

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ

VIVIANE DAL MOLIN DE SOUZA

NUVEM DE PARTÍCULAS APLICADA À SELEÇÃO DE ATRIBUTOS

**CURITIBA
2008**

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

VIVIANE DAL MOLIN DE SOUZA

NUVEM DE PARTÍCULAS APLICADA À SELEÇÃO DE ATRIBUTOS

Dissertação apresentada ao Curso de Pós-Graduação em Engenharia de Produção e Sistemas do Centro de Ciências Exatas e de Tecnologia da Pontifícia Universidade Católica do Paraná, como requisito parcial à obtenção do título de Mestre em Engenharia de Produção e Sistemas.

ORIENTADOR: Prof. Leandro dos Santos Coelho
CO-ORIENTADORA: Prof. Deborah Ribeiro Carvalho

**CURITIBA
2008**

VIVIANE DAL MOLIN DE SOUZA

NUVEM DE PARTÍCULAS APLICADA À SELEÇÃO DE ATRIBUTOS

Dissertação apresentada ao Curso de Pós-Graduação, em Engenharia de Produção e Sistemas da Pontifícia Universidade Católica do Paraná, como requisito à obtenção do título de Mestre.

Comissão Examinadora

Prof. Dr. Leandro Dos Santos Coelho
Pontifícia Universidade Católica do Paraná
Orientador

Prof. Dr. Deborah Ribeiro Carvalho
Universidade Tuiuti do Paraná
Co-orientadora

Prof. Dr. Aurora Trinidad Ramirez Pozo
Universidade Federal do Paraná

Curitiba, 17 de setembro de 2008.

AGRADECIMENTOS

Agradeço primeiramente à Deus, pois sem Ele com certeza não teria conseguido chegar ao final deste trabalho, tenho certeza que foi graças a Ele que nos momentos mais difíceis não deixei de lutar.

Agradeço ao Prof. Dr. Leandro dos Santos Coelho pela orientação acadêmica, constante apoio, disponibilidade e por tão prontamente me orientar.

Agradeço o apoio, o carinho e o incentivo dos meus pais Luiz e Soeli que estiveram presentes em todos os momentos.

Agradeço meu esposo Daniel e minha filha Alana pela compreensão, pelos dias que abdicaram da minha presença para que eu pudesse chegar ao final deste trabalho. Agradeço todo carinho e apoio que vocês me deram neste período. Com certeza vocês foram uma das principais motivações para eu persistir e chegar até o fim. Amo vocês!

SUMÁRIO

1	INTRODUÇÃO	10
2	REVISÃO DA LITERATURA DE MÉTODOS PARA MINERAÇÃO DE DADOS	13
2.1	Árvore de Decisão	14
2.2	Algoritmos Genéticos (AGs)	16
2.2.1	Função de <i>Fitness</i>	19
2.2.2	Seleção	20
2.2.2.1	Seleção Baseada em Roleta	20
2.2.2.2	Seleção Baseada em Torneio	21
2.2.3	Elitismo	21
2.2.4	Operadores Genéticos	21
2.2.4.1	Cruzamento (<i>crossover</i>)	22
2.2.4.2	Mutação	23
2.3	Otimização por Nuvem de Partículas	24
2.3.1	Topologia Estrela (<i>gbest</i>)	30
2.3.2	Topologia em Roda	30
2.3.3	Topologia em Círculo (<i>lbest</i>)	31
2.3.4	Topologia Aleatória	32
2.4	Algoritmos Evolutivos Aplicados à Mineração de Dados	32
3	OTIMIZAÇÃO DE MÚLTIPLOS OBJETIVOS	37
3.1.1	Conceitos Básicos sobre Otimização de Múltiplos Objetivos	38
3.1.2	Otimização de Múltiplos Objetivos Usando Nuvem de Partículas	41
4	MÉTODO PROPOSTO	53
4.1	Abordagem Multiobjetivo de Nuvem de Partículas para Mineração de Dados	55
4.1.1	Codificação do Indivíduo ou Partícula	55
4.1.2	Função de <i>Fitness</i>	55
4.1.3	Método de Seleção	56
4.1.4	Fator de Apendizagem e Peso da Inércia	57
4.1.5	Pseudo-Código	57
5	EXPERIMENTOS E RESULTADOS OBTIDOS	59
5.1	Base CRX	60
5.2	Base Dermatology	61
5.3	Base Glass	63
5.4	Base Ionosphere	65
5.5	Base Mushroom	67
5.6	Base Promoters	69
5.7	Base Sick-Euthyroid	71
5.8	Base Vehicle	73
5.9	Base Votes	75
5.10	Base Wine	77
5.11	Resumo dos Resultados Obtidos Considerando Erro e Tamanho das Árvore	79
5.12	Análise Considerando Métodos de Desempenho	82
5.13	Resultados Obtidos Com 5 Experimentos	85

6	CONCLUSÃO E RECOMENDAÇÕES PARA TRABALHOS FUTUROS	87
7	REFERÊNCIAS BIBLIOGRÁFICAS	88

LISTA DE TABELAS

Tabela 1 – Sumário das aproximações de MOPSO (Reyes-Sierra e Coello, 2006) adaptado.....	52
Tabela 2 – Características das bases de dados utilizadas.....	59
Tabela 3 – Resultados para a base CRX.	60
Tabela 4 – Resultados para a base Dermatology.....	62
Tabela 5 – Resultados para a base Glass.....	64
Tabela 6 – Resultados para a base Ionosphere.....	66
Tabela 7 – Resultados para a base Mushroom.....	68
Tabela 8 – Resultados para a base Promoters.	70
Tabela 9 – Resultados para a base Sick-Euthyroid.....	72
Tabela 10– Resultados para a base Vehicle.	74
Tabela 11– Resultados para a base Votes.....	76
Tabela 12– Resultados para a base Wine.....	78
Tabela 13 – Resumo dos resultados obtidos.....	80
Tabela 14 – Resumo dos resultados obtidos.....	81
Tabela 15 – Espaçamento e distância euclidiana média a origem dos eixos cartesianos.	84

LISTA DE FIGURAS

Figura 1 – Etapas do processo de KDD (Rezende, 2003) adaptado.....	14
Figura 2 – Exemplo de árvore de decisão.....	16
Figura 3 – Pseudo-código de um algoritmo genético simples.....	19
Figura 4 – Exemplo de cruzamento com um ponto de corte.....	22
Figura 5 – Exemplo de mutação.....	24
Figura 6 – Pseudo-código básico de um PSO.....	29
Figura 7 – Topologia estrela.....	30
Figura 8 – Topologia em roda.....	31
Figura 9 – Topologia em círculo.....	32
Figura 10 – Topologia aleatória.....	32
Figura 11 – Fronteira de Pareto (Castro, 2001).....	41
Figura 11 – Pseudo-código de um MOPSO.....	43
Figura 12 – Partícula representando atributos selecionados da base Íris.....	55
Figura 13 – Cálculo do <i>fitness</i>	56
Figura 14 – Pseudo-código do método PSO multiobjetivo.....	57
Figura 15 – Seleção dos conjuntos de dados na validação cruzada.....	58
Figura 16 – Conjunto de soluções obtidas para a base CRX.....	61
Figura 17 – Conjunto de soluções obtidas para a base Dermatology.....	63
Figura 18 – Conjunto de soluções obtidas para a base Glass.....	65
Figura 19 – Conjunto de soluções obtidas para a base Ionosphere.....	67
Figura 20 – Conjunto de soluções obtidas para a base Mushroom.....	69
Figura 21 – Conjunto de soluções obtidas para a base Promoters.....	71
Figura 22 – Conjunto de soluções obtidas para a base Sick-Euthyroid.....	73
Figura 23 – Conjunto de soluções obtidas para a base Vehicle.....	75
Figura 24 – Conjunto de soluções obtidas para a base Votes.....	77
Figura 25 – Conjunto de soluções obtidas para a base Wine.....	79
Figura 26 – Conjunto de Pareto para a base Sick-Euthyroid.....	85
Figura 27 – Conjunto de Pareto para a base Vehicle.....	86

LISTA DE ACRÔNIMOS

AE – Algoritmo Evolucionário

AG – Algoritmo Genético

MOPSO – *Multi-Objective Particle Swarm Optimization*

PSO – *Particle Swarm Optimization*

SI – *Swarm Intelligence*

KDD – *Knowledge Discovery in Database*

DPSO – *Discrete Particle Swarm Optimiser*

LDWPSO - *Linear Decreasing Weight Particle Swarm Optimiser*

CPSO – *Constricted Particle Swarm Optimiser*

UCI – *Machine Learning Repository – University of California Irvine*

MOEA – *Multi-objective Evolutionary Algorithm*

DPSO – *Discrete Particle Swarm Optimiser*

LDWPSO – *Linear Decreasing Weight Particle Swarm Optimiser*

CPSO – *Constricted Particle Swarm Optimiser*

DOEA – *Dual-Objective Evolutionary Algorithm*

NSGA-II – *Non-dominated Sorting Genetic Algorithm II*

PAES – *Pareto Archived Evolution Strategy*

MOGA2 – *Multi-Objective Genetic Algorithm 2*

KEA – *Kit for Evolutionary Algorithms*

VEGA – *Vector Evaluated Genetic Algorithm*

GRASP-PR – *Greedy Randomize Adaptive Search Procedure with Path-Relinking*

MOPS – *Multiple Objective Particle Swarm*

RESUMO

O algoritmo de otimização através de nuvem de partículas (PSO) é uma técnica metaheurística desenvolvida recentemente e pertence a categoria de técnicas inteligentes baseadas em enxames. Os enxames inteligentes são conceitos inspirados no comportamento social dos animais, tais como em enxames de aves, peixes e formigas. O PSO é um algoritmo que explora a população com base em uma população de indivíduos com buscas promissoras nas regiões da pesquisa espacial. O indivíduo é afetado pelo comportamento do melhor local ou pelo comportamento dos melhores globais individuais. O desempenho de cada indivíduo é medido usando uma função de *fitness*, semelhante aos algoritmos evolutivos. A população é tratada como um enxame ou nuvem e os indivíduos são nomeados de partículas. As partículas deslocam-se em um espaço multi-dimensional de pesquisa com velocidade adaptável. No PSO, as partículas lembram a melhor posição no passado e a melhor posição nunca alcançada pelas partículas. Esta propriedade ajuda as partículas na busca espacial multi-dimensional, deixando-a mais rápida. PSO pode ser útil em uma ampla variedade de tarefas de otimização e em muitos domínios, tais como: a função de otimização não linear, formação de redes neurais artificiais, *design* de sistemas *fuzzy*, programação de problemas, problema do caixeiro viajante, entre outros. Devido a sua habilidade natural de convergir rapidamente, o algoritmo PSO é também utilizado para resolver problemas de otimização multiobjetivo. Vários problemas envolvem otimização simultânea de múltiplos objetivos que muitas vezes são concorrentes. Em um problema de otimização multiobjetivo, pode não existir a melhor solução no que diz respeito a todos os objetivos. Normalmente, o objetivo é determinar a superfície de busca, que é um conjunto de pontos não dominados da solução, conhecida como Pareto-ótima. Levando em consideração que nenhuma das soluções do conjunto não dominado é melhor que qualquer outra solução, qualquer uma das soluções é aceitável. A escolha de uma solução requer um conhecimento de uma série de fatores relacionados ao problema. Recentemente, foram realizadas muitas tentativas para aplicar as abordagens do 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 desta dissertação é a seleção de atributos usando a abordagem de PSO baseada em otimização multiobjetivo e variáveis inteiras para a seleção e avaliação dos atributos selecionados. Além disso, esta dissertação apresenta igualmente uma análise de influência na seleção de atributos em uma tarefa de *data mining*. A proposta do método de seleção de atributos utilizando abordagem de PSO baseada em otimização 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. Os resultados encontrados mostraram que a proposta de abordagem PSO encontrou soluções melhores do que a solução completa (solução com todos os atributos), em seis das dez bases de dados da UCI. Além disso, a qualidade dos resultados obtidos pela simulação PSO foi semelhante à solução completa em duas bases de dados, mas em outras duas bases de dados, a solução apresentada pelo PSO foram piores que a solução completa.

Palavras-chave: Seleção de atributos, mineração de dados, otimização multiobjetivo, otimização por nuvem 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, ants and fish school. PSO is a population based algorithm that exploits a population of individuals to probe promising regions of the search space. The individual behavior is affected either by the best-local or best-global individual. The performance of each individual is measured using fitness function similar to evolutionary algorithms. The population is referred as a swarm and individuals are called particles. The particles move in a multi-dimensional search space with adaptable velocity. In PSO, the particles remember the best position in the past and the best position ever attained by the particles. This property helps the particles to search the multi-dimensional space faster. PSO has been found to be useful in a wide variety of optimization tasks in many fields, such as nonlinear function optimization, artificial neural network training, fuzzy system design, scheduling problems, traveling salesman problem, among others. Due to its natural ability to converge faster, PSO algorithm is also used to solve multi-objective optimization problems. Several problems involve simultaneous optimization of multiple objectives that often are competing. In a multi-objective optimization problem, there may not exist one solution that is best with respect to all objectives. Usually, the aim is to determine the tradeoff surface, which is a set of nondominated solution points, known as Pareto-optimal or noninferior solutions. In view of the fact that none of the solutions in the nondominated set is absolutely better than any other, any one of them is an acceptable solution. The choice of one solution over the other requires problem knowledge and a number of problem-related factors. Recently, any investigations have been undertaken to apply of PSO 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 post-processing. The objective of this dissertation is the attributes selection using a PSO approach based on multi-objective optimization and integer variables for selection and evaluation of selected attributes. Furthermore, this dissertation presents also an analysis of attributes selection influence in data mining tasks. The proposed attributes selection method based on 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. Simulation results showed that the proposed PSO approach found better solutions than the full solution (solution based on all attributes) in six of ten databases of UCI. Moreover, the quality of simulation results obtained by PSO were similar to the full solution in two databases, but in other two databases the solution presented by PSO were worst than the full solution.

Keywords: Attributes selection, data mining, multi-objective optimization, 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 à esta pesquisa denomina-se Descoberta de Conhecimento em Bases de Dados (KDD - *Knowledge Database Discovery*).

Segundo Fayyad (1996), o KDD é composto por diversas etapas, que são as seguintes:

- Seleção da base de dados e seleção dos atributos: etapa em que se efetua a seleção e coleta dos dados necessários;
- 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;
- 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;
- 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;
- 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;

- 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 (Kira e Rendell, 1992).

O principal objetivo desta dissertação é verificar o comportamento do procedimento de otimização com múltiplos objetivos usando nuvem de partículas aplicado à seleção de atributos em mineração de dados. Foram selecionadas 10 bases de dados disponíveis no repositório de dados da UCI (Murphy e Aha, 1994), sendo elas: CRX, Dermatology, Glass, Ionosphere, Mushroom, Promoters, Sick-Euthyroid, Vehicle, Votes e Wine. 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. Esta proposta foi motivada pelos promissores resultados obtidos com a utilização de algoritmos evolutivos multiobjetivos na otimização de diferentes problemas das mais variadas áreas do conhecimento.

Esta dissertação está organizada da seguinte forma,

No capítulo 2 é apresentada uma revisão da literatura de métodos para mineração de dados, sendo abordados o processo de KDD. Depois árvores de decisão, que são meios utilizados para representar os resultados obtidos através da mineração de dados e por fim MOPSO (*Multi-Objective Particle Swarm Optimization*), que foram inspirados em alguns processos de evolução natural, os MOPSO são métodos de otimização por busca inspirados nos mecanismos de evolução de populações de seres vivos. Os fundamentos de nuvem de partículas são também apresentados neste capítulo.

