# ABORDAGEM DE OTIMIZAÇÃO MULTIOBJETIVO APLICADA À SELEÇÃO DE ATRIBUTOS USANDO ALGORITMO DE NUVEM DE PARTÍCULAS

**Viviane Dal Molin de Souza[2], Helon Vicente Hultmann Ayala[1], Deborah Ribeiro Carvalho[3] e Leandro dos Santos Coelho[2]**

[1]Graduação em Engenharia Mecatrônica (Controle e Automação), Pontifícia Universidade Católica do Paraná, Rua Imaculada Conceição 1155, 80215-901, Curitiba, PR, Brasil

[2]Programa de Pós-Graduação em Engenharia de produção e Sistemas (PPGEPS), Pontifícia Universidade Católica do Paraná, Rua Imaculada Conceição 1155, 80215-901, Curitiba, PR, Brasil

[3]Universidade Tuiuti do Paraná, Curso de Ciência da Computação, Rua Sydney A. Rangel Santos, 238, 82010-330, Curitiba, PR, Brasil

viviane.molin@gmail.com; helon.ayala@pucpr.br; deborah@ipardes.pr.gov.br; leandro.coelho@pucpr.br

**Resumo –** O algoritmo de otimização através de nuvem de partículas (PSO) é uma técnica metaheurística desenvolvida recentemente e pertence à categoria de técnicas de inteligência de enxames. Os conceitos da inteligência de enxames são inspirados no comportamento social dos animais, tais como enxames de aves e cardumes de peixes. Devido a sua habilidade natural de convergir rapidamente, o algoritmo PSO é também utilizado para resolver problemas de otimização multiobjetivo. Recentemente, diversas investigações têm sido realizadas para aplicar abordagens do algoritmo PSO em descoberta de conhecimento em bases de dados (*KDD – Knowledge Discovery in Databases*). O processo de KDD é composto pelas seguintes etapas: seleção da base de dados, seleção de atributos, pré-processamento dos dados, mineração de dados e pós-processamento. O objetivo deste artigo é a seleção de atributos usando uma abordagem de PSO baseada em otimização multiobjetivo e variáveis inteiras para a seleção e avaliação dos atributos selecionados. A proposta do método de seleção de atributos utilizando uma abordagem de PSO multiobjetivo foi avaliada em dez bases de dados obtidas no repositório de dados da UCI (*Machine Learning Repository – University of California Irvine*). Neste contexto, o problema multiobjetivo resolvido pelo PSO considerou dois objetivos diferentes: i) minimizar a taxa de erro e ii) minimizar o tamanho das árvores obtidas pelo algoritmo C4.5. Além disso, o algoritmo foi definido como critério para a comparação das soluções encontradas pelo PSO com abordagem multiobjetivo.

**Palavras-chave –** otimização multiobjetivo, seleção de atributos, mineração de dados, enxame de partículas.

**Abstract** – The particle swarm optimization (PSO) algorithm is a recently developed metaheuristic technique and belongs to the category of swarm intelligence techniques. The swarm intelligence concepts are inspired by the social behavior of flocking animals such as swarms of birds and fish school. Due to its natural ability to converge quick, PSO algorithm is also used to solve multi-objective optimization problems. Recently, several investigations have been undertaken to apply of PSO algorithm approaches in Knowledge Discovery in Database (KDD) procedures. The KDD procedure can present the following steps: selection of database, attributes selection, data pre-processing, data mining, and data pos-processing. The objective of this paper is the attributes selection using a PSO approach based on multi-objective optimization and integer variables for selection and evaluation of selected attributes. The proposed attributes selection method based on a multi-objective PSO approach was evaluated to ten databases obtained of UCI (Machine Learning Repository - University of California - Irvine) repository. In this context, the multi-objective problem solved by PSO considered two different objectives: i) minimization of error rate and ii) minimization of trees size obtained by C4.5 algorithm. In addition, the C4.5 algorithm was defined as comparison criterion for the solutions found by multi-objective PSO approach.

**Keywords—** multiobjective optimization, attributes selection, data mining, particle swarm optimization.

## 1. Introdução

As empresas frequentemente estão buscando formas de auxiliar o gestor a adquirir um maior conhecimento sobre determinado problema, antes que o gestor tome uma decisão. Com o aumento da

capacidade de armazenamento de informações em bases de dados, a quantidade de dados com potencial de auxiliar o gestor no processo de tomada de decisão é imensa, porém nem sempre estes dados são facilmente vislumbrados, pois o seu tamanho e quantidade ultrapassam a capacidade humana de avaliação. Assim pode-se utilizar técnicas para descoberta de conhecimento em bases de dados e fazer uso deste conhecimento para auxiliar na tomada de decisão. A área da computação destinada a esta pesquisa denomina-se Descoberta de Conhecimento em Bases de Dados (*Knowledge Database Discovery*, KDD).

