

# Explorer Pattern Tool Adaptive: Uma Ferramenta para Mineração de Dados Utilizando Algoritmos Genéticos Adaptativos

Lorena Mascarenhas Lima<sup>1</sup>, Daniela Barreiro Claro<sup>1</sup>

<sup>1</sup>Laboratorio de Sistemas Distribuidos (LaSiD)  
Departamento de Ciência da Computação  
Universidade Federal da Bahia (UFBA)  
Av. Adhemar de Barros, s/n, Ondina – Salvador – BA – Brasil

lore.mascarenhas@gmail.com, dclaro@ufba.br

**Abstract.** *This work proposes the implementation of an adaptive genetic algorithm (GA), aiming to better determine the configuration parameters and as a consequence to improve the accuracy obtained within traditional genetic algorithms. Our proposed algorithm has been developed into a tool called AEPT (Adaptative Explorer Pattern Tool). Experiments have been performed and presented an improvement on the average accuracy face those traditional genetic algorithms.*

**Resumo.** *Este artigo descreve a implementação de um algoritmo genético (AG) adaptativo com o intuito de melhor definir os parâmetros de configuração dos AG e consequentemente melhorar o valor da acurácia obtida através da utilização somente dos algoritmos genéticos tradicionais. O algoritmo proposto foi implementado em uma ferramenta chamada AEPT (Adaptative Explorer Pattern Tool). Experimentos foram realizados e demonstraram que o algoritmo implementado alcançou uma melhora na acurácia média frente aos AG tradicionais.*

## 1. Introdução

O benefício da descoberta de padrões nas bases de dados está cada vez mais sendo utilizado em grandes corporações. A grande quantidade de informação atualmente disponível nestas bases dificulta a extração de conhecimento relevante para uma organização. Esse conhecimento, quando descoberto, pode ser usado na melhoria do processo de decisão [Freitas 2002]. Nesse contexto, o KDD (*Knowledge Data Discovery*) permite descobrir esses padrões através de métodos automáticos baseados na estatística e no aprendizado de máquina com o intuito de extrair conhecimento de alto nível.

Uma das tarefas do processo de KDD é a Mineração de Dados, na qual é aplicado um determinado algoritmo. Estes algoritmos podem ser executados através de diversas técnicas, onde destacam-se os algoritmos genéticos. A descoberta de padrões através dos algoritmos genéticos e a comparação com os extraídos através da lógica *fuzzy* foi desenvolvida através da ferramenta EPT (Explorer Pattern Tools) [Santos and Claro 2009]. Os resultados obtidos por esta ferramenta não foram satisfatórios, pois a acurácia não foi superior à obtida em técnica de árvores clássicas e nem utilizando as árvores *fuzzy*

[Souza and Claro 2008]. Constatou-se que esse problema poderia ser fruto das decisões relativas ao algoritmo genético utilizado, uma vez que os algoritmos genéticos são técnicas heurísticas de otimização global [Linden 2006] nas quais a busca por soluções depende da representação adotada para o problema e dos parâmetros de configuração. Mais especificamente, as taxas de mutação (*mutation*) e cruzamento (*crossover*) influenciam na convergência do algoritmo [Cox 2005].

As definições das taxas de mutação e *crossover*, assim como, de outros parâmetros de configuração existentes, não é uma tarefa trivial. Apesar de diversos estudos realizados nessa área, a determinação de um valor ótimo genérico que possa ser utilizado na resolução de qualquer problema ainda não foi fixado. Sendo assim, o estabelecimento desses parâmetros é feito, na maioria das vezes, de forma empírica.

Em termos de determinação de parâmetros, além de estudos empíricos que sugiram alguns valores ótimos, existe a abordagem adaptativa para a configuração do algoritmo genético. Os algoritmos genéticos adaptativos propõem uma configuração dinâmica dos valores dos parâmetros, possibilitando a alteração dos valores em tempo de execução. Sendo assim, os valores podem variar até que um valor adequado seja encontrado. Outro fator importante contemplado pelos algoritmos genéticos adaptativos é a variação da importância dos operadores de acordo com o estágio da busca. Diante disso, o presente trabalho propõe o desenvolvimento de um algoritmo genético adaptativo com intuito de identificar indivíduos com características semelhantes na população e ajustar a taxa do operador de mutação e o tipo do operador de *crossover* de acordo com a qualidade desses indivíduos.