No capítulo 3 é abordada a otimização com múltiplos objetivos, mostrando a sua importância e motivação para sua utilização e aplicação, uma vez que diversos dos problemas presentes nas áreas comercial e industrial a serem resolvidos possuem mais que um objetivo a ser considerado e otimizado.

No capítulo 4 é apresentado o método proposto, descrevendo como o mesmo foi elaborado e executado.

No capítulo 5 são apresentados os resultados obtidos após o processamento do método em bases disponíveis no repositório de dados da UCI (Murphy e Aha, 1994).

2 REVISÃO DA LITERATURA DE MÉTODOS PARA MINERAÇÃO DE DADOS

Essencialmente, a mineração dos dados consiste na extração (semi) automática do conhecimento dos dados. Esta afirmação pode levar ao seguinte questionamento: tipo do conhecimento deve-se tentar descobrir? Embora esta seja uma pergunta subjetiva, pode-se mencionar propriedades gerais as quais o conhecimento descoberto deve satisfazer incluindo-se que o conhecimento deve ser compreensível e interessante. Na mineração de dados, o foco é descobrir o conhecimento que possui algum poder preditivo, ou seja, prever o valor de um ou mais atributos, baseado em atributos previamente observados. Neste contexto, é desejado que o conhecimento descoberto possua uma alta taxa de exatidão e que seja compreensível para o usuário. Isto é importante e necessário sempre que o conhecimento descoberto servirá como suporte a procedimentos para a tomada de decisão.

Segundo Fayyad (1996), o KDD representa todo o processo de descoberta de conhecimento em base de dados e objetiva a extração do conhecimento implícito, desconhecido e potencialmente útil, tornando assim o conhecimento implícito em conhecimento explícito. O conhecimento implícito é um conceito que descreve o conhecimento que, embora ainda não tenha sido documentado, é passível de o ser. O conhecimento explícito é o conhecimento adquirido que tem-se consciência.

No processo de KDD, um objetivo é fundamental, ou seja, saber o que se quer descobrir e o que realizado feito com o conhecimento obtido. O êxito deste processo depende da participação do especialista do domínio da aplicação. Na Figura 1 são mostradas as etapas que compõem o processo de KDD, em que primeiramente identifica-se o problema (seleciona-se a base de dados e os atributos). Após esta base de dados passará pela etapa de pré-processamento, (nesta etapa são retirados os ruídos existentes na base). A próxima etapa será a de extração de padrões (processamento), depois existe a etapa de pós-processamento, etapa esta que destina-se a análise e avaliação do conhecimento descoberto. Finalmente, a etapa de utilização do conhecimento descoberto é realizada.

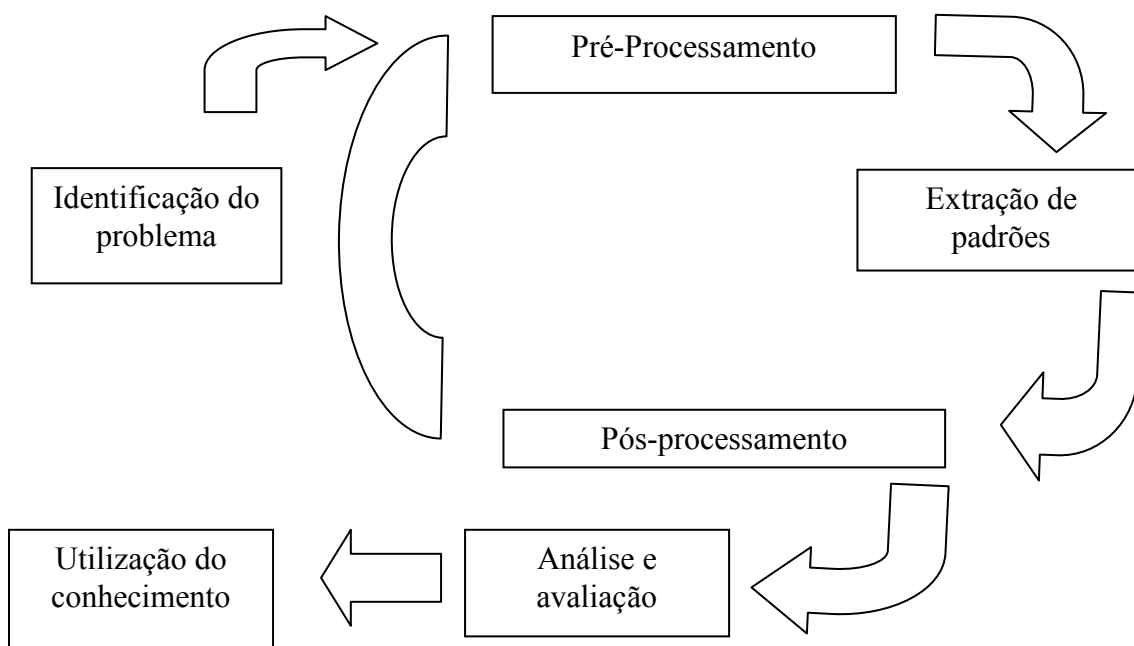


Figura 1 – Etapas do processo de KDD (Rezende, 2003) adaptado.

Para Gureck (2001) *apud* Fayyad (1996), o processo de KDD é interativo, pois permite que os usuários tomem decisões, e iterativo, porque envolve 6 etapas, podendo existir retorno a etapas anteriores.

2.1 Árvore de Decisão

As árvores de decisão são meios de representar os resultados de mineração de dados na forma de árvores. Dado um grupo de dados com numerosas colunas e linhas, uma ferramenta de árvore de decisão pede ao usuário para escolher uma das colunas como objeto de saída (atributo meta), e então mostra o único e mais importante fator correlacionado com aquele objeto de saída como primeiro ramo (nó) da árvore de decisão. Os outros fatores são subsequentemente classificados como nós do(s) nó(s) anterior(es). Desta maneira, o usuário pode entender o porquê do fator escolhido, e vai permitir que o usuário explore a árvore de acordo com suas necessidades de projeto, do mesmo modo como ele poderá

determinar grupos alvos. Os usuários podem também selecionar os dados fundamentais em qualquer nó da árvore, movendo-os para uma planilha ou outra ferramenta para posterior análise (Marath e Mayer, 2001).

Para Quilan (1993), uma árvore de decisão é uma representação simples de um classificador utilizado por diversos sistemas de aprendizado de máquina, como por exemplo, o C4.5 (Quilan, 1993).

Para Carvalho (2002), uma árvore de decisão é induzida a partir de um conjunto de exemplos de treinamento em que as classes são previamente conhecidas. A estrutura da árvore é organizada da seguinte forma:

- cada nó interno (não-folha) é rotulado com o nome de um dos atributos previsores;
- os ramos (ou arestas) saindo de um nó interno são rotulados com valores do atributo naquele nó;
- cada folha é rotulada com uma classe, a qual é a classe prevista para exemplos que pertençam àquele nó folha.

O processo de classificação de um exemplo ocorre fazendo aquele exemplo percorrer a árvore, a partir do nó raiz, procurando avaliar os arcos que unem os nós, de acordo com as combinações que estes mesmos arcos representam. Ao atingir um nó folha, a classe que rotula aquele nó folha é atribuída àquele exemplo (Carvalho, 2002).

Na figura 2 é apresentada uma representação de árvore de decisão. Esta árvore de decisão classifica os dias em adequados ou não para a prática de esportes ao ar livre.

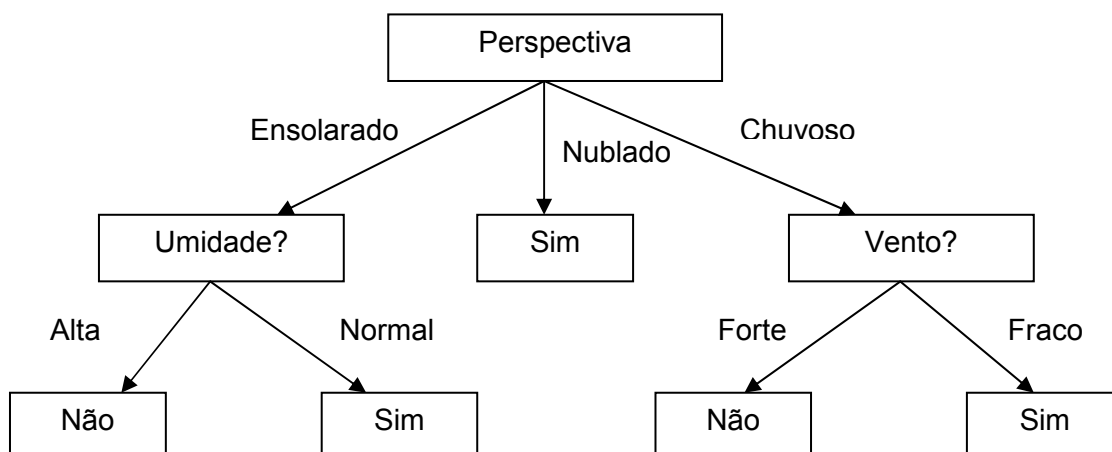


Figura 2 – Exemplo de árvore de decisão.

2.2 Algoritmos Genéticos (AGs)

Os AGs foram concebidos a partir da analogia natural, sendo uma área denominada computação evolucionária ou evolutiva. Os AGs são métodos de otimização por busca inspirados nos mecanismos de evolução de populações de seres vivos. Estes seguem o princípio da seleção natural e sobrevivência dos seres mais aptos.

Na natureza o processo de seleção natural controla a evolução de seres vivos. Os organismos mais adaptados ao seu ambiente tendem a viver tempo suficiente para se reproduzirem, enquanto organismos menos adaptados são mais propensos a morrer antes de sua reprodução (Goldberg, 1989).

Segundo Falkenauer *apud* Carvalho (2002), os AGs são algoritmos de busca baseados no mecanismo da seleção natural e na genética. Os AGs baseiam-se na sobrevivência da melhor solução candidata para um determinado problema. As soluções candidatas são normalmente representadas por indivíduos. A cada nova geração um novo conjunto de indivíduos é criado usando segmentos (partes) dos melhores indivíduos da geração anterior, conforme avaliado por uma função de *fitness* (função objetivo ou função de adequação).

Um AG difere dos algoritmos de busca tradicionais, principalmente, nos seguintes aspectos (Goldberg, 1989):

- AGs realizam uma busca usando uma população de solução, e não uma única solução;
- AGs usam operadores probabilísticos e não operadores determinísticos; e
- AGs usam diretamente a informação de uma função objetivo, e não dependem de derivadas ou conhecimento auxiliar sobre o problema.

A primeira e a segunda característica mencionadas contribuem para a robustez dos AGs, ajudando-os a escaparem de pontos de ótimo locais, a fim de alcançarem pontos ótimos globais. A terceira característica contribui para a generalidade de AGs que podem ser aplicados em um largo espectro de problemas de otimização, busca e aprendizagem de máquina (Carvalho, 2002).

Os principais mecanismos de um AG convencional são fáceis de entender, pois envolvem apenas trocas parciais de indivíduos e pequenas alterações de elementos dos indivíduos (Carvalho, 2002).

A seguir são listados alguns procedimentos usuais em sistemas que utilizam algoritmos genéticos (Berson e Smith, 1997):

1. Definir o problema a ser resolvido – codificar uma solução candidata em um indivíduo, e especificar uma função de *fitness*.
2. Iniciar a população – tipicamente atribuindo valores aleatórios aos indivíduos da população inicial.
3. Permitir a seleção dos mais aptos e extinção dos menos aptos.
4. Permitir que uma nova geração seja formada aplicando-se a seleção e os operadores de cruzamento e mutação aos indivíduos da geração anterior.
5. Parar quando algum(ns) do(s) seguintes critérios for(em) atingido(s):
 - A solução é boa o suficiente;
 - O AG atingiu o número determinado de gerações;
 - O AG não consegue mais evoluir.

Senão retornar ao passo 3.

Um AG é caracterizado pela sua capacidade de otimização de dados. Há dois meios para que esse processo de otimização possa ser realizado: pelo meio matemático e pelo meio heurístico. Quando a base de dados é puramente baseada em dados matemáticos, a aplicação de otimização pelo meio matemático pode gerar soluções ótimas. Porém, quando a base de dados dispõe de outros tipos de dados, a otimização heurística oferece uma solução mais eficiente (Goldberg, 1989).

Uma população aleatória normalmente é gerada e com cálculos de adequabilidade de cada indivíduo se faz combinações para gerar novos subconjuntos, e assim fazer com que futuras gerações tenham melhores resultados finais na mutação, tendo assim, um resultado, muitas vezes, mais adequado para que se possam fazer futuras análises (Goldberg, 1989).

A modelagem de um AG consiste da definição de 4 componentes principais (Pacheco *et al.*, 1999):

- Representação do cromossomo (ou indivíduo) – estrutura de dados que representa uma solução do problema;
- Avaliação – medida da qualidade de uma determinada solução;
- Operadores genéticos – atuam no processo de reprodução criando e/ou alterando indivíduos;
- Geração da população – o método que determina a população inicial.

Um exemplo de pseudocódigo para um AG é apresentado na figura 3.

```
AG
{
  t = 0;
  IniciarPopulação(t);
  CalculaFitness(t);
  Enquanto (não satisfaz condição de parada)
  {
    t = t + 1;
    Elitismo (t -1);
    Enquanto (NúmeroDeIndivíduos < TamanhoDaPopulação)
    {
      Crossover (t);
      Mutação (t);
      CalculaFitness (t);
    }
  }
}
```

Figura 3 – Pseudo-código de um algoritmo genético simples.

2.2.1 Função de *Fitness*

A função de *fitness* é responsável pela avaliação do quão bom é o indivíduo, ele fornece um valor que irá indicar a qualidade do indivíduo avaliado, logo todo o processo de seleção é diretamente influenciado pela função de *fitness*. Esta função é específica para cada aplicação do AG, assim deve ser definida para cada tipo de problema a ser resolvido.

Segundo Locatelli e Neto (2004), o *fitness* é uma medida de desempenho em que um valor numérico é associado a cada indivíduo. Esse valor determina quão boa é a sua capacidade de evolução, ou seja, o quanto esse indivíduo pode contribuir na geração de novos indivíduos.

2.2.2 Seleção

Segundo Ticona (2003), a seleção é o processo em que os indivíduos com melhor valor de aptidão têm maior probabilidade de gerar um ou mais descendentes para a geração seguinte. Este operador é a versão inspirada pela seleção das espécies do Darwinismo, que estabelece que os indivíduos mais aptos têm maiores chances de sobreviver, ou seja, ou mais fortes e menos vulneráveis aos predadores e doenças. Sendo assim, o principal objetivo da seleção é copiar boas soluções para a geração futura, eliminando as soluções menos adaptadas ou aptas.

Nas subseções 2.2.2.1 e 2.2.2.2 serão vistos dois métodos de seleção: a seleção baseada em roleta e a seleção baseada em torneio.

2.2.2.1 Seleção Baseada em Roleta

Os indivíduos são copiados para a próxima geração de acordo com probabilidades proporcionais ao seu valor de função de avaliação. A implementação desse método é normalmente realizada através de um mecanismo de roleta, na qual a roleta é dividida em N partes, N correspondendo ao número de indivíduos da população, e o tamanho de cada uma das partes é proporcional à função de avaliação que o indivíduo representa. A roleta é então girada N vezes, e a cada uma delas o indivíduo indicado pelo ponteiro é selecionado e inserido na nova população (Goldberg, 1989).

2.2.2.2 Seleção Baseada em Torneio

Aplicando-se o torneio, um grupo de q indivíduos são obtidos aleatoriamente da população, com reposição (o indivíduo retorna à população para possivelmente participar em outros torneios) ou sem reposição (o indivíduo selecionado não pode participar novamente em outros torneios). Estes q indivíduos participam do torneio, em que o indivíduo com a melhor função de avaliação é escolhido para a reprodução, e o processo se repete até uma nova população ser gerada (Goldberg, 1989).