Segundo [3], o KDD é composto por diversas etapas, que são as seguintes: i) *seleção da base de dados e seleção dos atributos*: etapa em que se efetua a seleção e coleta dos dados necessários; ii) *limpeza ou pré-processamento*: etapa em que os dados coletados são analisados, verifica-se a existência de ruídos e se houver estes ruídos, os dados devem ser tratados para que estes ruídos possam ser retirados; iii) *transformação e enriquecimento dos dados*: nesta etapa os dados já estão tratados. Então é realizada uma verificação sobre a necessidade de dados adicionais que possam enriquecer a base; iv) mineração de dados *(data mining*): é a principal etapa do processo de KDD. É a etapa na qual ocorre a mineração de dados através de algoritmos com o objetivo de extrair um padrão em um determinado conjunto de dados; v) *interpretação e avaliação ou pós-processamento*: no pós-processamento ocorre um refinamento do conhecimento descoberto e também a validade do conhecimento gerado é analisada, verificando se o conhecimento é relevante para o domínio em questão, já descartando o conhecimento irrelevante; vi) *consolidação do conhecimento descoberto ou análise*: nesta etapa os resultados dos algoritmos de mineração de dados devem ser analisados, interpretados e avaliados.

A seleção de atributos tem como objetivo descobrir um subconjunto de atributos relevantes para uma tarefa alvo, considerando os atributos originais, e é importante, entre outras coisas, por gerar um processo de aprendizagem eficiente. Os atributos redundantes prejudicam o desempenho do algoritmo de aprendizagem tanto na velocidade, devido à dimensionalidade dos dados, quanto na taxa de acerto, devido à presença de informações redundantes que podem confundir o algoritmo, ao invés de auxiliá-lo na busca de um modelo correto para o conhecimento [6].

A contribuição deste artigo é o de verificar o comportamento do procedimento de otimização com múltiplos objetivos usando o algoritmo de nuvem (ou enxame) de partículas (*Particle Swarm Optimization*, PSO) aplicado à seleção de atributos em mineração de dados.

Neste artigo foram selecionadas dez bases de dados disponíveis no repositório de dados da UCI (*Machine Learning Repository – University of California Irvine*) [7] para validação da abordagem de PSO multiobjetivo. Estas bases foram submetidas a um classificador, primeiramente sem a seleção dos atributos e depois com a seleção de atributos através de otimização com múltiplos objetivos usando nuvem de partículas, assim pode-se comparar os resultados do classificador com a seleção e sem a seleção dos atributos.

O restante do artigo está organizado da seguinte forma. Os fundamentos do PSO clássico e sua abordagem multiobjetivo são apresentados na seção 2. Alguns aspectos relativos relacionados à seleção de atributos são mencionados na seção 3. Na seção 4, os resultados de simulação para seleção de atributos usando o algoritmo PSO multiobjetivo são apresentados e discutidos. Finalizando, a conclusão é abordada na seção 5.

## 2. Otimização usando algoritmo PSO

O algoritmo PSO é uma metodologia originalmente proposta por [5] esta metodologia é baseada em população de soluções, em que cada solução candidata (denominada partícula) possui associda a ela uma velocidade. Esta velocidade é ajustada de acordo com a experiencia da particula correspondente e de acordo com a experiência das outras partículas da população.

O algoritmo PSO utiliza uma população de partículas em um enxame, onde cada partícula $i$ tem sua posição $x_i$ e também a velocidade $v_i$ atualizadas, a cada geração $t$ (iteração do algoritmo), através de fatores de sua própria memória (fator cognitivo), $co_i(t)$, e de seu conhecimento social (fator social), $so_i(t)$. Estes fatores são regidos respectivamente pelas ponderações: social $c_1$ e cognitiva $c_2$.