O algoritmo genético adaptativo proposto utiliza o cálculo da probabilidade de mutação [Srinivas et al. 1994] para cada indivíduo. A variação do *crossover* dar-se-á baseada na técnica de *crossover* utilizada em [Srinvasa et al. 2007]. Neste caso, indivíduos com alto valor de *fitness*<sup>1</sup> sofrem *crossover* de um ponto, enquanto os com baixo *fitness* sofrem *crossover* uniforme.

O algoritmo genético adaptativo foi implementado na ferramenta EPTA (Explorer Pattern Tree Adaptive) onde experimentos foram realizados com o intuito de validar o algoritmo proposto. Os resultados obtidos contêm valores médios da acurácia superior aos valores obtidos pelos experimentos realizados utilizando algoritmos genéticos tradicionais.

O restante deste artigo está dividido da seguinte maneira: a seção 2 descreve as abordagens de aplicação dos AG na Mineração de Dados. A seção 3 apresenta o algoritmo genético adaptativo desenvolvido e a ferramenta AEPT; a seção 4 descreve os experimentos realizados para a validação do algoritmo desenvolvido e discute os resultados obtidos; finalizando, a seção 5 apresenta as conclusões obtidas e os trabalhos futuros.

## **2. AGs aplicados à descoberta de regras de classificação**

A utilização dos algoritmos genéticos para a descoberta de regras pressupõe que os indivíduos da população representem as regras. Assim, cada indivíduo deve ser codificado de maneira a representar as regras de predição (*IF-THEN*). Existem duas abordagens

---

<sup>1</sup>O *fitness* é a medida que determina a qualidade de um indivíduo como solução do problema em questão [Linden 2006].

clássicas para a codificação dos indivíduos: a Michigan e a Pittsburgh [Freitas 2002]. Em ambas abordagens, a parte antecedente da regra é composta de uma conjunção de condições, diferindo somente a quantidade de regras que um indivíduo representa. Na abordagem Pittsburgh, cada indivíduo representa um conjunto de regras de predição. Por sua vez, na abordagem Michigan cada indivíduo representa uma única regra de predição.

Neste trabalho, a abordagem utilizada foi a Michigan devido à simplificação dos indivíduos e a não interação entre as regras. Nesta abordagem os indivíduos são menores, implicando na simplificação dos operadores genéticos.

A representação de uma regra do tipo *IF* ( $c_1$ ) *and ... and* ( $c_n$ ) *THEN* (class =  $c_i$ ) utilizando a abordagem Michigan e a codificação binária é composta de  $n$  condições (antes do THEN), onde cada condição verifica o valor de um determinado atributo. Um determinado atributo que pode assumir  $x$  valores (discretos) necessitará de  $x$  bits para ser codificado. Por exemplo, considere uma regra contendo o atributo *outlook*, cujos valores possíveis são *sunny*, *overcast*, e *rainy*. Este exemplo requer três bits para codificar esta regra. Assim, a parte antecedente da regra *IF* (outlook = sunny) *THEN* (play = yes) é representada pelo cromossomo 100.

### 3. Algoritmo Adaptativo Desenvolvido

Com o intuito de obter uma maior acurácia nas regras obtidas através da extração de conhecimento, foi desenvolvido um algoritmo genético adaptativo 1.

---

#### Algorithm 1 Algoritmo Genético Adaptativo desenvolvido

---

```
1: procedure executaGAClassificador(arquivo, areaSaida)
2:   int qtdadeIndividuos;
3:   for  $i = 0$  to numGeracoes do
4:     this.calculaMediaFitness(this.populacao);
5:     novaPopulacaoFinal.add(populacao.indiceMaior);
6:     this.sufragioUniversal(arquivo);
7:     this.individuosMelhoresFitness();
8:     this.individuosPioresFitness();
9:     this.tentaAplicarMutacao(novaPopulacaoParcial, arquivo);
10:    this.populacao.removeAll(this.populacao);
11:    this.populacao.addAll(novaPopulacaoFinal);
12:  end for
13: end procedure
```

---

Inicialmente calcula-se a média do *fitness* da população (linha 4) e o indivíduo de maior *fitness* da população é buscado a fim de evitar a necessidade de iterar novamente a população inteira, evitando custos adicionais de processamento. Em seguida, a estratégia elitista (linha 5) <sup>2</sup> é aplicada copiando esse indivíduo de maior *fitness* da população, ou seja, o melhor indivíduo para a geração seguinte.