2.2.3 Elitismo

O operador de elitismo mantém as melhores soluções encontradas previamente nas gerações futuras, implicando que a melhor solução de uma geração não é perdida nas gerações seguintes.

A sobrevivência de um indivíduo depende diretamente do gerenciamento da população. A cada geração, o indivíduo é modificado através de cruzamento ou mutação, o que pode resultar na perda da solução ótima. Para evitar essa perda, surgiu a estratégia elitista que, através do cálculo de seu *fitness*, garante que os melhores indivíduos da população permanecerão intactos na geração seguinte, garantindo que boas soluções não sejam perdidas durante o processo de evolução (Goldberg, 1989).

2.2.4 Operadores Genéticos

Um AG simples é constituído de dois operadores genéticos básicos: cruzamento e mutação (Goldberg, 1989). A seguir serão descritos estes operadores.

2.2.4.1 Cruzamento (*crossover*)

O cruzamento permite a criação de novos indivíduos para a população a partir da recombinação de outros indivíduos, ou seja, permite a troca de material genético já disponível na população. Isto ocorre quando dois indivíduos previamente selecionados (denominados de cromossomos pais), são divididos em determinada parte da *string* – escolhida aleatoriamente, e assim são trocadas as informações entre as duas metades dos indivíduos, gerando dois novos indivíduos (cromossomos filhos).

O cruzamento pode ocorrer quando a *string* é quebrada uma vez (cruzamento de um ponto), quando a *string* é quebrada duas vezes (cruzamento com dois pontos de corte), ou quando a *string* é quebrada em x segmentos (onde $x > 2$). Pode também ocorrer o cruzamento uniforme, em que todos os genes têm a mesma probabilidade de serem trocados, independentes de sua posição.

O cruzamento é realizado a partir de um parâmetro genético, denominada taxa de cruzamento. A figura 4 mostra um exemplo de cruzamento de um ponto, em um AG com representação binária.

PAIS	CRUZAMENTO	FILHOS
1 0 1 0 1 1 0	→	1 0 1 0 0 1 1
1 1 0 0 0 1 1	→	1 1 0 0 1 1 0

Figura 4 – Exemplo de cruzamento com um ponto de corte.

Existem diversos tipos de cruzamento, a seguir são comentados alguns destes tipos para AG com representação binária:

- Cruzamento com um ponto de corte: escolhe-se aleatoriamente um ponto de cruzamento (ou seja, uma posição da seqüência que será dividida) de modo que o material além deste ponto seja trocado pelos pais. Assim, considerando

os pais $p_1 = 100111$ e $p_2 = 011001$ e o ponto 3 para cruzamento, são gerados os filhos $f_1 = 100001$ e $f_2 = 011111$;

- Cruzamento com 2 pontos de corte: similar ao cruzamento com um ponto, as posições para corte e troca são escolhidas aleatoriamente, efetuando-se a troca do material genético entre estes pontos. Assim, para os pais p_1 e p_2 já descritos e para os pontos 2 e 4, são gerados os filhos $f_1 = 101011$ e $f_2 = 010101$;

- Cruzamento com n pontos: idéia semelhante às anteriores. Destaque para a variação do ponto de corte, conforme o número de pontos escolhidos. Se algum ponto de corte foi sorteado mais de uma vez, não se procura por outro;

- Cruzamento uniforme: seguindo cada *bit* do primeiro indivíduo, verifica-se se ocorreu um evento com probabilidade de 50%. Se positivo, ali será um ponto de corte, caso contrário, repete-se o procedimento para o próximo *bit*.

- Cruzamento por variável: o cruzamento terá um ponto de corte por variável. Desta maneira, sendo consideradas 3 variáveis tem-se 3 pontos de corte;

- Cruzamento entre vários indivíduos: seleciona-se o indivíduo base e depois, para cada variável, escolhe-se aleatoriamente um parceiro e um ponto de corte.

2.2.4.2 Mutação

O operador genético de mutação introduz inovações no material genético da população e isto ocorre de forma aleatória (Bäck, 1994). Em algoritmos genéticos existem diversas formas de implementar o operador mutação. Por exemplo, se o indivíduo consiste de genes binários, a mutação consiste em inverter o valor de um bit; se o indivíduo consiste de genes que representam condições de regras (atributo, operador e valor), a mutação pode alterar o operador, ou mesmo o valor (Goldberg, 1989).

A mutação pode introduzir material genético que não está presente em nenhum indivíduo da população, logo a mutação contribui para aumentar a

diversidade genética da população. A mutação impede que o AG tenha uma convergência rápida, evitando desta forma que o mesmo se estabeleça em regiões de mínimos locais.

Muitas vezes, em problemas de mineração existe uma taxa de mutação que controla a frequência com que as mutações acontecerão, e em geral esta taxa é menor que a taxa de cruzamento.

A figura 5 mostra um exemplo da operação de mutação.

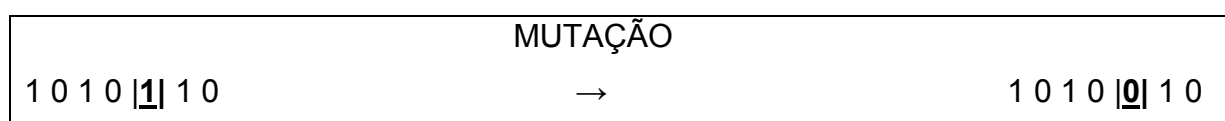


Figura 5 – Exemplo de mutação.

2.3 Otimização por Nuvem de Partículas

Nuvem ou enxame de partículas (*Particle Swarm Optimization*, PSO) é uma metodologia originalmente proposta por Kennedy e Eberhart (1995), esta metodologia é baseada em população de soluções, em que cada solução candidata (denominada partícula) possui associada a ela uma velocidade. Esta velocidade é ajustada de acordo com a experiência da partícula correspondente e de acordo com a experiência das outras partículas da população.

PSO é uma técnica heurística de busca, sendo uma abordagem baseada em uma população de indivíduo, porém ao contrário da maioria dos algoritmos evolutivos, a PSO não é motivada pela sobrevivência do indivíduo mais apto e sim motivada pelo comportamento social, ou seja na convivência do grupo de partículas e sua sinergia.

De acordo com Angeline (1998), pode-se fazer duas distinções principais entre o algoritmo PSO e um algoritmo evolucionário (AE):

1. Os algoritmos evolucionários (AEs) usam três mecanismos no seu processamento: representação do pai, seleção dos indivíduos e o ajuste fino de

seus parâmetros. O PSO confia somente em dois mecanismos no seu processamento, desde que PSO não adote uma função explícita da seleção. A ausência de um mecanismo da seleção em PSO é compensada pelo uso de líderes guiar a busca. Entretanto, não existe nenhuma noção da geração da prole em PSO como nos AEs.

2. Uma segunda diferença entre AEs e PSO é a maneira como os indivíduos são manipulados. PSO usa um operador que ajusta a velocidade de uma partícula em um sentido particular. Isto pode ser visto como um operador direcional de mutação em que o sentido é definido pela partícula melhor local e pela melhor global (do enxame ou núvem). Se o sentido da melhor local é similar ao sentido da melhor global, o ângulo de sentidos potenciais será pequeno, visto que um ângulo maior fornecerá uma escala maior da exploração. Ao contrário, os AEs um operador de mutação pode ajustar um indivíduo em todos os sentidos, embora as probabilidades relativas para cada sentido pode ser diferente.

A simplicidade relativa do PSO e o fato que é uma técnica baseada em população fizeram dele um candidato natural a ser estendido para otimização com múltiplos objetivos, este tema será abordado na sub-seção 2.3.1.

Para estabelecer uma terminologia comum, a seguir algumas definições referentes a PSO são apresentadas.

1. Nuvem ou enxame (*swarm*): População de soluções (partículas) do algoritmo.
2. Partícula: Membro (individual) da nuvem. Cada partícula representa uma solução potencial ao problema que esta sendo resolvido. A posição de uma partícula é determinada pelo solução que representa.
3. *pbest* (*personal best*): melhor posição pessoal de uma partícula dada, assim distante. Isto é, a posição da partícula que forneceu o maior sucesso (medido nos termos de um valor escalar análogo à aptidão adotada em AEs).
4. *lbest* (*local best*): Posição do melhor membro da partícula da vizinhança de uma dada partícula.
5. *gbest* (*global best*): Posição da melhor partícula de todo o enxame.

6. Líder: Partícula que é usada guiar uma outra partícula para regiões melhores do espaço de busca.
7. Velocidade: Este vetor dirige a otimização do processo, isto é, determina o sentido em que uma partícula necessita se movimentar, a fim melhorar sua posição atual.
8. Peso da inércia: Denotado por w , o peso da inércia é empregado para controlar o impacto da história precedente das velocidades na velocidade atual de uma partícula dada.
9. Fator de aprendizagem: Representa a atração que uma partícula tem para o seu próprio sucesso ou de seus vizinhos. Dois são os fatores usados: c_1 e c_2 . O fator c_1 é o fator de aprendizagem cognitivo e representa a atração que uma partícula tem para seu próprio sucesso. O c_2 é o fator de aprendizagem social e representa a atração que uma partícula tem para o sucesso de seus vizinhos. Ambos, c_1 e c_2 são definidos geralmente como constantes.
10. Topologias: dependendo de como os indivíduos da população estão ligados, existem diversas topologias de implementação, algumas delas são tratadas nas sub-seções 2.3.1 a 2.3.4.

A PSO é iniciada com uma população de soluções geradas aleatoriamente, usando distribuição uniforme, cada solução potencial (partícula) recebe uma velocidade aleatória. Estas partículas são movimentadas pelo espaço de busca do problema. As partículas conservam o conhecimento do melhor valor da função de aptidão (*fitness*) denotado por *pbest* (versão *local*). Um outro valor melhor é “seguido” pela versão *global*, *gbest*, do otimizador por nuvem de partícula e sua localização obtida de alguma partícula que compõe a população de partículas (Coelho e Mariani, 2005).

O conceito da PSO consiste de, a cada passo iterativo, mudar a velocidade (acelerando) de cada partícula em direção às localizações do *pbest* e do *gbest*. A aceleração desta busca é ponderada através de um termo gerado de forma aleatória vinculando este de forma separada as localizações do *pbest* e do *gbest*.

Segundo Bonissone *et al.* (1999) o procedimento para implementação da PSO é regido pelas seguintes etapas :

i) Iniciar uma população (matriz) de partículas, com posições e velocidades em um espaço de problema n dimensional, de forma aleatória com distribuição uniforme;

ii) Para cada partícula, avaliar a função de aptidão;

iii) Comparar a avaliação da função de aptidão da partícula com o $pbest$ da partícula. Se o valor corrente é melhor que $pbest$, então o valor de $pbest$ passa a ser igual ao valor da função de aptidão da partícula, e a localização do $pbest$ passa a ser igual à localização atual no espaço n dimensional;

iv) Comparar a avaliação da função de aptidão com o prévio melhor valor de aptidão da população. Se o valor atual é melhor que o $gbest$, atualizar o valor de $gbest$ para o índice e valor da partícula atual;

v) Modificar a velocidade e a posição da partícula de acordo com as equações (4) e (5), respectivamente (Shi *et al.*, 1998; Shi *et al.*, 1999):

$$v_{id}^{(K+1)} = wv_{id}^{(K)} + c_1r_1 \times (p_{id}^{(k)} - x_{id}^{(K)}) + c_2r_2 \times (g_{id}^{(K)} - x_{id}^{(K)}) \quad (4)$$

onde V_i^k é a velocidade da partícula i na iteração k , V_i^{k+1} é a velocidade da partícula i na iteração $k+1$, r_1 e r_2 são números aleatórios entre 0 e 1, X_i^k é a posição atual da partícula i na iteração k , p_{id} é o $pbest$ da partícula i , p_{gd} é o $gbest$ do grupo, w é a função do peso para a velocidade da partícula i , c_1 e c_2 são os coeficientes do peso para cada termo, k é o número da geração atual. A posição atual pode ser modificada pela seguinte equação:

$$x_{id}^{(K+1)} = x_{id}^{(K)} + v_{id}^{(K+1)} \cdot \Delta t \quad (5)$$

onde $\Delta t = 1$.

As velocidades das partículas em cada dimensão são validadas por uma velocidade máxima ($V_{m\acute{a}x}$). Se a soma das acelerações fizer com que a velocidade nesta dimensão exceda a $V_{m\acute{a}x}$, que é um parâmetro especificado pelo

usuário, a velocidade nesta dimensão é limitada a $V_{máx}$. $V_{máx}$ influencia o desempenho do PSO. Um $V_{máx}$ maior facilita a exploração global, enquanto um $V_{máx}$ menor incentiva a exploração local (Parsopoulos e Vrahatis, 2002a).

As constantes c_1 e c_2 representam os pesos da aceleração, puxando cada partícula para as posições mais $pbest$ e mais $gbest$. Ao valores baixos permitem que as partículas vaguem longe das regiões do alvo, já os valores elevados resultam no movimento abrupto para a região do alvo.

vi) Ir para a etapa (ii) até que um critério de parada seja encontrado, usualmente uma função de aptidão suficientemente precisa ou um número máximo de iterações (gerações).

As notações usadas neste trabalho são:

$x_i = [x_{i1}, x_{i2}, \dots, x_{in}]^T$ armazena a posição da i -ésima partícula do algoritmo de PSO ;

$v_i = [v_{i1}, v_{i2}, \dots, v_{in}]^T$ armazena a velocidade da i -ésima partícula da PSO, e o vetor ;

$p_i = [p_{i1}, p_{i2}, \dots, p_{in}]^T$ representa a posição do melhor valor de aptidão da i -ésima partícula.

O índice g representa o índice da melhor partícula entre todas as partículas do grupo. A variável w é a ponderação de inércia, c_1 e c_2 são constantes positivas; r_1 e r_2 são duas funções para geração de números aleatórios com distribuição uniforme no intervalo $[0, 1]$, respectivamente.

As velocidades das partículas em cada dimensão são limitadas a um valor máximo da velocidade, $V_{máx}$. O $V_{máx}$ é um parâmetro importante, pois determina a precisão com que as regiões próximas as soluções atuais serão exploradas. Se $V_{máx}$ é muito alto, a PSO facilita a busca global, enquanto um valor $V_{máx}$ pequeno enfatiza as buscas locais.

A primeira parte na equação (4) é um termo de momento da partícula. A ponderação de inércia w representa o grau de momento da partícula. A segunda parte consiste da parte “cognitiva”, que representa o “conhecimento” da partícula independentemente. A terceira parte é a parte “social”, que representa a colaboração entre as partículas. As constantes c_1 e c_2 representam as contribuições

“cognição” e “social” que influenciam cada partícula em direção as posições *pbest* e *gbest*, respectivamente. Estes parâmetros são usualmente ajustados por heurísticas de tentativa e erro (Eberhart e Shi, 2001).

Um exemplo de pseudocódigo para um PSO pode ser como segue:

```

Início
  Iniciar a população
  Localizar gbest
   $t = 0$  //geração
  Enquanto  $t < tmax$ 
    Para toda partícula
      Atualizar posição (vôo)
      Atualização da velocidade
      Atualizar pbest
    Fim Para
      Atualizar gbest
       $t = t + 1$ 
  Fim Enquanto
Fim

```

Figura 6 – Pseudo-código básico de um PSO.