Resumindo, o comportamento de cada partícula $i$ é definido segundo as equações que se seguem:

$$v_i(t+1) = w \cdot v_i(t) + c_1 \cdot r_1 \cdot co_i(t) + c_2 \cdot r_2 \cdot so_i(t) \quad (1)$$

onde

$$co_i(t) = pbest_i - x_i(t) \quad (2)$$

$$so_i(t) = gbest - x_i(t) \quad (3)$$

$$x_i(t+1) = x_i(t) + \Delta t \cdot v_i(t+1) \quad (4)$$

onde $i$=1,...,*npop*, *npop* é o tamanho da população, $x_i$ e $v_i$ denotam respectivamente a posição e a velocidade da $i$-ésima partícula, *pbest* (*personal best*) e *gbest* (*global best*) são, respectivamente, a melhor posição obtida por uma partícula em uma determinada posição e de toda população em uma determinada vizinhança (enxame); $w$ é o fator de inércia, $r_1$ e $r_2$ são números aleatórios gerados usando uma distribuição uniforme no intervalo [0,1] e $\Delta t$ = 1.

O algoritmo PSO foi inicialmente proposto para otimização mono-objetivo, mas devido a sua simplicidade de implementação, robustez e eficiência, este foi estendido à otimização de múltiplos objetivos em diversas pesquisas. Um apanhado destas pesquisas é apresentado em [10]. Neste artigo é adotada a abordagem de PSO multiobjetivo (MOPSO) proposta em [9] baseado em distância de multidão (*crowding distance*) e arquivo externo (*archiving technique*).

No algoritmo MOPSO, as soluções não dominadas que compõem o conjunto de Pareto ficam armazenadas no arquivo externo. Este arquivo tem outra função que é a de representar o conjunto de Pareto de que a cada iteração as soluções não dominadas são ordenadas conforme um atributo denominado distância de multidão, que representa o quão densa populacionalmente é a região em que a solução se encontra.

## 3. Resultados de Simulação

O pré-processamento dos dados é uma das etapas do processo de descoberta de conhecimento em bases de dados pode ser executada a tarefa de seleção de atributos. A seleção de atributos é importante porque bases de dados costumam conter muitos atributos irrelevantes, assim este processo destina-se a retirar possiveis atributos irrelevantes, refinando assim a mineração de dados.

Segundo [4], a seleção do atributos é uma atividade importante do pré-processamento na mineração dos dados, particularmente na descoberta de conhecimento. Quando se quer descobrir regras que predizam o valor de um atributo meta, é crucial que os atributos usados para descobrir estas regras que sejam relevantes para predizer o atributo meta.

O problema da seleção de atributos é um típico problema de otimização com múltiplos objetivos, uma vez que uma base de dados contém inúmeros atributos, e o foco é validar quais destes atributos devem participar da etapa de mineração de dados, ou seja não existe uma solução única para o problema e sim um conjunto, neste caso um conjunto de atributos.

O método proposto neste artigo tem como objetivo avaliar os efeitos da aplicação dos conceitos do MOPSO adaptado do proposto em [9] na seleção de atributos na tarefa de classificação. Para representar as partículas foi escolhida a representação binária, ou seja, cada partícula possuí características, que são os atributos previsores da base de dados escolhida, se a característica possuir o valor 0, indica que este atributo é ausente na partícula e se possuir o valor igual a 1 indica que esta presente na partícula.

Para o cálculo do *fitness* (problema de minimização dos dois objetivos) para cada partícula foi executado o processo de criação de uma base de treinamento e uma base de teste considerando somente os atributos presentes na partícula. O algoritmo C4.5 [8] foi executado sobre as bases geradas e uma árvore de decisão foi gerada. Por se tratar de um problema multiobjetivo foi adotado como $f_1$ a taxa de erro e como $f_2$ o tamanho da árvore gerada, sendo estes resultados obtidos com a execução do algoritmo C4.5 [8]. A figura 1 mostra como é realizado o cálculo da função de avaliação do método proposto.