Posteriormente, o método de seleção *sufrágio universal* (linha 6) é aplicado à população corrente com o objetivo de retornar os indivíduos que através da aplicação dos

---

<sup>2</sup>Elitismo é um método adicional de seleção que consiste em manter os melhores indivíduos da geração corrente na próxima geração[Mitchell 1996].

operadores de *crossover* e mutação irão gerar a nova população. Os indivíduos mais votados terão mais chances de serem selecionados, já que a parcela da roleta de cada indivíduo será proporcional a quantidade de votos que o indivíduo recebeu. O sufrágio universal foi adaptado e os indivíduos selecionados passam a ser divididos em dois grupos: um grupo com *fitness* acima da média do *fitness* da população (linha 7), e outro pelos indivíduos com valores abaixo da média (linha 8).

Após a aplicação do sufrágio universal, haverá duas listas de indivíduos selecionados, representadas pelas variáveis *indMelhoresFitness* e *indPioresFitness*. Caso a lista dos melhores possua indivíduos (linha 7), então será aplicado o *crossover*, gerando o número de indivíduos determinado. Por sua vez, o mesmo procedimento é realizado com a lista dos piores indivíduos (linha 8). Os detalhes adaptativos são descritos na seção 3.1.

Feito isso, os indivíduos gerados poderão sofrer mutação de acordo com a probabilidade de ocorrência da mesma. O cálculo dessa probabilidade é feito para cada indivíduo da população de maneira adaptativa. Os métodos *tentaAplicarMutacao* (linha 9) e *calculaProbabilidadeMutacao* são os responsáveis por este processamento e são descritos na seção 3.2.

Por fim, a população corrente é substituída pela nova população gerada (linha 10 e 11), as listas são esvaziadas e todo o processo descrito é repetido até que o número de gerações estabelecido seja atingido.

### **3.1. Crossover adaptativo**

A facilidade da convergência para uma solução ótima deve ter grande probabilidade de preservar os melhores indivíduos enquanto que os piores devem sofrer alterações mais significativas. Neste caso, aos melhores indivíduos é aplicado o *crossover* de um ponto, enquanto que aos piores indivíduos é aplicado o *crossover* uniforme. O método sorteia um ponto de corte dentre os possíveis e divide os cromossomos selecionados nesse ponto de corte. Um ponto de corte é definido aleatoriamente para determinar qual parte de cada cromossomo pai formará o cromossomo filho.

De acordo com [Srinvasa et al. 2007], regiões promissoras devem ter o número de indivíduos aumentados e a probabilidade de mutação, assim como o número de pontos de *crossover*, diminuídos. De maneira semelhante, as regiões com baixo *fitness* devem ter o número de indivíduos reduzidos, enquanto que a taxa de mutação e o número de pontos de *crossover* aumentados.

Uma comparação com a abordagem implementada pode ser feita, destacando algumas semelhanças e diferenças. O tamanho da população varia na abordagem proposta por [Srinvasa et al. 2007], enquanto que neste trabalho ele permanece constante durante a execução. Para favorecer a exploração das regiões não promissoras e promissoras, respectivamente, Srinvasa et. al adota a diminuição, no primeiro caso, e o aumento, no segundo, do número de pontos de *crossover*. Semelhantemente, como descrito anteriormente, o *crossover* adaptativo adotado é dividido em dois tipos de *crossover*: o de um ponto e o uniforme.