Segundo Reyes-Sierra e Coello (2006a), dois são os aspectos que tornaram o PSO tão popular:

1. O algoritmo principal de PSO é relativamente simples (desde que em sua versão original, adota somente um operador para criar soluções novas, ao contrário de a maioria dos AEs) e sua execução é, conseqüentemente, direta. Adicionalmente, há uma abundância de código de fonte de PSO disponível em domínio público.
2. PSO pode ser eficaz em uma grande variedade de aplicações, podendo produzir resultados promissores com um custo computacional baixo.

2.3.1 Topologia Estrela (*gbest*)

Conhecida também como *gbest*, é uma relação totalmente conectada, todos os indivíduos estão conectados entre si, ou seja, cada partícula possui todas as outras como vizinhos, para Guo, Yu e Xu (2006), isto implica que a melhor partícula/posição global para todas as partículas é idêntica. Segundo Novaes (2002), esta implementação conduz a uma convergência mais rápida porém os riscos de uma convergência para um máximo local é bem maior que em outras implementações. A figura 7 mostra um exemplo desta topologia.

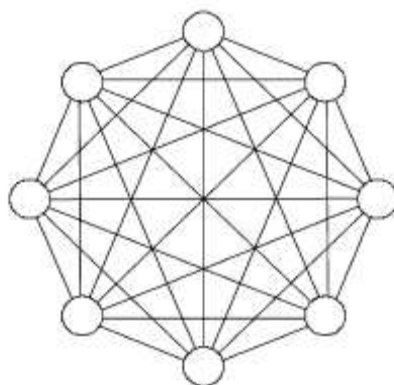


Figura 7 – Topologia estrela.

2.3.2 Topologia em Roda

Na topologia em roda um indivíduo é conectado a todos os demais, que por sua vez são completamente isolados entre si. O indivíduo conectado a todos os demais é denominado indivíduo central. O desempenho do melhor indivíduo é sempre passado ao indivíduo central que por sua vez passa a influenciar todos os demais. Todos os indivíduos se influenciam mutuamente como na topologia em estrela, porém de forma mais lenta, através de um filtro formado pelo indivíduo central. Esta topologia reduz um pouco a possibilidade de convergência em um

máximo local. O número de conexões nesta topologia é menor se comparada com a topologia em círculo (que será apresentada na sub-seção 2.3.3), porém a informação fluirá mais rapidamente uma vez que não serão realizados dois passos para avaliar a melhor posição global nova. A figura 8 mostra um exemplo desta topologia.

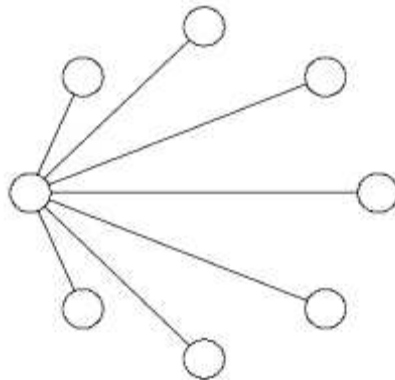


Figura 8 – Topologia em roda.

2.3.3 Topologia em Círculo (*Ibest*)

Cada indivíduo é conectado apenas a dois vizinhos. A influência entre indivíduos distantes é pequena, pois precisa forçosamente passar por toda a cadeia de indivíduos entre eles. Essa topologia é mais eficaz em não convergir em máximos locais pois indivíduos distantes daquele que encontrar um máximo local não serão imediatamente influenciados por este máximo local e continuarão sua busca em outras regiões do espaço de busca. A topologia em círculo corresponde a um anel, o fluxo de informação é reduzido acentuadamente se comparada a topologia estrela. A figura 9 mostra um exemplo desta topologia.

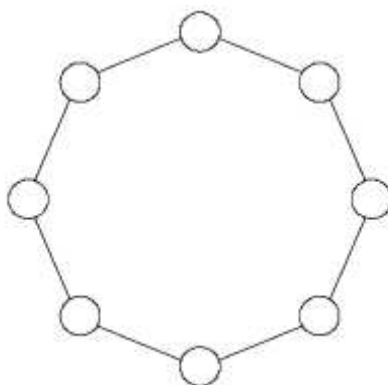


Figura 9 – Topologia em círculo.

2.3.4 Topologia Aleatória

Nesta topologia os grupos de indivíduos são conectados a outros grupos de forma aleatória. A figura 10 mostra um exemplo desta topologia.

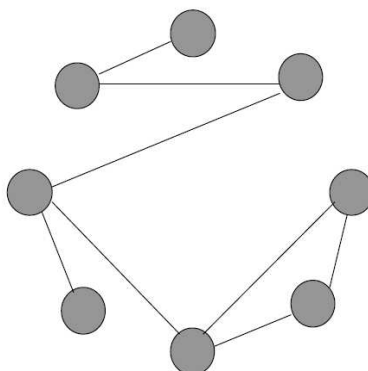


Figura 10 – Topologia aleatória.

2.4 Algoritmos Evolutivos Aplicados à Mineração de Dados

Aqui serão citados alguns trabalhos de sucesso que fizeram uso de algoritmos evolutivos e também PSO na mineração de dados.

Nos trabalhos publicados por Sousa, Silva e Neves (2003, 2004, 2006) foi proposto o uso de PSO na descoberta de regras de classificação. As

regras de classificação foram geradas por três variações do PSO, que são as seguintes : *Discrete Particle Swarm Optimiser* (DPSO), *Linear Decreasing Weight Particle Swarm Optimiser* (LDWPSO) e *Constricted Particle Swarm Optimiser* (CPSO), além de um outro algoritmo evolutivo (um algoritmo genético) e o J48, este uma execução Java do C4.5 (Quilan, 1993). Os resultados, a respeito do desempenho da exatidão e da velocidade, foram comparados empiricamente entre as três variações do PSO, do AG e do J48. Os resultados obtidos indicaram que o PSO é competitivo com o AG e podem ser aplicados com sucesso em outros domínios.

Tan, Yu e Ang (2006) apresentam um método para extrair múltiplas listas de regras da decisão na mineração dos dados, visando satisfazer aos critérios da classificação da exatidão e da elevar a compreensão do usuário. O algoritmo incorpora o conceito do dominância de Pareto. O desempenho do DOEA (*Dual-Objective Evolutionary Algorithm*) foi testado em oito bases de dados obtidas no repositório de aprendizagem de máquina da UCI (Murphy e Aha, 1994). Os resultados de classificação foram analisados qualitativamente e estatisticamente, e mostraram que o DOEA (*Dual-Objective Evolutionary Algorithm*) é competitivo em comparação aos classificadores como o C4.5 (Quilan, 1993), o PART (Frank *et al.*, 1998) e o DT (Kohavi, 1995), produzindo regras compreensíveis com exatidão.

Santos e Naval (2006) utilizaram otimização multiobjetivo para a evolução dos classificadores da árvore de decisão. Foi empregado otimização com nuvem de partículas multiobjetivo para evoluir uma população de árvores de decisão com dois objetivos: a exatidão da classificação e a complexidade do classificador baseada no princípio mínimo do comprimento da descrição. O algoritmo de otimização multiobjetivo gerou árvores de decisão com poucos nós e com taxas de erro menores do que os produzidos pelo algoritmo C4.5 (Quilan, 1993). As funções objetivas usadas impediram sobre-treinamento das árvores de decisão no conjunto de treinamento. A validade e o desempenho desta aproximação foram avaliados em diversas séries de dados comumente utilizados. Assim, provou-se o potencial de usar um algoritmo evolucionário visando a otimização de Pareto, pois este produziu modelos promissores e simples de classificação para de indução da árvores de decisão.

Pappa (2002) objetivou observar o comportamento de versões multiobjetivo de métodos que estão consolidados na literatura para a tarefa de seleção de atributos. O algoritmo C4.5 (Quilan, 1993) foi utilizado como base de comparação para as soluções encontradas. O algoritmo genético multiobjetivo de seleção de atributos proposto encontrou soluções melhores que a padrão (consistindo de todos os atributos) em 16 das 18 bases da UCI (Murphy e Aha, 1994) utilizadas nos experimentos, resultados muito superiores aos encontrados pelo AG simples, que só conseguiu soluções com taxas de erro e tamanhos de árvore menores que a solução padrão em uma das bases de dados testadas. O algoritmo genético multiobjetivo proposto foi também aplicado a uma base de dados de produtos Motorola, composta por uma média de 15.000 registros distribuídos em 2 classes e descritos por 109 atributos. O método selecionou 27 desses 109 atributos, com taxas de erro e tamanhos de árvore obtidos menores que o da solução padrão (solução com todos os atributos) executada com o C4.5 (Quilan, 1993). Com base nos resultados, Pappa (2002) validou e confirmou que métodos de busca, no caso da tarefa de seleção de atributos, podem ser utilizados com conceitos multiobjetivos com eficácia e eficiência.

Segundo Freitas (2002), a principal motivação para a utilização de AGs na descoberta de regras com um alto nível de predição é que executam uma busca global e também lidam melhor com a interação do atributo do que os algoritmos de indução de regras usados frequentemente na mineração dos dados. Um dos problemas chaves para preparar uma série de dados para a mineração é o problema da seleção de atributo. Este problema consiste em selecionar, entre todos os atributos disponíveis, um subconjunto dos atributos relevantes para predizer o valor do atributo do objetivo. O uso de um AGs para a seleção do atributo parece natural. A razão principal é a dificuldade na seleção do atributo vinculada a interação do atributo, e uma das potencialidades dos AGs é que lidam geralmente bem com as interações do atributo.

Jourdan, Dhaenens e Talb (2001) utilizaram um AG para descobrir as características genéticas e os fatores ambientais que estavam envolvidos com doenças como obesidade e diabetes. Para isto foi utilizada uma base de dados fornecida pelo Instituto Biológico Lille. Foi escolhido o AG porque o objetivo não era

considerar apenas uma característica e sim uma quantidade de características grande. O primeiro objetivo foi isolar as associações mais relevantes das características, e a classificação do indivíduo que possui a doença foi feita de acordo com estas associações. O desempenho do método foi testado com relação ao tamanho do problema que ele pode tratar, os autores afirmaram que o tempo de execução cresce linearmente com o número de características, e também concluíram que o método pode lidar com problemas de grande porte. O AG foi executado diversas vezes e ele foi capaz de selecionar características interessantes e classificar pares de indivíduos de acordo com estas características e confirmar associações interessantes destas características. Os resultados foram relevantes, e o AG pareceu ser robusto e mostrou associações interessantes.

Khabzaoui, Dhaenens e Talbi (2005) utilizaram regras de associação para descobrir e descrever associações em grandes bases de dados, ou seja, descobrir conhecimento interessante. O problema foi modelado como um problema combinatorial de otimização multiobjetivo. Foi proposto resolver este problema utilizando um AG, e para isto foram criados mecanismos específicos de mutação, operadores de cruzamento e elitismo. Foi desenvolvido uma abordagem evolucionária cooperativa baseada em AG, e este modelo desenvolvido foi testado diversas vezes afim de encontrar a melhor configuração dos parâmetros. Este modelo também foi comparado a aproximações não cooperativas, e os melhores resultados obtidos foram encontrados pelo modelo de AG cooperativo.

Dehuri, Jagadey, Ghosh e Mall (2006) apresentam uma técnica para extração de regras de associação e para isto foi utilizado um AG multiobjetivo. As funções objetivo foram construídas com base no fator de confiança, compreensibilidade e interesse das regras de associação. Foi adotada esta técnica devido ao grande volume de dados contidos na base de dados utilizada. Os resultados da técnica empregada foi um conjunto de soluções não dominantes, como o volume de dados era muito, para superar esta dificuldade optou-se por uma técnica rápida de escalabilidade, usando processamento paralelo e uma rede homogênea dedicada, assim todos os dados que estavam participando da mineração e a população de indivíduos foram distribuídos através de todos os

processadores disponíveis. O resultado experimental mostrou que o algoritmo teve bom desempenho mesmo com o alto volume de dados e velocidade considerável.

Minaei-Bidgoli e Punch (2003) apresentam uma aproximação para classificar estudantes e prever sua classe final baseado nas características extraídas dos dados registrados em um sistema educacional web. A pesquisa feita pelos autores foi executada sobre uma parte do sistema educacional web, assim foram obtidos dois tipos de séries de dados: um deles referentes aos recursos educacionais (*web pages*, demonstrações, simulações, entre outros) e outro referente às informações dos usuários que cria, modifica, avalia e usa os recursos. Neste estudo os autores se propuseram responder algumas perguntas: pode-se encontrar classes de estudantes? Ou seja existe algum grupo de estudantes que utilize este sistema educacional web de maneira semelhante? Se sim, como pode-se fazer este grupo aproveitar melhor os recursos do sistema? Pode-se classificar as funcionalidades utilizadas pelos estudantes? Se sim pode-se auxiliar os instrutores a desenvolver com maior eficácia as atividades disponibilizadas no sistema?

Foram utilizados quatro classificadores para segregar os estudantes, esta combinação de classificadores conduziu a uma melhoria significativa da exatidão da classificação. A utilização do AG minimizou a taxa de erro e melhorou a precisão de pelo menos 10% da predição de todos os exemplos de duas, três e de nove classes.

3 OTIMIZAÇÃO DE MÚLTIPLOS OBJETIVOS

A tomada de decisão é algo que pode nos confundir, pois a todo momento toma-se decisões. Estas decisões são tomadas mediante o uso do sentido comum ou intuição, porém existem situações em que esta tomada de decisão deve ser eficiente e rápida, não podendo acontecer enganos. Na tomada de decisão pode-se deparar com um problema de otimização e que normalmente estão envolvidos dois ou mais objetivos, e normalmente estes objetivos são conflitantes entre si. Em muitas aplicações são realizadas simplificações no problema original a fim de combinar todos os objetivos em uma única função, ou escolhendo apenas os objetivos classificados como prioritários ou importantes.

Esse tipo de problema é conhecido como otimização de múltiplos objetivos e possui uma característica particular, pois não apresenta uma solução única e sim um conjunto de soluções válidas, em que cada uma destas soluções pode ser considerada uma solução para o problema em questão, ou seja, existe uma investigação que tem como objetivo encontrar um vetor de variáveis de decisão, que satisfaça as restrições e otimize um vetor em que os elementos representam as funções objetivo. Estas funções formam uma descrição matemática dos critérios de desempenho, que normalmente estão em conflito. As soluções encontradas são ótimas por não existem outras soluções no espaço de busca melhores do que elas quando todos os objetivos são simultaneamente considerados, sendo conhecidas como soluções ótimas de Pareto.

Nos últimos anos vêm crescendo o número de pesquisas na área da otimização multiobjetivos, a motivação para este aumento vem do cotidiano, pois a maioria dos problemas a serem resolvidos possuem multiobjetivos. Um exemplo de aplicação multiobjetivos é a compra de um imóvel, quer-se comprar um imóvel com o menor custo, com a maior área útil e com localização privilegiada. Que imóveis atenderiam a estes três objetivos? Observa-se que estes objetivos são conflitantes, já que para menor custo, menor será a área útil do imóvel e pior será a sua localização.

3.1.1 Conceitos Básicos sobre Otimização de Múltiplos Objetivos

Um problema de otimização de múltiplos objetivos (MOP) consiste basicamente na obtenção de um conjunto de soluções que satisfaça as condições e minimize ou maximize uma função constituída por vários objetivos. Desta forma pode-se definir um problema de otimização com múltiplos objetivos da seguinte forma:

Minimizar ou maximizar $z = (f_1(x), f_2(x), \dots, f_r(x))$;

sujeito a $x \in X^*$, em que:

$g(x)$: restrições;

X : espaço de decisões;

$x = (x_1, x_2, \dots, x_n) \in X$, onde x é o vetor de decisão;

Z é a imagem de X ou espaço objetivo;

$z = (z_1, z_2, \dots, z_r) \in Z$, onde Z é o vetor objetivo;

$X^* = \{x \in X : g(x) \leq b\}$, sendo X^* o conjunto de soluções factíveis;

$b \in \mathfrak{R}^p$.