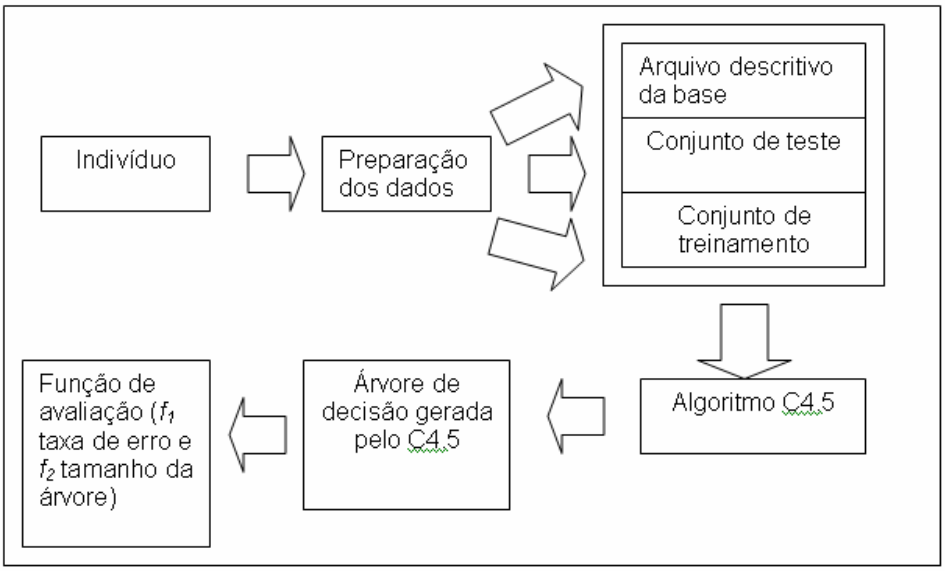


**Figura 1 -** Cálculo da função de avaliação (*fitness*).

No MOPSO, os fatores de aprendizagem ($c_1$ e $c_2$) e o peso (ponderação) de inércia ($w$) foram mantidos constantes, fixados em 2,05 para $c_1$ e $c_2$ e em 0,9 o $w$, estes valores foram adotados baseados em [2].

Na implementação adotada para cada partícula ser avaliada foi necessário criar uma base de dados retirando os atributos ausentes da partícula (foi criada uma base de treinamento e uma de teste), sobre a base foi executado o algoritmo C4.5. Para cada partícula o algoritmo é executado dez vezes, isto porque a base original foi particionada em 10 sub bases mantendo a mesma distribuição das classes, a cada vez que o algoritmo C4.5 é executado, uma sub base é eleita a base de teste, porém nunca é repetida e as demais são agrupadas e geram a base de treinamento. O algoritmo C4.5 é executado sobre a base e uma taxa de erro e um tamanho de árvore são obtidos, estes valores são obtidos a cada vez que o C4.5 é executado, e a média da taxa de erro e a média do tamanho de árvore são utilizados no cálculo do fitness ($f_1$ e $f_2$).

A figura 2 mostra um resumo de como é realizada a seleção dos conjuntos de dados utilizados na validação cruzada.

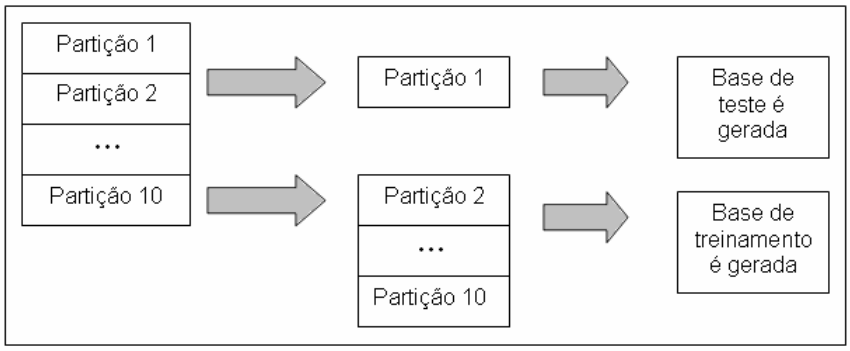


**Figura 2 -** Seleção dos conjuntos de dados na validação cruzada.

O MOPSO foi processado por 500 gerações, com uma população de 100 indivíduos. Para cada base de dados após o processamento do MOPSO foi selecionada a partícula com menor $f_1$, a partícula com menor $f_2$ e a partícula com a menor média de $f_1$ e $f_2$, a partir destas partículas foram geradas 3 árvores de decisão para cada base de dados e 1 árvore de decisão contendo todos os atributos, isto para conseguir validar a eficácia do método proposto. A tabela 1 mostra as características das bases testadas.

| Base de Dados | Quantidade de Atributos | Atributos Discretos | Atributos Categóricos | Número de Classes | Quantidade de Registros |
|---|---|---|---|---|---|
| Crx | 15 | 6 | 9 | 2 | 690 |
| Dermatology | 34 | 1 | 33 | 6 | 366 |
| Glass | 10 | 10 | 0 | 7 | 214 |
| Ionosfhere | 34 | 34 | 0 | 2 | 351 |
| Mushroom | 22 | 0 | 22 | 2 | 8124 |
| Promoters | 57 | 0 | 57 | 2 | 106 |
| Sick-Euthyroid | 25 | 6 | 9 | 2 | 3163 |
| Vehicle | 18 | 18 | 0 | 4 | 846 |
| Votes | 16 | 0 | 16 | 2 | 435 |
| Wine | 13 | 13 | 0 | 2 | 178 |