### **3.2. Mutação Adaptativa**

A probabilidade de mutação é calculada adaptativamente a cada geração para um indivíduo específico. Essa probabilidade é determinada pela média do *fitness* da população

$(\bar{f})$ , *fitness* máximo da população ( $f_{max}$ ) e ainda pelo *fitness* do indivíduo para o qual a probabilidade será calculada ( $f'$ ). Para valores de *fitness* menores que a média, o valor da probabilidade de mutação será 0,5. Já para os valores acima da média, a probabilidade é calculada pelas fórmulas 1 e 2. No caso do *fitness* máximo ser igual a média do *fitness*, haveria uma divisão por zero no cálculo. Logo, quando esse caso ocorre, a probabilidade de mutação é considerada 0.

$$pm = \frac{k_2(f_{max} - f)}{f_{max} - \bar{f}}, f \geq \bar{f}, \quad (1)$$

$$pm = k_4, f < \bar{f}, \quad (2)$$

Uma vez que o cálculo da probabilidade de mutação é feito com base nos valores de *fitness* médio e máximo da população, a mutação só deve ser então aplicada após o *crossover*.

### 3.3. Ferramenta desenvolvida

A ferramenta AEPT (Adaptative Explorer Pattern Tool) foi desenvolvida com intuito de melhor disseminar a utilização do algoritmo adaptativo proposto. Esta ferramenta é uma extensão da ferramenta EPT [Santos and Claro 2009] contemplando as características adaptativas.

## 4. Experimentos

A fim de avaliar e conseqüentemente validar o trabalho desenvolvido, foram realizados testes com bases de dados reais. Os testes retornaram a acurácia média que foi utilizada como parâmetro de avaliação. As taxas de acurácia obtidas através da execução do algoritmo genético adaptativo implementado foram então comparadas com as obtidas pelo algoritmo genético clássico implementado na EPT [Santos and Claro 2009].

### 4.1. Critério de avaliação

Como definido anteriormente, o critério de avaliação utilizado nos experimentos realizados foi a acurácia média. Uma das características desejáveis na descoberta de conhecimento, em especial na classificação, é a descoberta de regras que possuam um bom poder de predição. Esse poder de predição, também conhecido como acurácia preditiva (AC), pode e deve ser medido.

Finalmente, a acurácia preditiva de uma regra pode ser medida através da fórmula 3 [Freitas 2002]. Nesta fórmula,  $|A\&C|$  representa o número de exemplos que satisfazem os atributos de  $A$  e são classificados como  $C$ .  $|A|$ , por sua vez, representa todos os exemplos que satisfazem todas as condições da parte antecedente da regra [Freitas 2002]. Assim, se existem dez exemplos cobertos pela regra  $|A|$ , e somente sete deles possuem o valor da classe igual a  $C$ , o valor da acurácia preditiva dessa regra, calculada pela equação 3, será:  $AC = \frac{|A\&C|}{|A|} = \frac{7}{10} = 70\%$ .

$$AC = \frac{|A\&C|}{|A|} \quad (3)$$

Cada regra de classificação possui uma matriz de confusão [Freitas 2002] que permite uma outra maneira de medir a acurácia preditiva. A matriz de confusão deve ser preenchida com os seguintes valores:

- Verdadeiros Positivos (VP): representa a quantidade de exemplos que satisfazem tanto a parte antecedente, quanto a parte consequente da regra, ou seja exemplos que satisfazem as condições de A e são classificados como C;
- Falsos Positivos (FP): representa a quantidade de exemplos que satisfazem as condições de A (parte antecedente da regra), mas possuem classe diferente C (parte consequente da regra);
- Falsos Negativos (FN): representa a quantidade de exemplos que não satisfazem as condições de A (parte antecedente da regra), mas são classificados com a classe C (parte consequente da regra);
- Verdadeiros Negativos (VN): representa a quantidade de exemplos que não satisfazem nenhuma parte da regra, nem atendem às condições de A (parte antecedente da regra) nem são classificados como C (parte consequente da regra).

Com os valores devidamente preenchidos o cálculo da acurácia preditiva é dado pela equação [Freitas 2002]:

$$AC = \frac{|VP|}{|VP + FP|} \quad (4)$$

Analisando a fórmula 4 é notável que quanto maior for o número de exemplos classificados como verdadeiros positivos e quanto menor for o número de exemplos classificados como falsos positivos maior será o valor de AC, logo, melhor a regra.