Diz-se que $x^* \in X^*$ é uma solução ótima de Pareto se não existir outra solução $x \in X^*$ tal que x domine x^* . Neste caso, $z^* = f(x^*)$ é denominado ponto Pareto-ótimo. O conjunto de todas as soluções eficientes é denominado conjunto Pareto-ótimo. A imagem em Z^* do conjunto Pareto-ótimo é denominada Fronteira Pareto-ótimo.

O conjunto de soluções consiste em todos os vetores de decisão que não podem melhorar o valor de qualquer objetivo, sem a degradação de pelo menos algum dos outros. Estes vetores são os conhecidos Pareto-ótimos. Sem perda de generalidade, pode-se descrever as soluções ótimas de Pareto formalmente para um problema de minimização puro, isto é, onde todas as funções objetivas devem ser minimizadas, pelas seguintes definições:

Definição 1 – Otimização de Múltiplos Objetivos Geral: Encontrar um vetor $\vec{x}^* = [x_1^*, x_2^*, \dots, x_n^*]^T$ que satisfaça as m restrições de desigualdade:

$$g_i(\vec{x}) \geq 0 \quad i = 1, 2, \dots, m .$$

E as p restrições de igualdade:

$$h_i(\vec{x}) = 0 \quad i = 1, 2, \dots, p .$$

E otimize o vetor função

$$\vec{f}(\vec{x}) = [f_1(\vec{x}), f_2(\vec{x}), \dots, f_k(\vec{x})]^T \text{ em que } \vec{x} = [x_1, x_2, \dots, x_n]^T \text{ é o vetor das}$$

variáveis de decisão.

Definição 2 - Ótimos de Pareto: Um ponto $\vec{x}^* \in \Omega$ (Ω é a região factível) é um ponto Pareto-ótimo se para cada $\vec{x} \in \Omega$ $y \in I = \{1, 2, \dots, k\}$ tanto,

$$\forall_{i \in I} (f_i(\vec{x}) = f_i(\vec{x}^*)) .$$

Ou, existe ao menos uma $i \in I$ tal que

$$f_i(\vec{x}) > f_i(\vec{x}^*) .$$

Esta definição diz que \vec{x}^* é um Pareto-ótimo se não existe um vetor \vec{x} factível que vá decrementar algum critério sem causar um aumento simultâneo em pelo menos algum outro critério. O significado de Pareto-ótimo é considerado respectivo a todo o espaço das variáveis de decisão a menos que seja especificado o contrário, ou seja, um ponto A é uma solução Pareto-ótimo e pertence à fronteira de Pareto se: não existe nenhum outro ponto B, tal que B domine A. Em geral, para cada região viável existe um subconjunto S de soluções não dominadas.

Um conjunto de soluções não dominadas é definido no contexto de uma amostra do espaço de busca, enquanto que o conjunto de Pareto-ótimo é definido em relação a todo o espaço de busca, esta é a diferença entre um conjunto de soluções não dominadas e um conjunto de Pareto-ótimo.

Definição 3 – Dominância de Pareto: Um vetor $\vec{u} = (u_1, \dots, u_k)$ diz-se que domina outro vetor $\vec{v} = (v_1, \dots, v_k)$ se e somente se u é parcialmente menor que v , por exemplo, $\forall i \in \{1, \dots, k\}, u_i \leq v_i \wedge \exists i \in \{1, \dots, k\} : u_i < v_i$.

Definição 4 – Conjunto de Ótimos de Pareto: Dado um MOP $\vec{f}(x)$, o conjunto de Pareto-ótimos (P^*) é definido como:

$$P^* := \{x \in \Omega \mid \neg \exists x' \in \Omega \vec{f}(x') \prec \vec{f}(x)\}.$$

Definição 5 – Fronteira de Pareto: Para um MOP $\vec{f}(x)$ e um conjunto de Pareto-ótimos P^* dado, a Fronteira de Pareto (PF^*) é definida como:

$$PF^* := \{\vec{u} = \vec{f} = (f_1(x), \dots, f_k(x)) \mid x \in P^*\}.$$

Esta definição caminha na direção dos limites da região para o qual o problema foi proposto, ou seja, nos pontos inferiores no plano das funções objetivo. Na Figura 11 a linha em vermelho é utilizada para delimitar os limites de um problema, a região de pontos definida por esta linha é denominada Fronteira de Pareto. A linha em azul representam as soluções dominadas e f_1 e f_2 são as funções objetivo do problema.

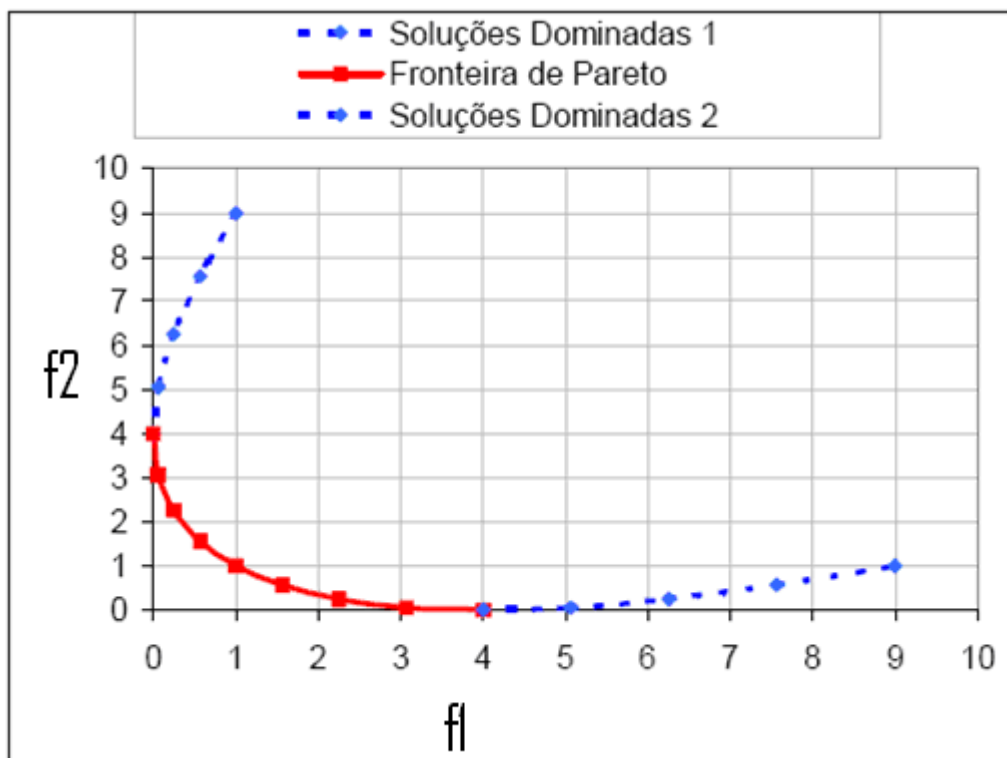


Figura 11 – Fronteira de Pareto (Castro, 2001).

3.1.2 Otimização de Múltiplos Objetivos Usando Nuvem de Partículas

Segundo Kennedy e Eberhart (2001), a otimização com nuvem de partículas é adaptável para otimização com múltiplos objetivos devido principalmente a alta velocidade de convergência que o algoritmo apresenta para problemas de otimização com um único objetivo.

Para aplicar a estratégia de PSO para a resolução de problemas com múltiplos objetivos são necessárias algumas modificações. Como foi apresentado no início do Capítulo 3, o conjunto de soluções de um problema com múltiplos objetivos não consiste em uma única solução como em um problema de otimização global. A otimização com múltiplos objetivos busca encontrar um conjunto de soluções (conjunto de Pareto-ótimo).

Geralmente, na resolução de um problema com múltiplos objetivos, três são os objetivos a serem atingidos:

1. Maximizar o número dos elementos do conjunto de Pareto-ótimo encontrado.
2. Minimizar a distância da Fronteira de Pareto produzida pelo algoritmo em relação a verdadeira Fronteira de Pareto (global), assumindo que sua localização é conhecida.
3. Maximizar a propagação das soluções encontradas, de modo a obter uma distribuição dos vetores mais uniforme possível.

Assim, como em outros algoritmos evolucionários, as três principais questões a serem consideradas quando PSO for estendido a otimização com múltiplos objetivos:

1. Como selecionar partículas (para serem usadas como líderes) a fim dar a preferência para soluções não dominantes em relação a soluções dominantes?
2. Como reter as soluções não dominantes encontradas durante o processo da busca a fim relatar as soluções que são não dominantes com respeito a todas as populações passadas e não somente com respeito ao atual? Também, é desejável que estas soluções sejam propagadas ao longo da Fronteira de Pareto.
3. Como manter a diversidade da população de partículas (*swarm*) a fim evitar a convergência a uma única solução?

Em problemas de otimização de múltiplos objetivos, cada partícula pode ter um conjunto de líderes diferentes e apenas um pode ser selecionado a fim atualizar sua posição. Tal conjunto de líderes é armazenado geralmente em um lugar diferente da população, como, por exemplo, em um arquivo externo.

Este arquivo externo é um repositório em que as soluções não dominantes encontradas são armazenados. As soluções contidas no arquivo externo são usadas como líderes quando as posições das partículas da população tiverem que ser atualizadas. A Figura 11 mostra a maneira em que um algoritmo MOPSO em geral funciona. Primeiramente, a população do PSO é iniciada. Então, um conjunto de líderes é inicializado também com partículas não dominantes da população. Como mencionado anteriormente, o conjunto de líderes é armazenado em um arquivo externo. Mais tarde, uma medida da qualidade é calculada para todos os líderes a fim selecionar (geralmente) um líder para cada partícula da população. Em

cada geração, para cada partícula, um líder é selecionado e o voo é executado. A maioria dos MOPSOs existentes aplicam algum tipo de operador de mutação após ter executado o voo.

Um exemplo de pseudocódigo para um MOPSO pode ser conforme apresentado na Figura 11.

```

Início
  Iniciar a população de partículas
  Iniciar os líderes em um arquivo externo
  Qualidade (líderes)
   $t = 0$ 
  Enquanto  $t < tmax$ 
    Para cada partícula
      Selecionar líder
      Atualizar Posição (voo e velocidade)
      Atualizar pbest (personal best)
    Fim Para
    Atualizar os líderes no arquivo externo
    Qualidade (líderes)
     $t = t + 1$ 
  Fim Enquanto
  Reportar resultados no arquivo externo
Fim

```

Figura 11 – Pseudo-código de um MOPSO.

A partícula é avaliada e seu correspondente *pbest* é atualizado. Após todas as partículas serem atualizadas, o conjunto de líderes é atualizado. Finalmente, a medida da qualidade do conjunto de líderes é recalculado. Este processo é repetido geralmente para um certo número fixo de iterações.

Algumas questões que são levantadas ao se tratar problemas com múltiplos objetivos estão relacionadas com dois aspectos principais do algoritmo:

1. Selecionar e atualizar dos líderes:

Como selecionar um único líder fora do conjunto de soluções não dominantes quando todas são igualmente boas? Selecionar este líder de uma maneira aleatória ou usar um critério adicional (para promover diversidade, por exemplo)?

Como selecionar as partículas que devem permanecer dentro o arquivo externo de uma iteração a outra?

2. Criação de novas soluções:

Como promover a diversidade através de dois mecanismos para criar soluções novas: atualização de posições e operador de mutação.

Existem diversas variações de MOPSO, Reyes e Coello (2006a) tratam de diversas destas variações, eles classificam estas variações em:

1. Agregação de aproximações: nesta categoria estão inseridas as aproximações que agregam todos os objetivos do problema em um único objetivo. Ou seja, o problema multiobjetivo é transformado em um problema de único-objetivo.

2. Aproximação baseada em sub-população: estas aproximações envolvem a utilização de várias sub-populações como otimizadores com objetivo único. Assim, as sub-populações trocam de alguma maneira a informação ou recombinação entre si que produzem um compromisso entre as soluções diferentes geradas previamente para os objetivos que otimizam separadamente.

3. Aproximação baseada em Pareto: estas aproximações usam as técnicas da seleção de líder baseadas na dominância de Pareto. A idéia básica de todas as aproximações consideradas é selecionar como partículas líderes que são não dominantes com relação a população.

Santana-Quintero *et al.* (2006) apresentam um algoritmo evolucionário multiobjetivo que consiste em uma aproximação de otimização de núvem de partículas e alguns conceitos da teoria de jogos. A idéia principal da aproximação foi combinar a taxa de convergência elevada do algoritmo de otimização de núvem de partículas com uma aproximação local da busca baseada em jogos para obter soluções não dominadas, de modo a encontrar uma boa distribuição ao longo da Fronteira de Pareto. A aproximação proposta pode convergir em diversas funções de teste com 10 a 30 variáveis da decisão com somente 4.000 avaliações da função da aptidão. A aproximação proposta foi validada usando nove funções padrão de teste adotadas geralmente na literatura especializada e os resultados foram comparados com um algoritmo evolucionário multiobjetivo denominado NSGA-II (*Non-dominated Sorting Genetic Algorithm II*) (Deb *et al.*, 2002).

Baltar e Fontane (2006) apresentam uma aplicação de um algoritmo evolucionário de otimização para a análise multiobjetivo da retirada seletiva de um reservatório térmico estratificado. Um MOPSO é usado encontrar as soluções não dominantes de Pareto, visando minimizar as variações na qualidade da água, tais como: temperatura, oxigênio dissolvido, sólidos dissolvidos totais e potencial do hidrogênio, estes são os objetivos. O algoritmo de MOPSO é executado como um add-in para o MS-Excel e pode encontrar soluções não dominadas combinando os 4 objetivos mencionados. Foi desenvolvido um método gráfico interativo para indicar soluções não dominadas de tal maneira que as melhores soluções do conjunto possam ser identificadas pela importância relativa dada a cada um dos objetivos. O método permite que o responsável pelas decisões explore o Pareto ajustado e visualize não somente a melhor solução do conjunto, e também permite ajustar também as soluções que fornecem valores similares.

Branke e Mostaghim (2006) investigam a influência das melhores partículas pessoais (*pbest*) no MOPSO. Eles mostram que a seleção de uma guia pessoal apropriada tem um impacto significativo no desempenho do algoritmo. Assim, foi proposta uma abordagem que permita a cada partícula memorizar todas as *pbest* não dominadas encontradas. Isto significa que se a partícula *pbest* atualizada for indiferente a *pbest* antiga, mantêm-se ambas no arquivo externo. Também são propostas diversas estratégias para selecionar uma *pbest* do arquivo pessoal. Estes métodos são comparados para em alguns problemas da literatura.

Raquel e Naval (2005) apresentam uma aproximação de PSO que se estende para resolver problemas multiobjetivo de otimização incorporando um mecanismo de aglomerar a distância no algoritmo de PSO, especificamente na seleção da melhor solução global e no método para apagar o arquivo externo de soluções não dominadas. O mecanismo de aglomerar a distância junto com um operador de mutação mantêm a diversidade de soluções não dominadas no arquivo externo. O desempenho desta aproximação é avaliado em funções e em métricas de teste propostas na literatura. Os resultados mostram que a aproximação proposta é altamente competitiva na convergência para a Fronteira de Pareto e gera um conjunto distribuído de soluções não dominadas.

Li (2004) menciona que a estratégia de maximização e minimização teve sua origem na teoria de jogos, mas pode ser adotada para obter uma otimização multiobjetivo eficaz. Li (2004) propõe um MOPSO, denominado maximinPSO, que usa uma função de *fitness* derivada da estratégia do *maximin* para determinar a Dominância de Pareto. A função de *fitness* do *maximin* tem algumas propriedades desejáveis no que diz respeito à otimização multiobjetivo, uma vantagem é que nenhuma técnica adicional é necessária, o *fitness* do *maximin* de uma solução pode dizer se uma solução está dominada ou não. O artigo demonstrou que na série da função de teste, o maximinPSO produz uma convergência e uma propagação quase perfeitas das soluções ao longo da frente Pareto-ótima em todas as medidas de desempenho usadas e testadas.