**Tabela 1** - Características das bases de dados utilizadas.

Para gerar as árvores de decisão foi utilizada a base de dados original dividida em 70% para treinamento e 30% para teste, mantendo proporcional a distribuição das classes, foi escolhida esta divisão, pois esta é justificada na literatura para bases de dados de grande magnitude [1]. Os resultados obtidos são resumidos na tabela 3. Analisando todos os resultados para as bases testadas apresentados na tabela 2 pode-se afirmar que o pior resultado obtido foi na base Vehicle, pois com a redução dos atributos em nenhuma das partículas foi obtido melhor resultado que o obtido com todos os atributos presentes. Nas demais bases pode-se afirmar que os resultados obtidos foram satisfatórios, pois em pelo menos uma das soluções (representada por uma partícula) ocorreu uma redução do percentual de erro ou do tamanho da árvore gerada.

| Bases de Dados | Árvore de decisão gerada | $f_1$ | $f_2$ | Taxa de Erro | Tamanho da árvore | Quantidade de atributos |
|---|---|---|---|---|---|---|
| Crx | Com todos os atributos | - | - | 14,0% | 51 | 15 |
| | Partícula com Menor $f_1$ | 13,04 | 53,90 | 13,0% | 51 | 7 |
| | Partícula com Menor $f_2$ | 44,48 | 1,00 | 44,4% | 1 | 1 |
| | Partícula com Média de $f_1$ e de $f_2$ | 14,46 | 3,00 | 14,5% | 3 | 3 |
| Dermatology | Com todos os atributos | - | - | 4,5% | 14 | 34 |
| | Partícula com Menor $f_1$ | 2,46 | 16,00 | 4,5% | 14 | 18 |
| | Partícula com Menor $f_2$ | 7,07 | 11,30 | 6,3% | 11 | 13 |
| | Partícula com Média de $f_1$ e de $f_2$ | 2,75 | 14,00 | 4,5% | 14 | 13 |
| Glass | Com todos os atributos | - | - | 3,0% | 11 | 10 |
| | Partícula com Menor $f_1$ | 1,91 | 11,40 | 3,0% | 11 | 2 |
| | Partícula com Menor $f_2$ | 53,66 | 3,20 | 53,0% | 3 | 1 |
| | Partícula com Média de $f_1$ e de $f_2$ | 2,36 | 11,00 | 3,0% | 11 | 3 |
| Ionosphere | Com todos os atributos | - | - | 6,7% | 27 | 34 |
| | Partícula com Menor $f_1$ | 5,35 | 19,60 | 3,8% | 19 | 13 |
| | Partícula com Menor $f_2$ | 6,22 | 16,40 | 6,7% | 23 | 15 |
| | Partícula com Média de $f_1$ e de $f_2$ | 5,42 | 17,80 | 6,7% | 15 | 12 |
| Mushroom | Com todos os atributos | - | - | 11,2% | 14 | 22 |
| | Partícula com Menor $f_1$ | 0,00 | 15,00 | 11,2% | 14 | 9 |
| | Partícula com Menor $f_2$ | 0,04 | 12,00 | 9,2% | 12 | 13 |
| | Partícula com Média de $f_1$ e de $f_2$ | 0,00 | 13,00 | 11,2% | 13 | 13 |
| Promoters | Com todos os atributos | - | - | 15,6% | 18 | 57 |
| | Partícula com Menor $f_1$ | 10,17 | 15,90 | 18,8% | 16 | 29 |
| | Partícula com Menor $f_2$ | 13,50 | 12,90 | 15,6% | 17 | 26 |
| | Partícula com Média de $f_1$ e de $f_2$ | 10,33 | 15,60 | 12,5% | 15 | 29 |
| Sick-euthyroid | Com todos os atributos | - | - | 2,4% | 27 | 25 |
| | Partícula com Menor $f_1$ | 2,09 | 26,00 | 2,2% | 25 | 14 |
| | Partícula com Menor $f_2$ | 9,29 | 1,00 | 9,3% | 1 | 14 |
| | Partícula com Média de $f_1$ e de $f_2$ | 2,56 | 9,40 | 2,4% | 7 | 11 |
| Vehicle | Com todos os atributos | - | - | 21,4% | 155 | 18 |
| | Partícula com Menor $f_1$ | 26,63 | 62,40 | 28,2% | 165 | 11 |
| | Partícula com Menor $f_2$ | 31,91 | 8,40 | 31,7% | 171 | 9 |
| | Partícula com Média de $f_1$ e de $f_2$ | 31,91 | 8,40 | 31,7% | 171 | 9 |
| Votes | Com todos os atributos | - | - | 2,3% | 11 | 16 |
| | Partícula com Menor $f_1$ | 3,20 | 10,60 | 2,3% | 11 | 6 |
| | Partícula com Menor $f_2$ | 4,35 | 3,00 | 3,8% | 3 | 8 |
| | Partícula com Média de $f_1$ e de $f_2$ | 4,35 | 3,00 | 3,8% | 3 | 8 |
| Wine | Com todos os atributos | - | - | 15,1% | 11 | 13 |
| | Partícula com Menor $f_1$ | 3,35 | 11,00 | 5,7% | 9 | 3 |
| | Partícula com Menor $f_2$ | 3,41 | 10,00 | 1,9% | 9 | 4 |
| | Partícula com Média de $f_1$ e de $f_2$ | 3,41 | 10,00 | 1,9% | 9 | 8 |