#### 4.2. Bases de dados utilizadas

Os experimentos foram realizados utilizando as mesmas bases que foram utilizadas em [Souza and Claro 2008] e em [Santos and Claro 2009]. Adicionalmente, testes foram realizados com a base de dados Wisconsin Breast Cancer (Wisc) também utilizada em [Srinvasa et al. 2007]. As bases: Iris, Spambase, e Wisconsin são de origem do repositório UCI (UC Irvine Machine Learning Repository) [Asuncion and Newman 2007]. As bases restantes, Segment-challenge e Segment-test, são bases do repositório do WEKA (Waikato Environment for Knowledge Analysis) [Frank and Trigg 1993]. Abaixo uma breve descrição das bases [Santos and Claro 2009]:

- Segment-Challenge - essa base de 1500 objetos e 20 atributos é formada por objetos selecionados aleatoriamente de uma base de dados composta por imagens de outdoors segmentadas. O atributo *class* pode ter os seguintes valores: *brickface*, *sky*, *foliage*, *cement*, *window*, *path*, e *grass* ;
- Segment-Test - base de dados formada por 810 registros obtidos da base Segment-Challenge;
- SPAMBASE [Hopkins et al. 2001] - base constituída de 4601 emails que podem ser classificados como spam ou não (1 ou 0). Além do atributo *class*, existem mais 57 atributos contínuos;
- Iris [R.A. Fisher 1988] - essa base é composta por 50 registros de cada uma das três classes de flores: Iris-setosa, Iris-versicolor e iris-virginica. Os exemplos são classificados de acordo com os quatro atributos presentes na base: comprimento da sépala, largura da sépala, comprimento da pétala e largura da pétala;

- Wisconsin Breast Cancer Database (Wisc) [Wolberg 1992] - essa base foi obtida da University of Wisconsin Hospitals, Madison. Seus 699 exemplos são casos clínicos. A base possui 10 atributos e um atributo *class* que pode conter os valores 2 para benigno ou 4 para maligno.

A tabela 1 resume as principais características dessas bases.

**Tabela 1. Resumo das características das bases de dados**

Conjunto de Dados	Classes	Atributos de Entrada	Tipos dos Atributos	Qtde. De instancias
Segment-challenge	7	19	inteiro e real	1500
Segment-test	7	19	inteiro e real	810
SPAMBASE	2	57	inteiro e real	4601
Iris	3	4	Real	150
Wisconsin	2	9	inteiro	699

### 4.3. Métodos Experimentais Utilizados

Dentre os métodos de experimentação disponíveis na ferramenta EPT [Santos and Claro 2009] os seguintes foram utilizados na realização dos experimentos:

- Percentagem de Treino: método no qual o usuário define a porcentagem dos exemplos da base de dados que serão utilizados como conjunto de treinamento. Os exemplos que farão parte do conjunto de treinamento são selecionados de maneira aleatória, por sua vez, os exemplos que farão parte do conjunto de teste serão aqueles que não foram selecionados para o conjunto de treinamento [Souza and Claro 2008];
- Validação Cruzada: método no qual a base de dados é dividida em  $k$  folds e cada um é selecionado por vez para ser o conjunto de teste. O teste é repetido até que cada um dos  $k$  folds seja utilizado como conjunto de teste. Os outros  $k-1$  são utilizados como conjunto de treinamento. Ao final, a acurácia é dada pela média das acurácias obtidas para cada um dos  $k$  folds [Souza and Claro 2008].

### 4.4. Procedimentos Experimentais

Os experimentos objetivaram comparar as acurácias médias obtidas nas execuções do algoritmo genético adaptativo implementado com as obtidas nas execuções do algoritmo genético clássico da ferramenta EPT.

Os parâmetros utilizados na configuração da classe de testes foram os seguintes:

- Número de execuções: 300;
- Porcentagem do conjunto de dados: 60% (para os testes que utilizaram divisão por porcentagem);
- Porcentagem do conjunto de testes: 40% (para os testes que utilizaram divisão por porcentagem);
- Número de folds: 10 (para os testes que utilizaram validação cruzada);
- Rótulo Classificador: *span* nos testes com a base SPAMBASE e *class* nos testes com as demais bases;
- Quantidade de intervalos para os dados contínuos: 5.

