w) - \min(n_w)} \quad (15)$$

$$d_w(C_i) = \sum_{x_i, x_j \in C_i} dist(x_i, x_j) \quad (16)$$

### IX. RESULTADOS OBTIDOS

Os algoritmos implementados SACA, Ant Based Clustering e A<sup>2</sup>CA, foram aplicados a dados de monitoramento da bacia hidrográfica do rio Urussanga, composta por 20 atributos e 425 registros coletados ao longo de 37 pontos de monitoramentos espalhados por essa bacia.

O objetivo dos testes realizados por meio da aplicação dessa base de dados, consistiu na análise da qualidade dos *clusters* formados pelos algoritmos por meio dos índices de validação, a quantidade de *cluster* que cada algoritmo forma e o tempo que cada um necessita para formar os *clusters*, além da demonstração da variação dos resultados por meio do desvio padrão, média e coeficiente de variação. Na definição dos parâmetros empregados foram necessários vários testes até encontrar a parametrização ideal. Os três algoritmos foram executados 10 vezes cada um. Nos testes pode-se observar que alguns algoritmos, como por exemplo, Ant Based Clustering

necessitam de mais iterações para começar a formar *clusters*, devido a isso se tem a diferença desses parâmetros na execução dos testes.

#### A. Algoritmo SACA

Na avaliação dos resultados gerados pelo algoritmo SACA, os parâmetros utilizados foram os seguintes:

- a) número de formigas: 50;
- b) total de iterações: 2500;
- c) memória: 50;
- d) alfa: 0.3;
- e)  $k_p$ : 0.1;
- f)  $kd$ : 0.3
- g) campo de visão: 5;
- h) atributo de entrada: pH e fe.

Na tabela I pode-se observar um dos resultados obtidos nos testes realizados com o algoritmo SACA.

TABELA I. RESULTADO DE UM DOS TESTES REALIZADOS COM O ALGORITMO SACA.

Teste	$C^a$	PNC <sup>b</sup>	Tempo	Índice de Dunn	Índice C-Index
1	6	21	00m: 16s: 422ms	1,3518	C1 = 0,0294 C2 = 0,1631 C3 = 0,1966 C4 = 0,0912 C5 = 0,0800 C6 = 0,1355

<sup>a</sup> Clusters.  
<sup>b</sup> Pontos Não Classificados.

Conforme a tabela I o *C-index* apresentou resultados próximos de zero, indicando que os *clusters* formados estão compactados, porém o número de *clusters* poderia ser menor de acordo com a base de dados utilizada.

A tabela II apresenta a variação dos resultados obtidos nos testes.

TABELA II. VARIAÇÃO DOS RESULTADOS DO ALGORITMO SACA.

Medidas	Clusters	Pontos não classificados	Índice de Dunn
Desvio padrão	1,9544	3,9357	0,0465
Média	6,1	21,9	1,367
Coefficiente de variação	32,0393%	17,9712%	3,4081%

Pode-se observar na tabela II que houve uma grande variação nos números de *clusters* gerados pelo SACA, comprovando que o mesmo gera resultados diferentes a cada execução, porém o índice de validação *C-index*, que avalia internamente cada *cluster*, teve em todos os testes realizados valores próximos de zero o que indica que mesmo havendo variação no número de clusters gerados, esses são formados por objetos similares.

#### B. Algoritmo Ant Based Clustering

Na avaliação dos resultados gerados pelo algoritmo Ant Based Clustering, os parâmetros utilizados foram os seguintes:

- a) número de formigas: 10;
- b) total de iterações: 100000;
- c) memória: 10;
- d) atributo de entrada: pH e fe.

Os parâmetros *número de formigas* e *memória* utilizados foram os mesmos sugeridos como ideais pela autora do algoritmo, apenas o parâmetro total de iterações foi ajustado de acordo com os testes realizados previamente.

Na tabela III pode-se observar um dos resultados gerados pelo algoritmo nas 10 execuções.

TABELA III. RESULTADO DE UM DOS TESTES REALIZADOS COM O ALGORITMO ANT BASED CLUSTERING

Teste	CF <sup>a</sup>	PNC <sup>b</sup>	Tempo	Índice de Dunn	Índice C-Index
1	5	12	02m: 50s: 798ms	1,3518	C1 = 0,0998 C2 = 0,1712 C3 = 0,2796 C4 = 0,1108 C5 = 0,0897

<sup>a</sup> Clusters.  
<sup>b</sup> Pontos Não Classificados.