Mostaghim e Teich (2004) propuseram um método usando a otimização multiobjetivo através de nuvem de partículas para descobrir a Fronteira de Pareto. O método trabalha em duas fases: na primeira fase o objetivo é obter uma aproximação boa da Fronteira de Pareto e na segunda fase são geradas subpopulações para cobrir a Fronteira de Pareto. O método foi avaliado usando funções diferentes de teste e comparado com um método existente, o MOEA – *Multi-objective Evolutionary Algorithm*).

Reyes-Sierra e Coello (2006b) propõem um mecanismo dinâmico para variar a probabilidade do *fitness* e isto é aplicado durante o funcionamento de um MOPSO, a fim obter uma redução no custo computacional (do que obtido com uma probabilidade fixa), sem afetar a qualidade dos resultados. Os resultados obtidos mostraram que é possível reduzir o custo computacional por 32% sem afetar a qualidade da Fronteira de Pareto obtida.

Coello, Pulido e Lechuga (2004) apresentam uma aproximação em que a Dominância de Pareto foi incorporada no PSO a fim permitir uma heurística que resolva problemas com diversas funções objetivo. O algoritmo proposto usa um repositório externo secundário de partículas que é usado mais tarde por outras partículas guiar seu próprio vôo. Foi incorporado também um operador especial de mutação que pode melhorar o potencial de exploração do algoritmo. Esta proposta foi validada usando diversas funções e métricas de teste retirados da literatura padrão. Os resultados indicaram que a aproximação é altamente competitiva e

que pode ser considerada uma alternativa viável para resolver problemas de otimização multiobjetivos.

Kukkonen e Deb (2006) propuseram um algoritmo para podar um conjunto de soluções não dominadas. O algoritmo é baseado no cálculo da distância usado dentro do elitismo da abordagem de algoritmo genético denominado NSGA-II (*Non-dominated Sorting Genetic Algorithm II*) (Deb *et al.*, 2002). Os resultados mostraram que para problemas bi-objetivos usados no teste, o algoritmo de poda proposto forneceu uma distribuição melhor se comparada ao algoritmo de poda original do NSGA-II. Entretanto, com problemas multiobjetivos testados não houve melhora.

Cagnina e Esquivel (2005) propõem um MOPSO que incorpora Dominância de Pareto, uma política de elitismo e duas técnicas para manter a diversidade: um operador de mutação e uma barreira que é usada como uma posição geográfica espaço excedente da função objetivo. Para validar aproximação proposta os autores utilizaram três funções de teste propostas na literatura especializada. Os resultados preliminares das simulações foram comparados com os resultados obtidos por dois outros algoritmos: o *Pareto Archived Evolution Strategy* (PAES) e o *Multi-Objective Genetic Algorithm 2* (MOGA2). Estes resultados mostraram a aproximação proposta é uma alternativa promissora para resolver problemas de otimização multiobjetivo.

Bartz-Beielstein, Mehnen, Naujoks, Schmitt e Zibold (2004) descrevem um ambiente computacional para o desenvolvimento, a análise e a aplicação de algoritmos evolucionários multiobjetivo chamado KEA (*Kit for Evolutionary Algorithms*). Este projeto oferece um ambiente apropriado para vários tipos de tarefas de otimização, fornecendo uma relação para avaliar as funções multiobjetivo usando uma variedade de algoritmos evolucionários multiobjetivos. Além disso, o KEA (*Kit for Evolutionary Algorithms*) contém diversos métodos avançados de comparação para avaliar medidas de desempenho dos algoritmos, sendo capaz de indicar o progresso da otimização em uma exposição dinâmica ou de indicar apenas os resultados de otimização em uma modalidade de estática de visualização.

Santos e Naval (2006) mencionam que a árvore de decisão é um modelo popular de classificação. Os dois objetivos principais na indução da árvore de decisão são previsões exatas para exemplos despercebidos e a compreensibilidade humana. Com base nestes conceitos, Santos e Naval (2006) utilizam a otimização multiobjetivo para a evolução dos classificadores de árvores de decisão. Algumas árvores mais simples de decisão são geralmente compreensíveis aos seres humanos à custa da precisão. Foi empregado um MOPSO usando o algoritmo MOPSO-CD para evoluir uma população das árvores de decisão com dois objetivos: exatidão e classificação baseada no princípio mínimo do comprimento da descrição. O desempenho desta aproximação foram avaliados em diversas séries de dados largamente utilizados e os resultados mostraram que a aproximação é certamente eficaz em induzir as árvores múltiplas de decisão que são exatas e simples.

Yusiong e Naval (2006) sugerem uma aproximação para o treinamento de uma rede neural com otimização simultânea das arquiteturas e dos pesos com PSO em abordagem multiobjetivo. A maioria de métodos de treinamento computacionais baseados algoritmos evolucionários formulam o problema em uma única função objetiva, tornando o modelo pesado. O objetivo dos autores foi determinar se o PSO multiobjetivo pode treinar redes neurais que envolvem dois objetivos: exatidão e complexidade. Foram propostas regras para o apagamento automático de nós desnecessários da rede baseada na seguinte idéia: uma conexão é podada se seu peso é menor do que o valor da polarização. Os experimentos foram executados nas séries de dados obtidas da mostra do repositório da aprendizagem de máquina da UCI (Murphy e Aha, 1994). Estes experimentos mostraram que aproximação fornece uma eficácia significativa para o treinamento de redes neurais e que é competitivo com outros métodos da computação evolutiva.

Mostaghim e Teich (2003a) apresentam um estudo da influência do *e-dominância* em métodos MOPSO. O papel o mais importante do *e-dominância* é limitar o número das soluções não dominadas armazenadas no arquivo, que tem influências na época, na convergência e na diversidade computacionais das soluções. O *e-dominância* foi comparado com a técnica aglomeração existente para reparar o tamanho do arquivo e as soluções são comparadas em termos de tempo

computacional, convergência e diversidade. Para diversidade uma nova métrica foi sugerida neste artigo. Os resultados mostram que as soluções do não dominadas do método do *e-dominância* possuem em alguns casos melhor convergência e também diversidade.

Baumgartner, Magele, Preis e Renhart (2004) propõem uma alternativa para detectar a Fronteira de Pareto. O algoritmo proposto mostrou um desempenho promissor e forneceu também uma representação confiável para a Frente de Pareto-ótima. Os autores modificam o PSO para detectar fronteira de Pareto.

Bartz-Beielstein *et al.* (2003) propõem uma técnica nova de otimização através de nuvem de partículas, denominada DOPS. O DOPS integra técnicas arquivamento bem conhecidas dos algoritmos evolucionários de otimização através de nuvem de partículas. Modificações e extensões das técnicas arquivamento são empiricamente analisados e diversas funções de teste foram utilizadas para ilustrar a usabilidade da aproximação proposta. Foi apresentada uma análise estatística dos resultados obtidos, e estes resultados foram considerados bons.

Mostaghim e Teich (2003b) introduzem um método novo chamado de Sigma, este método novo foi usado para encontrar os melhores guias locais para cada partícula da população. O método Sigma foi executado e comparado com um outro método, que usa a estratégia de um método existente de MOPSO para encontrar as guias locais. Estes métodos foram testados para funções diferentes e o resultado foi comparado com os resultados de um algoritmo evolucionário multiobjetivo (MOEA – *Multi-objective Evolutionary Algorithm*).

Fieldsend e Singh (2002) utilizaram uma ordenação dos membros do conjunto de soluções não dominadas pela estrutura de dados para facilitar a escolha de um melhor indivíduo global para cada membro do enxame, a de fim dirigir suas velocidades. O método foi validado usando diversas funções de teste, e foi significativamente melhor que dois algoritmos multiobjetivo (MOAS – *Multi-objective Algorithms*) existentes.

Coello e Lechuga (2002) introduziram uma proposta que estendeu o PSO para problemas de otimização com múltiplos objetivos. Esta proposta usou o

conceito de dominância de Pareto para determinar o sentido do vôo de uma partícula e mantêm previamente os vetores não dominados em um repositório global que foi usado mais tarde por outras partículas para guiar o vôo. A aproximação foi validada usando diversas funções padrão de teste da literatura especializada. Os resultados obtidos indicaram que aproximação é competitiva com técnicas evolucionárias de otimização com múltiplos objetivos atuais: o o *Pareto Archived Evolution Strategy* (PAES) e o NSGA-II (*Non-dominated Sorting Genetic Algorithm II*).

Hu e Eberhart (2002) apresentam um algoritmo de PSO para otimização multiobjetivo. PSO foi modificado para usar uma estratégia dinâmica de vizinhança, atualizá-la na memória da partícula, e pela otimização da dimensão tratar os objetivos múltiplos. Diversos casos foram testados e mostraram que PSO pode eficientemente encontrar soluções de Pareto-ótimas.

Parsopolous e Vrahatis (2002b) constituíram um estudo do PSO em problemas de otimização multiobjetivo. A habilidade de PSO de detectar pontos de Pareto e a forma da captação da Frente de Pareto foi estudada com experiências. Os aspectos críticos da aproximação de VEGA (*Vector Evaluated Genetic Algorithm*) (Schaffer, 1985) para otimização multiobjetivo que usa algoritmos genéticos foram adaptados à estrutura de PSO para que possa lidar eficazmente com o problema multiobjetivo. As conclusões foram derivadas e foram propostas idéias para uma pesquisa adicional.

Zhang *et al.* (2003) propõem um algoritmo para resolver problemas de otimização multiobjetivo baseado em PSO com uma melhoria na maneira de selecionar a melhor partícula global e individual. A busca para o conjunto de Pareto-ótimo de problemas multiobjetivo de otimização foi executada e as simulações numéricas mostraram a eficácia do algoritmo proposto.

Pulido, Coello e Santana-Quintero (2007) apresentam em seu trabalho um otimizador multiobjetivo eficiente que utiliza nuvem de partículas (EMOPSO), que é uma versão melhorada de um algoritmo evolucionário multiobjetivo (MOEA – *Multi-objective Evolutionary Algorithm*) proposto anteriormente pelos autores. No artigo são apresentados: o mecanismo para manter um conjunto bem distribuído de soluções não dominadas, operador que evita a

convergência prematura, o esquema de manipulação, e o estudo dos parâmetros que conduziu os autores a proporem um mecanismo de adaptação. O algoritmo final pode produzir aproximações razoavelmente boas da Frente de Pareto de problemas com até 30 variáveis de decisão, executando somente 2000 avaliações da função da aptidão. Os resultados foram comparados com os resultados do NSGA-II (*Non-dominated Sorting Genetic Algorithm II*) (Deb *et al.*, 2002) em 12 funções de teste propostas na literatura especializada.

Reyes-Sierra e Coello (2006c) descrevem algumas experiências feitas a fim de explorar o impacto que os parâmetros principais do algoritmo de PSO possuem, ao usá-lo para otimização multiobjetivo. Foram propostos três mecanismos diferentes para adaptar os valores dos parâmetros de inércia, fatores cognitivo e social. Os mecanismos proposto foram validados usando como teste sete diferentes funções obtidas da literatura especializada de otimização multiobjetivo. Os resultados obtidos mostraram que é possível projetar mecanismos capazes para manter e melhorar a qualidade das soluções obtidas, sem aumento do custo computacional.

Ishida *et al.* (2008) apresentam um método para descoberta de regras de classificação utilizando duas técnicas: o GRASP-PR (*Greedy Randomize Adaptive Search Procedure with Path-Relinking*) e o MOPS (*Multiple Objective Particle Swarm*). O primeiro deles se mostrou competitivo ao se compara com outros algoritmos propostos na literatura, como o ROCCER, C4.5 e CN2. O GRASP-PR superou o MOPS em quase todas as bases de dados, mostrando-se competitivo.

Moore e Chapman (1999) demonstram como PSO pode ser modificado para resolver problemas de otimização para múltiplos objetivos (MOPSO). Salazar-Lechuga e Rowe (2005) também propõem um algoritmo que emprega os conceitos do PSO e otimização multiobjetivo.

Na tabela 1 são mostrados resumidamente as principais características das variações propostas e seus autores.

	Topologia	Seleção de líderes baseada em	Arquivo Externo	W dinâmico	Operador de mutação
Abordagem de agregação					
Parsopolous e Vrahatis (2002b)	Estrela	Único objetivo	Não	Sim	Não
Baumgartner et al. (2004)	Estrela	Único objetivo	Não	Não	Não
Ordenação lexicográfica					
Hu e Eberhart (2002)	Anel	Único objetivo	Não	Sim	Não
Hu et al. (2003)	Anel	Único objetivo	Sim	Sim	Não
Abordagem baseada em sub-população					
Parsopoulos et al. (2004)	Estrela	Único objetivo	Sim	Não	Não
Chow e Tsui (2004)	Estrela	Único objetivo	Não	Não	Não
Abordagem baseada em pareto					
Moore e Chapman (1999)	Anel	Dominância	Não	Não	Não
Ray e Liew (2002)	Estrela	Estimativa de densidade	Sim	Não	Não
Fieldsend e Singh (2002)	Estrela	Dominância	Sim	Não	Sim
Coello et al. (2002, 2004)	Estrela	Densidade das soluções	Sim	Não	Sim
Toscano e Coello (2004)	Estrela	Aleatório	Não	Não	Não
Srinivasan e Hou (2003)	Estrela	Dominância e contagem do nicho	Não	Não	Sim
Mostaghim e Teich (2003)	Estrela	Valor de σ	Sim	Não	Sim
Mostaghim e Teich (2003)	Estrela	Valor de σ	Sim	Não	Sim
Mostaghim e Teich (2004)	Estrela	Valor de σ	Sim	Não	Sim
Bartz et al. (2003)	Estrela	Densidade das soluções	Sim	Não	Não
Li (2003)	Estrela	Contagem do nicho e Estimativa de densidade	Sim	Sim	Sim
Reyes-Siera e Coello (2005)	Estrela	Estimativa de densidade	Sim	Sim	Sim
Alvarez-Benitez et al. (2005)	Estrela	Dominância	Sim	Não	Sim
Ho et al. (2005)	Estrela	<i>Fitness</i> e idade	Sim	Sim	Sim
Villalobos-Arias et al. (2005)	Estrela	<i>Stripes</i>	Sim	Não	Sim
Salazar-Lechuga and Rowe (2005)	Estrela	Contagem do nicho	Sim	Não	Não
Raquel e Naval (2005)	Estrela	Estimativa de densidade	Sim	Não	Sim
Zhao e Cao (2005)	Estrela	Sociedade <i>fuzzy</i>	Sim	Não	Não
Janson e Merkle (2005)	Estrela	Aleatório	Sim	Não	Não

Tabela 1 – Sumário das aproximações de MOPSO (Reyes-Sierra e Coello, 2006) adaptado.

4 MÉTODO PROPOSTO

No 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 possíveis atributos irrelevantes, refinando assim a mineração de dados.

Segundo Freitas (2002), a seleção dos 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 prever o atributo meta.

Um exemplo simples da importância da seleção de atributos - supondo que se quer prever se um cliente é um bom ou mal pagador, baseado no número de atributos precursores sobre o cliente. Provavelmente, na base de dados sobre o cliente existirá um número de identificação original para cada cliente. Caso este atributo seja incluído no conjunto dos atributos dados ao algoritmo da descoberta de regras, é possível que sejam descobertas regras do tipo:

SE (sexo_cliente = feminino) ENTÃO (mau_pagador = sim)

Embora esta regra esteja correta, ela pode ser considerada uma regra irrelevante uma vez que não pode-se afirmar que um cliente que possua sexo igual a feminino será um mau pagador, poderia-se afirmar isso somente se os atributos do antecedente da regra fossem relevantes ao domínio em questão.