Os primeiros testes realizados variaram o tamanho da população e o número de gerações do algoritmo. No caso do tamanho da população foram utilizados os valores 30, 60 e 100. Já para o número de gerações os valores utilizados foram: 15, 30, 80, 100. Dois tipos de testes foram utilizados: a validação cruzada (VC) e a divisão por porcentagem (DP).

#### 4.5. Resultados dos experimentos utilizando a nova base de dados

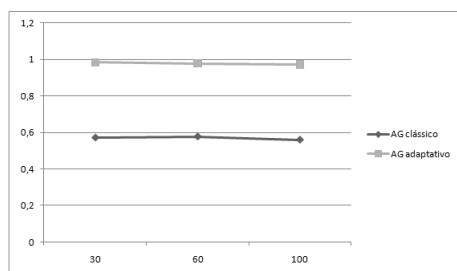
Os resultados obtidos com a nova base de dados (Wisconsin) foram realizados tanto para o AG adaptativo quanto para o AG clássico através da ferramenta AEPT. Os experimentos foram realizados para populações de tamanho 30, 60 e 100 e para gerações de tamanho 15 e 30.

Os resultados obtidos na execução dos dois algoritmos (adaptativo e clássico) com os parâmetros definidos na subseção 4.4 estão dispostos na tabela 2.

**Tabela 2. Resultados dos experimentos utilizando a base Wisconsin**

Algoritmo	Teste	VC		DP		VC		DP	
		30	30	60	60	100	100		
Adaptativo	15	0,992	0,964	0,990	0,962	0,986	0,954		
Adaptativo	30	0,998	0,984	0,997	0,979	0,994	0,973		
Clássico	15	0,566	0,565	0,569	0,555	0,560	0,552		
Clássico	30	0,561	0,572	0,566	0,578	0,566	0,560		

Analisando a tabela 2, é notória a superioridade dos resultados obtidos através da aplicação do AG adaptativo em relação aos obtidos na aplicação do AG clássico. Essa constatação pode ser melhor visualizada no gráfico da figura 1 onde foi utilizado o método Divisão por Porcentagem para os dois algoritmos, com 30 gerações e variando o tamanho da população entre 30, 60 e 100.



**Figura 1. Gráfico de comparação entre o AG clássico e o adaptativo na divisão por porcentagem com geração 30 e base Wisconsin.**

#### 4.6. Resultados dos experimentos após ajustes no código do algoritmo genético clássico

No algoritmo genético clássico, implementado na ferramenta EPT [Santos and Claro 2009], toda a geração era gerada dos mesmos pais. Ajustes foram feitos com o intuito de adequar o funcionamento deste algoritmo garantindo que para a formação da nova geração sejam utilizados diversos pais (iterando-os) e não sempre os mesmos dois.

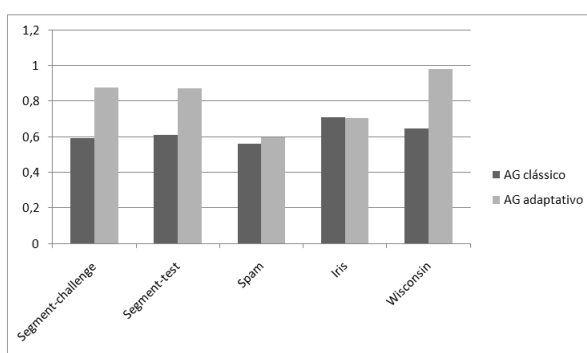


Esses experimentos foram realizados para todas as bases de dados utilizadas. Os parâmetros utilizados foram: 60 para o tamanho da população e 30 para o número de gerações e o tipo do teste foi divisão por porcentagem. A tabela 3 resume os resultados obtidos.

**Tabela 3. Resultados dos experimentos de divisão por porcentagem com população 60 e geração 30.**

Algoritmo/Base	Segment-challenge	Segment-test	Spam	Iris	Wisconsin
AG clássico	0,590	0,612	0,558	0,708	0,648
AG adaptativo	0,877	0,873	0,597	0,703	0,979

Um gráfico comparativo destes resultados obtidos é apresentado na figura 2.