Nos testes pode-se observar que o algoritmo necessita de muitas iterações para começar a formar *clusters*. A utilização de um baixo Campo de Visão e a alteração das equações  $k_p$  e  $k_d$  do algoritmo possibilita a formação de grupos bem compactos, porém o tempo de formação dos grupos é maior comparando com o algoritmo SACA.

A adaptação do Alfa  $\alpha$  tem efeitos positivos para o algoritmo, pois com essa adaptação possibilita que elementos de um grupo possam ser analisados novamente e movidos para outros grupos, onde a vizinhança é mais similar a ele conforme vão passando as iterações.

No momento que o número de iteração do algoritmo ultrapassa o valor de início e o algoritmo começa a utilizar a outra função vizinhança, já apresentada, acontece um espalhamento dos grupos, e quando termina essa etapa os grupos são formados novamente, porém em locais diferentes.

Entre as vantagens apresentadas por este algoritmo em relação ao SACA, destaca-se a facilidade de parametrização, pois o mesmo necessita somente da informação dos parâmetros *número de formigas*, *total de iterações* e *memória*. A adaptação de  $\alpha$  elimina o problema de determinar esse parâmetro, que é crucial para a boa formação dos grupos.

Na tabela IV pode-se observar o desvio padrão, média e coeficiente de variação dos resultados obtidos pelo algoritmo.

TABELA IV. VARIAÇÃO DOS RESULTADOS DO ALGORITMO ANT BASED CLUSTERING

Medidas	Clusters	Pontos não classificados	Índice de Dunn
Desvio padrão	1,7349	1,5132	0,1591
Média	6,3	11,1	1,4031
Coefficiente de variação	27,5381%	13,6324%	11,3391%

Analisando a tabela IV, pode-se verificar que o algoritmo Ant Based Clustering tem uma variação de números de

*clusters* e pontos não classificados um pouco menor que o SACA. Na tabela 3, comparando com o SACA, o tempo de execução do algoritmo aumentou, isso se justifica pelo fato do algoritmo Ant Based Clustering precisar de muito mais iterações que o algoritmo SACA. Já o índice de validação *C-Index* não foi melhorado no Ant Based Clustering, porém o objetivo desse algoritmo é diminuir a quantidade de parâmetros de entrada.

### C. Algoritmo A<sup>2</sup>CA

Na avaliação dos resultados gerados pelo algoritmo A<sup>2</sup>CA, foram realizados vários testes a fim de se determinar os ideais para esse algoritmo, considerando essa base de dados. Os parâmetros utilizados foram os seguintes:

- a) número de formigas: 30;
- b) total de iterações: 8000;
- c) memória: 20;
- d) alfa: 0.3;
- e)  $k_p$ : 0.9;
- f)  $k_d$ : 0.3;
- g) atributo de entrada: pH e fe.

Na tabela V pode-se observar um dos resultados obtidos nos testes pelo algoritmo A<sup>2</sup>CA.

TABELA V. RESULTADO DE UM DOS TESTES REALIZADOS COM O ALGORITMO A<sup>2</sup>CA

Teste	CF <sup>a</sup>	PNC <sup>b</sup>	Tempo	Índice de Dunn	Índice C-Index
1	5	20	03m: 12s: 435ms	1,3498	C1 = 0,0644 C2 = 0,2131 C3 = 0,0841 C4 = 0,0245 C5 = 0,0795

<sup>a</sup> Clusters.  
<sup>b</sup> Pontos Não Classificados.

Esse algoritmo, apesar de necessitar dos mesmos parâmetros que o algoritmo SACA, desconsiderando o Campo de Visão, apresenta resultados bons em relação ao índice C-Index, pois os valores apresentados são mais próximos de zero. Isso graças à heurística de feromônio e a sua adaptação do Campo de Visão para cada formiga. Esse algoritmo, como o SACA, tende a formar muitos grupos no início, porém graças as suas adaptações, pequenos grupos são desfeitos e seus elementos passam a pertencer a grupos maiores e que contenham objetos similares a ele. Quanto maior o número de iterações do algoritmo, mais compactos são os grupos formados por ele.