**Tabela 2** - Resumo dos resultados obtidos para as dez bases de dados.

## 5. Conclusão

Este trabalho propôs um algoritmo de nuvem de partículas multiobjetivo para seleção de atributos em tarefas de classificação. Neste contexto, o principal objetivo era avaliar o comportamento do algoritmo para a seleção de atributos e validar se tal algoritmo foi eficiente na seleção dos atributos. Como funções objetivo foram definidas a taxa de erro e o tamanho de árvore, sendo estes gerados pelo algoritmo C4.5.

O C4.5 foi definido como base de comparação para as soluções encontradas pelo método proposto, os experimentos realizados mostraram que o método proposto conseguiu encontrar soluções melhores que a padrão (consistindo de todos os atributos) em seis das dez bases da UCI utilizadas nos experimentos. Em duas bases de dados, os resultados foram semelhantes a solução padrão e em duas bases os resultados foram piores que a solução padrão. Portanto, os resultados deste trabalho fornecem uma evidência que o método proposto baseado em nuvem de partículas, no caso da tarefa de seleção de atributos, podem também ser utilizados com eficácia e eficiência usando critérios de otimização multiobjetivo.

## Referências:


[1] Carvalho, D. R. (2005). Árvore de decisão / algoritmo genético para tratar o problema de pequenos disjuntos em classificação de dados, **Tese de Doutorado**, COPPE, Universidade Federal do Rio de Janeiro - UFRJ, Rio de Janeiro, RJ.

[2] Eberhart, R. C.; Shi, Y., Kennedy, J. (2001). **Swarm intelligence**. Morgan Kaupmann Publishers, USA.

[3] Fayyad, U. M.; Piatetsky-Shapiro, G.; Smyth, P.; Uthurusamy, R. (1996). **Advances in knowledge discovery and data mining**. American Association for Artificial Intelligence, Menlo Park, CA, USA, 1996.

[4] Freitas, A. A. (2002). **Data mining and knowledge discovery with evolutionary algorithms**, Natural Computing Series, Springer-Verlag, Berlin, Germany.

[5] Kennedy, J.; Eberhart, R. C. (1995). Particle Swarm Optimization. **Proceedings of IEEE International Conference on Neural Networks**, Washington, DC, USA.

[6] Kira, K.; Rendell, L. A. (1992). The feature selection problem: traditional methods and a new algorithm, **Proceedings of 10th Conference on Artificial Intelligence**, Menlo Park, CA, USA, p. 129-136.

[7] Murphy, P. M.; Aha, D. W. (1994). **UCI repository of machine learning databases.** [http://www.ics.uci.edu/~mlearn/MLRepository.html]. University of California, Department of Information and Computer Science, Irvine, CA, USA.

[8] Quilan, J. R. (1993). **C4.5: Programs for machine learning**, Morgan Kaufmann Publisher, San Francisco, CA, USA.

[9] Raquel, C. R.; Naval Jr., P. C. (2005). An effective use of crowding distance in multiobjective particle swarm optimization, **Proceedings of Genetic and Evolutionary Computation Conference (GECCO 2005)**, Washington DC, USA.

[10] Reyes-Serra, M.; Coello, C. A. C. (2006). Multi-objective particle swarm optimizers: a survey of the state-of-the-art, **International Journal of Computational Intelligence Research**, 2(3): 287-308.