**Figura 2. Gráfico comparativo entre as acurácias médias obtidas.**

Analisando a tabela 3 e a figura 2 obtidos após a realização dos últimos testes algumas considerações podem ser feitas:

- Nas bases de dados Segment-challenge, Segment-test, Spam e Wisconsin houve um aumento na acurácia média obtida utilizando os AG adaptativos;
- Na base de dados Iris houve uma diminuição na acurácia média obtida através do uso dos AG adaptativos. Isso ocorre devido à pequena quantidade de atributos de entrada que esta base contém, não garantindo a precisão nos resultados alcançados.

Diante do apresentado, o algoritmo genético adaptativo desenvolvido obteve resultados superiores na maioria das bases de dados utilizadas. Assim sendo, o AG adaptativo foi superior ao AG clássico utilizada na Mineração de Dados.

## 5. Conclusão

O processo de escolha dos valores dos parâmetros de configuração de um algoritmo genético visa encontrar valores adequados. Esse processo não é trivial, já que nenhum estudo definiu valores padrões aceitáveis para qualquer tipo de problema. Portanto, estes valores devem ser automatizados de forma que sejam ajustados durante a execução.

Este trabalho propôs a implementação de um algoritmo genético adaptativo, a fim de eliminar a necessidade de definição de parâmetros iniciais e ainda possibilitar a variação destes durante a execução do algoritmo. De posse do algoritmo adaptativo implementado, diversos experimentos foram executados, objetivando uma posterior

comparação entre os resultados destes e os obtidos com o algoritmo genético clássico. Este algoritmo foi implementado na ferramenta AEPT para uma melhor utilização e difusão. Quanto aos resultados dos experimentos, no geral obteve-se resultados superiores quando comparados aos obtidos utilizando a abordagem clássica para o algoritmo genético.

Como trabalho futuro, pretende-se substituir o método de discretização utilizado pelo algoritmo e analisar outros tipos de algoritmos, como os algoritmos meméticos no processo de mineração de dados.

## Referências

- Asuncion, A. and Newman, D. (2007). UCI machine learning repository. Último acesso 24 de outubro de 2009.
- Cox, E. (2005). *Fuzzy Modeling and Genetic Algorithms for Data Mining and Exploration*. Elsevier/Morgan Kaufmann.
- Frank, E. and Trigg, L. (1993). Weka. Último acesso em 14 de novembro de 2007.
- Freitas, A. A. (2002). A survey of evolutionary algorithms for data mining and knowledge discovery. In *Advances in Evolutionary Computation*, pages 819–845. Springer-Verlag.
- Hopkins, M., Reeber, E., Forman, G., and Suermondt, J. (2001). Spambase. Último acesso em 27 de outubro de 2007.
- Linden, R. (2006). *Algoritmos Genéticos. Uma importante ferramenta da Inteligência Computacional*. Brasport, Rio de Janeiro, RJ, BR.
- Mitchell, M. (1996). *An introduction to genetic algorithms*. MIT Press, Cambridge, MA, USA.
- R.A. Fisher, M. M. (1988). Iris. Último acesso em 14 de novembro de 2007.
- Santos, J. and Claro, D. B. (2009). Explorer pattern tools: Uma ferramenta para mineração de dados utilizando algoritmos genéticos. *ERBASE 2009 - WTICG-BASE (Workshop de Trabalhos de Iniciação Científica e Graduação Bahia, Alagoas e Sergipe)*.
- Souza, E. N. P. and Claro, D. B. (2008). Explorer fuzzy tree: uma ferramenta para experimentação de técnicas de classificação baseadas em árvores de decisão fuzzy. *ERBASE 2008 - WTICG-BASE (Workshop de Trabalhos de Iniciação Científica e Graduação Bahia, Alagoas e Sergipe)*.
- Srinivas, M., Patnaik, L. M., and Fellow (1994). Adaptive probabilities of crossover and mutation in genetic algorithms. *IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETIC*, 24 n4.
- Srinvasa, K. G., Venugopal, K. R., and Patnaik, L. M. (2007). A self-adaptive migration model genetic algorithm for data mining applications. *Information Sciences: an International Journal*, 177(20):4295–4313.
- Wolberg, D. W. H. (1992). Base de dados wisconsin breast cancer. Último acesso em 24 de outubro de 2009.