Na tabela VI pode-se observar o desvio padrão, média e coeficiente de variação dos resultados obtidos pelo algoritmo A<sup>2</sup>CA.

TABELA VI. VARIAÇÃO DOS RESULTADOS DO ALGORITMO A<sup>2</sup>CA

Medidas	Clusters	Pontos não classificados	Índice de Dunn
Desvio padrão	0,7745	3,618	0,0008
Média	5	22,9	1,3501
Coefficiente de variação	15,49	15,7991	0,0592

A tabela VI demonstra que a quantidade de *clusters* gerados pelo A<sup>2</sup>CA é menor comparado com os dois algoritmos anteriores, os demais resultados também tiveram uma variação menor comprovando dessa forma o objetivo do algoritmo que é diminuir a variação dos resultados do algoritmo SACA. O A<sup>2</sup>CA apresentou os melhores resultados do índice de validação C-Index. Levando-se o desempenho em consideração pode-se concluir que o A<sup>2</sup>CA obteve o melhor desempenho.

## X. CONSIDERAÇÕES FINAIS

Este artigo apresentou os algoritmos SACA, Ant Based Clustering e A<sup>2</sup>CA que utiliza o conceito de agrupamento baseado em inteligência de enxame por colônia de formigas para a tarefa de agrupamento da Shell Orion Data Mining Engine.

O algoritmo A<sup>2</sup>CA se diferenciou do SACA no fato de não ser tão sensível com os parâmetros informados devido a sua heurística de feromônio, que possibilitou que os pequenos grupos formados durante o processo do algoritmo fossem desmanchados e seus elementos colocados onde tem maior concentração de feromônio. O algoritmo Ant Based Clustering, apresentou uma demora considerável na formação dos *clusters*, devido ao fato de necessitar de mais iterações do que os algoritmos SACA e A<sup>2</sup>CA, os resultados dos índices de validação não foram tão bons comparados aos demais. Pode-se concluir que o algoritmo A<sup>2</sup>CA foi o que apresentou o melhor resultado na qualidade interna dos *clusters* formados, considerando-se os dados referentes a indicadores de monitoramento de bacias hidrográficas da região carbonífera do Sul de Santa Catarina.

## REFERÊNCIAS

- [1] R. Goldschmidt, E. L. Passos, Data mining: uma guia prático. Rio de Janeiro: Elsevier, 2005.
- [2] M. Dorigo, T. Stützle, Ant colony optimization. Cambridge: MIT Press, 2004.
- [3] E. D. Lumer, B. Faieta, Diversity and adaptation in populations of clustering ants. In Proceedings of the Third International Conference on Simulation of Adaptive Behaviour, pages 501–508, 1994.
- [4] J. Handl, Ant-based methods for tasks of clustering and topographic mapping: extensions, analysis and comparison with alternative

techniques. Masters Thesis, Universität, Erlangen-Nürnberg, Erlangen, Germany, 2003.

- [5] A. L. Vizine, L. N. Castro, E. R. Hruschka, R. R. Gudwin, Towards Improving Clustering Ant: An Adaptative Ant Clustering Algorithm, 29, pages 143-154, 2005.
- [6] J. A. Berry, G. S. Linoff, Mastering Data Mining. New York: John Wiley & Sons; 2000.
- [7] E. Bonabeu, G. Theraulaz, M. Dorigo, Swarm Intelligence: from natural to artificial systems. New York, USA: Oxford University Press, 1999.
- [8] A. L. Lauro, Agrupamento de dados utilizando algoritmo de colônia de formigas. 2008. Dissertação de Mestrado – Pós-graduação e Engenharia Civil, Universidade Federal do Rio de Janeiro, Rio de Janeiro, Rio de Janeiro, 2008.
- [9] A. K. Jain, R. C. Dubes, Algorithms for clustering data. Englewood Cliffs: Prentice Hall, 1988.
- [10] G. W. Milligan, M. C. Cooper, An examination of Procedures for determining the Number of clusters in data set. Psychometrika, Columbus. p. 159-179. Jun. 1985.