Há diversas motivações possíveis para o uso da seleção de atributos. Um subconjunto bem selecionado de atributos conduz frequentemente à descoberta das regras com precisão preditiva mais elevada do que uma descoberta com o conjunto inteiro dos atributos. Este efeito é, às vezes, informalmente descrito como “menos é mais” na literatura. Os atributos irrelevantes podem “confundir” o algoritmo de mineração de dados, de modo que a remoção dos mesmos pode realmente melhorar a qualidade de regras descobertas.

Segundo Pappa (2002), intuitivamente quanto maior o número de atributos em uma base de dados, maior o poder discriminatório do classificador e a facilidade de extrair modelos de conhecimento da base. Porém, quando a teoria é colocada em prática, o mundo real apresenta provas de que nem sempre isso é verdade. Isso porque, em primeiro lugar, muitos métodos de indução sofrem da maldição da dimensionalidade, ou seja, o tempo computacional do algoritmo aumenta exponencialmente de acordo com o número de atributos presentes.

Os objetivos da seleção de atributos são melhorar o desempenho do algoritmo de mineração de dados, remover ruídos e diminuir a dimensionalidade dos dados.

Com o intuito de solucionar problemas com mais de um objetivo, surgiu o conceito de otimização multiobjetivo. Quando utilizado esse conceito permite que, para um determinado problema, um conjunto de soluções ótimas seja apresentado sem privilegiar um ou outro objetivo, e deixa a cargo do usuário ou projetista a escolha da solução que mais se adapte às suas necessidades.

Pensando na seleção de atributos como um problema de otimização multiobjetivo, este trabalho propõe o uso de nuvem de partículas multiobjetivo para a etapa de seleção de atributos. 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 objetivo é a de aplicar otimização multiobjetivo utilizando nuvem de partículas para seleção de atributos. A motivação deste projeto é o sucesso da nuvem de partículas uma abordagem de AE (outros autores a definem como uma técnica de inteligência coletiva) e ao sucesso de busca multiobjetivo, assim unindo duas técnicas de sucesso.

4.1 Abordagem Multiobjetivo de Nuvem de Partículas para Mineração de Dados

O método aqui proposto tem como objetivo avaliar os efeitos da aplicação dos conceitos de nuvem de partículas multiobjetivo na área de mineração de dados, mais especificamente seleção de atributos na tarefa de classificação. Para isto foi utilizado um código publicado em Raquel e Naval (2005) e este código foi adaptado para o problema em questão.

4.1.1 Codificação do Indivíduo ou Partícula

Para representar as partículas foi escolhida a representação binária, ou seja, cada partícula possui 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. Conforme apresentado na Figura 12 da base de dados Íris:

0	1	1	0
---	---	---	---

Figura 12 – Partícula representando atributos selecionados da base Íris.

A base Íris é composta por 4 atributos previsores e um atributo meta, sendo eles: *sepal length*, *sepal width*, *petal length* e *petal width* (atributos previsores) e tipo da flor (atributo meta). A partícula representa os valores dos atributos previsores, ou seja, a Figura 12 indica a presença dos atributos *sepal width* e *petal length* e ausência dos atributos *sepal length* e *petal width*.

4.1.2 Função de *Fitness*

Para o cálculo do *fitness* 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 (Quilan, 1993) 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 (Quilan, 1993).

A Figura 13 mostra como é realizado o cálculo da função de avaliação do método proposto.

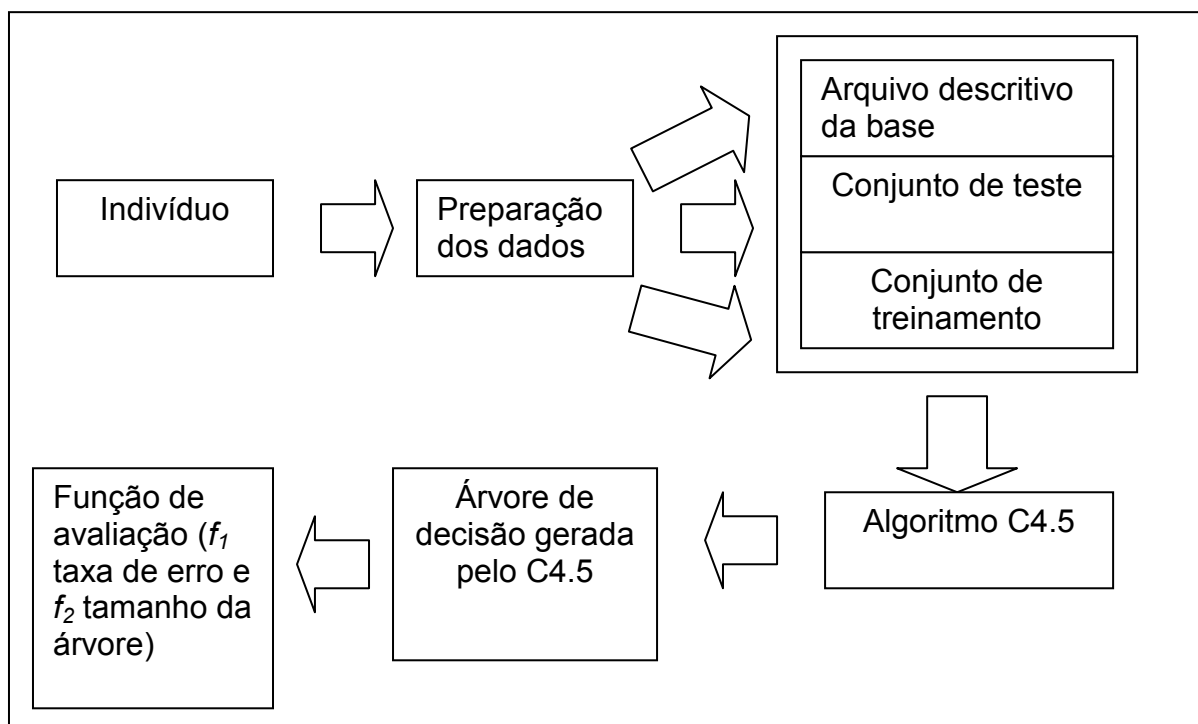


Figura 13 – Cálculo do *fitness*.

4.1.3 Método de Seleção

O método de seleção não teve alterações do algoritmo original e este é baseado em densidade para seleção dos líderes. É selecionada a partícula com mais alto valor de *crowding*, que é a densidade de soluções (partículas) que rodeiam a partícula.

4.1.4 Fator de Apendizagem e Peso da Inércia

No MOPSO os fatores de aprendizagem (C_1 e C_2) e o peso da 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 no livro de Eberhart, Shi e Kennedy (2001).

4.1.5 Pseudo-Código

O pseudo-código apresentado na figura 14 representa o método proposto.

```

Início
  Iniciar população
  Iniciar os líderes em um arquivo externo
  Qualidade (líderes)
   $t = 0$ 
  Enquanto  $t < tmax$ 
    Para cada partícula
      Selecionar líder
      Atualizar Posição (vôo e velocidade)
      Montar base de dados de acordo com a partícula
      Validação cruzada
      Calcular  $f_1$  e  $f_2$ 
      Atualizar  $pbest$ 
    Fim Para
    Atualizar os líderes no arquivo externo
    Qualidade (líderes)
     $t = t + 1$ 
  Fim Enquanto
  Reportar resultados no arquivo externo
Fim

```

Figura 14 – Pseudo-código do método PSO multiobjetivo.

As alterações efetuadas no código original foram realizadas na evolução, ou seja, 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 (Quilan, 1993). 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 (Quilan, 1993) 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 (Quilan, 1993) 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 (Quilan, 1993) 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 15 mostra um resumo de como é realizada a seleção dos conjuntos de dados utilizados na validação cruzada.

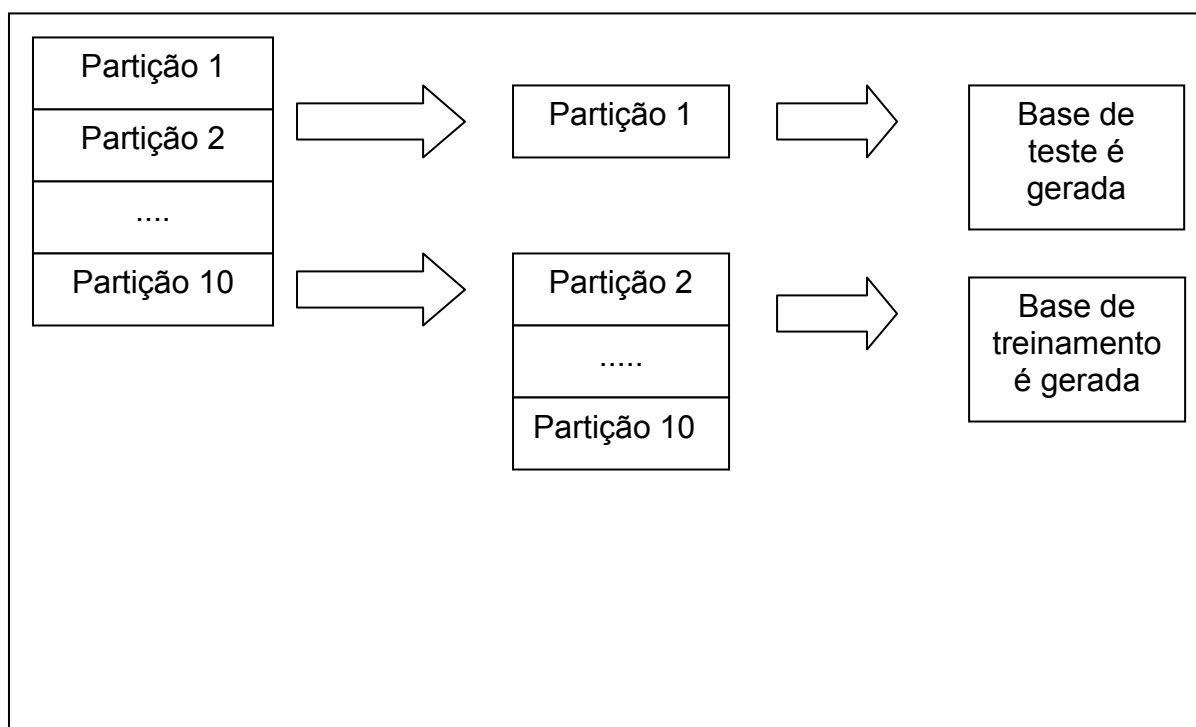


Figura 15 – Seleção dos conjuntos de dados na validação cruzada.

5 EXPERIMENTOS E RESULTADOS OBTIDOS

O algoritmo do MPSO foi processado sob 10 bases de dados obtidas no Repositório de Dados da UCI (Murphy e Aha, 1994). O algoritmo foi processado por 500 gerações, com uma população de 100 indivíduos. Para cada base de dados após o processamento do MPSO foi selecionada a partícula com menor f_1 , a partícula com menor f_2 e a partícula com a 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 2 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
Ionosphere	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 2 – 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 (Carvalho, 2005). Os resultados obtidos são descritos a seguir, nas próximas subseções.

5.1 Base CRX

O base CRX originalmente possui 15 atributos previsores e um atributo meta. Dos atributos previsores, 6 deles são contínuos e 9 atributos são categóricos. O atributo meta pode ser positivo ou negativo. Com a geração das árvores de decisão a partir das partículas selecionadas como relatado no início do Capítulo 5, foram obtidos os resultados apresentados na tabela 3.

Solução	Matriz de confusão gerada no C4.5	%Erro/Tamanho da árvore no teste antes da poda	%Erro/Tamanho da árvore no teste depois da poda
1 - Com todos os atributos	(a) (b) <- classificado como ---- ---- 77 15 (a): classe + 14 101 (b): classe -	15,5% - 91	14,0% - 51
2 - Partícula com Menor f_1	(a) (b) <- classificado como ---- ---- 79 13 (a): classe + 14 101 (b): classe -	11,6% - 123	13,0% - 51
3 - Partícula com Menor f_2	(a) (b) <- classificado como ---- ---- 92 (a): classe + 115 (b): classe -	44,4% - 1	44,4% - 1
4 - Partícula com Média de f_1 e de f_2	(a) (b) <- classificado como ---- ---- 85 7 (a): classe + 23 92 (b): classe -	15,0% - 6	14,5% - 3

Tabela 3 – Resultados para a base CRX.

Na base CRX pode-se observar que com a redução da quantidade de atributos selecionados ocorreu um aumento da taxa de erro tanto na solução 2 quanto na solução 3, porém na solução 4 tanto o percentual de erro quanto o tamanho da árvore foram reduzidos.

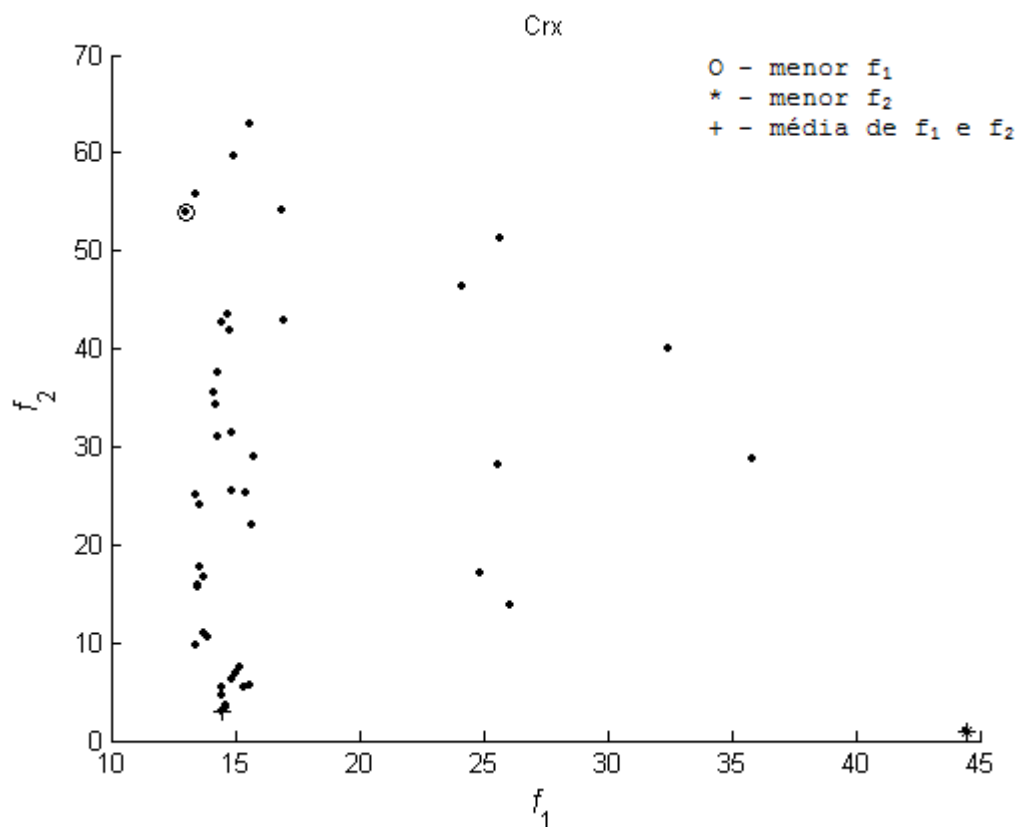


Figura 16 – Conjunto de soluções obtidas para a base CRX.

A figura 16 mostra o conjunto de soluções obtidas na base CRX, a partícula marcada por um círculo é a partícula que apresentou o menor f_1 . A partícula marcada por um asterisco é a partícula que apresentou o menor f_2 , e a partícula marcada por um sinal de positivo é a partícula que apresentou o média de f_1 e f_2 .

5.2 Base Dermatology

O base Dermatology originalmente possui 34 atributos previsores e um atributo meta. Dos atributos previsores 1 deles é contínuo e os outros 33 são categóricos. O atributo meta pode ser 1,2,3,4,5 ou 6. Com a geração das árvores de decisão a partir das partículas selecionadas como relatado no item 5, foram obtidos os resultados mencionados na tabela 4.

Solução	Matriz de confusão gerada no C4.5	%Erro/Tamanho da árvore no teste antes da poda	%Erro/Tamanho da árvore no teste depois da poda
1 - Com todos os atributos	(a) (b) (c) (d) (e) (f) <- classificado como ----- 32 2 (a): classe 1 17 (b): classe 2 21 1 (c): classe 3 1 14 (d): classe 4 16 (e): classe 5 6 (f): classe 6	4,5% - 14	4,5% - 14
2 - Partícula com Menor f_1	(a) (b) (c) (d) (e) (f) <- classificado como ----- 32 2 (a): classe 1 17 (b): classe 2 21 1 (c): classe 3 1 14 (d): classe 4 16 (e): classe 5 6 (f): classe 6	4,5% - 14	4,5% - 14
3 - Partícula com Menor f_2	(a) (b) (c) (d) (e) (f) <- classificado como ----- 32 2 (a): classe 1 17 (b): classe 2 20 1 1 (c): classe 3 2 13 (d): classe 4 16 (e): classe 5 6 (f): classe 6	7,2% - 34	6,3% - 11
4 - Partícula com Média de f_1 e de f_2	(a) (b) (c) (d) (e) (f) <- classificado como ----- 32 2 (a): classe 1 17 (b): classe 2 21 1 (c): classe 3 1 14 (d): classe 4 16 (e): classe 5 6 (f): classe 6	4,5% - 14	4,5% - 14

Tabela 4 – Resultados para a base Dermatology.

Na base Dermatology pode-se observar que com a redução da quantidade de atributos selecionados, na solução 2 o percentual de erro e o tamanho da árvore se mantiveram constantes em relação solução 1, a solução 3 teve um aumento no erro e uma redução no tamanho da árvores em relação a solução 1, já na solução 4 o percentual de erro teve uma redução e o tamanho da árvore não teve alteração.

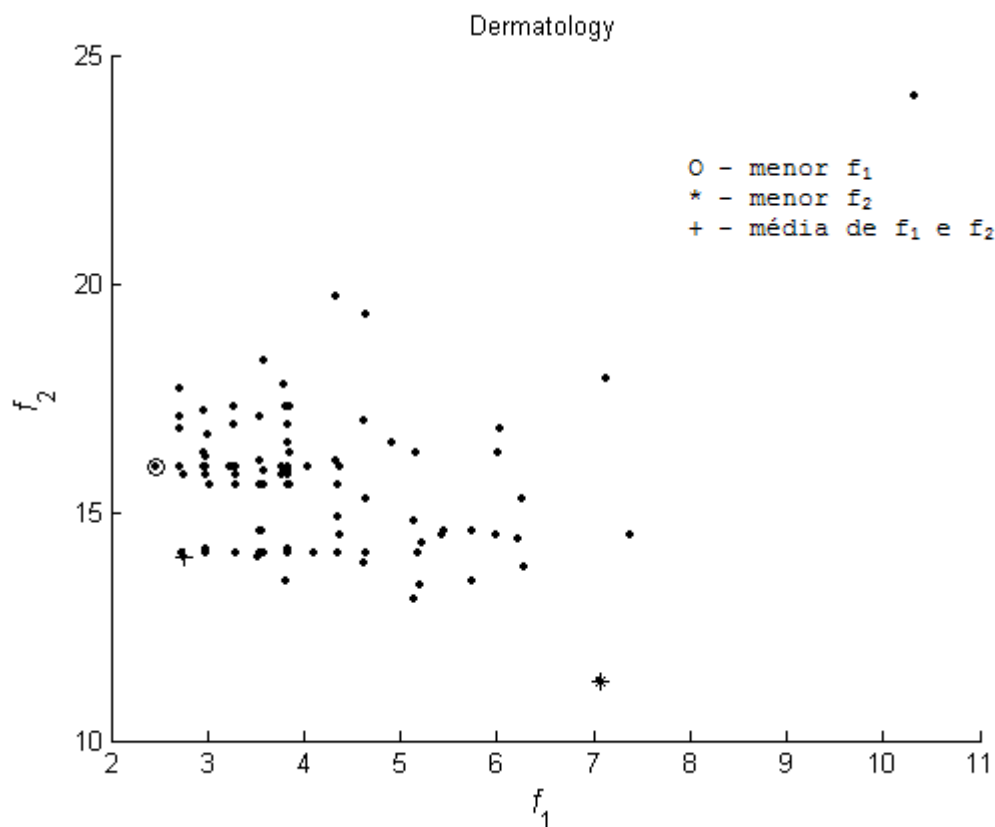


Figura 17 – Conjunto de soluções obtidas para a base Dermatology.

A figura 17 mostra o conjunto de soluções obtidas na base Dermatology, a partícula marcada por um círculo é a partícula que apresentou o menor f_1 . A partícula marcada por um asterisco é a partícula que apresentou o menor f_2 , e a partícula marcada por um sinal de positivo é a partícula que apresentou o média de f_1 e f_2 .

5.3 Base Glass

O base Glass originalmente possui 10 atributos previsores e um atributo meta. Todos os atributos previsores são contínuos. O atributo meta pode ser 1, 2, 3, 4, 5 ou 6. Com a geração das árvores de decisão a partir das partículas

selecionadas como relatado no item 5, foram obtidos os resultados apresentados na tabela 5.

Solução	Matriz de confusão gerada no C4.5	%Erro/Tamanho da árvore no teste antes da poda	%Erro/Tamanho da árvore no teste depois da poda
1 - Com todos os atributos	<pre> (a) (b) (c) (d) (e) (f) (g) <- classificado como ----- 21 (a): classe 1 22 1 (b): classe 2 5 1 (c): classe 3 4 (d): classe 4 3 (e): classe 5 9 (f): classe 6 (g): classe 7 </pre>	3,0% - 11	3,0% - 11
2 - Partícula com Menor f	<pre> (a) (b) (c) (d) (e) (f) (g) <- classificado como ----- 21 (a): classe 1 22 1 (b): classe 2 5 1 (c): classe 3 4 (d): classe 4 3 (e): classe 5 9 (f): classe 6 (g): classe 7 </pre>	3,0% - 11	3,0% - 11
3 - Partícula com Menor f_2	<pre> (a) (b) (c) (d) (e) (f) (g) <- classificado como ----- 20 (a): classe 1 23 (b): classe 2 6 (c): classe 3 4 (d): classe 4 3 (e): classe 5 1 (f): classe 6 8 (g): classe 7 </pre>	53,0% - 3	53,0% - 3
4 - Partícula com Média de f_1 e de f_2	<pre> (a) (b) (c) (d) (e) (f) (g) <- classificado como ----- 21 (a): classe 1 22 1 (b): classe 2 5 1 (c): classe 3 4 (d): classe 4 3 (e): classe 5 9 (f): classe 6 (g): classe 7 </pre>	3,0% - 11	3,0% - 11

Tabela 5 – Resultados para a base Glass.

Na base Glass pode-se observar que as soluções 2 e 4 mantiveram os valores de percentual de erro e tamanho de árvores iguais ao da partícula 1, e a solução 3 teve um aumento do percentual de erro e uma redução no tamanho da árvore.

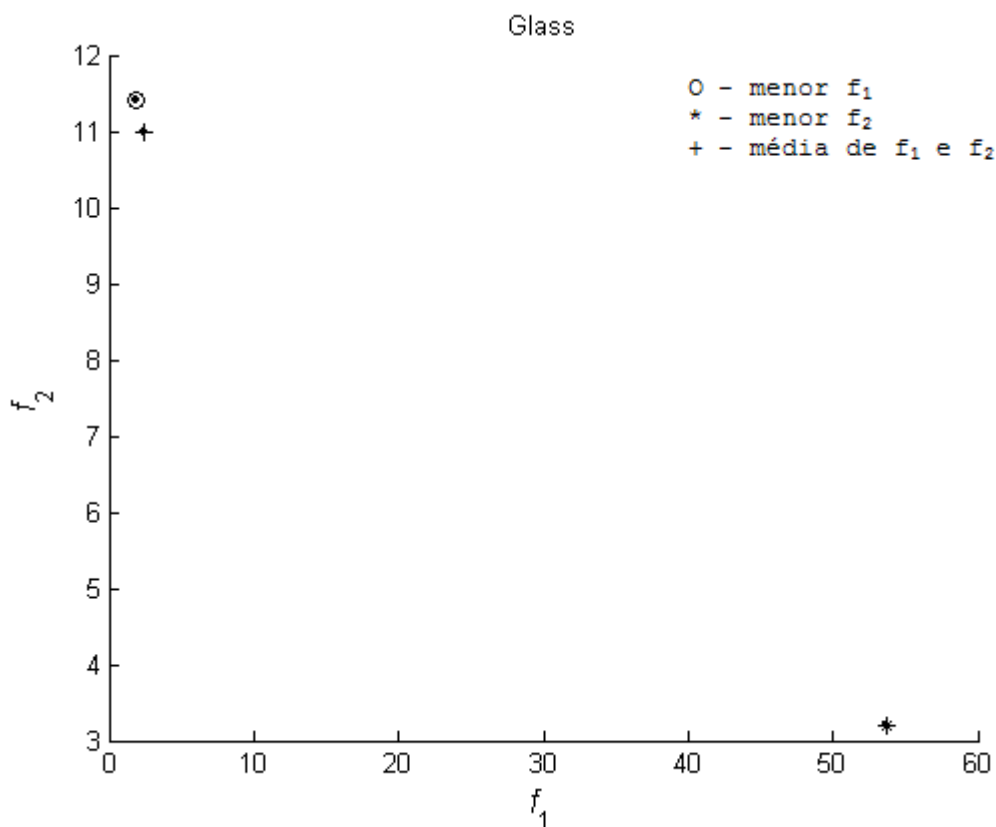


Figura 18 – Conjunto de soluções obtidas para a base Glass.

A figura 18 mostra o conjunto de soluções obtidas na base Glass, a partícula marcada por um círculo é a partícula que apresentou o menor f_1 . A partícula marcada por um asterisco é a partícula que apresentou o menor f_2 , e a partícula marcada por um sinal de positivo é a partícula que apresentou o média de f_1 e f_2 .

5.4 Base Ionosphere

O base Ionosphere originalmente possui 34 atributos previsores e um atributo meta. Todos os atributos previsores são contínuos. O atributo meta pode ser g ou b . Com a geração das árvores de decisão a partir das partículas selecionadas como relatado no item 5, foram obtidos os resultados descritos na tabela 6.

Solução	Matriz de confusão gerada no C4.5	%Erro/Tamanho da árvore no teste antes da poda	%Erro/Tamanho da árvore no teste depois da poda
1 - Com todos os atributos	(a) (b) <- classificado como ----- 67 2 (a): classe g 5 31 (b): classe b	7,6% - 31	6,7% - 27
2 - Partícula com Menor f_1	(a) (b) <- classificado como ----- 67 2 (a): classe g 2 34 (b): classe b	3,8% - 27	3,8% - 19
3 - Partícula com Menor f_2	(a) (b) <- classificado como ----- 66 3 (a): classe g 4 32 (b): classe b	7,6% - 27	6,7% - 23
4 - Partícula com Média de f_1 e de f_2	(a) (b) <- classificado como ----- 66 3 (a): classe g 4 32 (b): classe b	13,3% - 33	6,7% - 15

Tabela 6 – Resultados para a base Ionosphere.

Na base Ionosphere pode-se observar redução no percentual de erro apenas na solução 2.

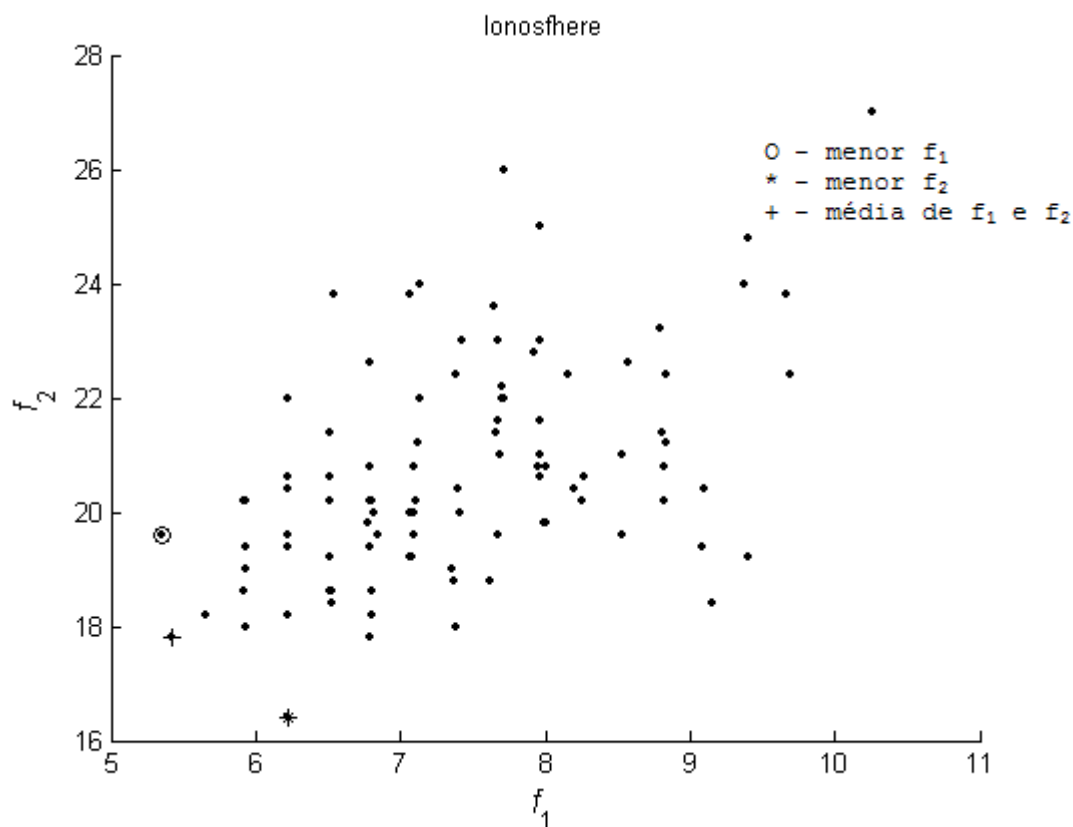


Figura 19 – Conjunto de soluções obtidas para a base lonosphere.

A figura 19 mostra o conjunto de soluções obtidas na base lonosphere, a partícula marcada por um círculo é a partícula que apresentou o menor f_1 . A partícula marcada por um asterisco é a partícula que apresentou o menor f_2 , e a partícula marcada por um sinal de positivo é a partícula que apresentou o média de f_1 e f_2 .

5.5 Base Mushroom

O base Mushroom originalmente possui 22 atributos previsores e um atributo meta. Todos os atributos previsores são categóricos. O atributo meta pode ser e ou p . Com a geração das árvores de decisão a partir das partículas selecionadas como relatado no item 5, foram obtidos os resultados descritos na tabela 7.

Solução	Matriz de confusão gerada no C4.5	%Erro/Tamanho da árvore no teste antes da poda	%Erro/Tamanho da árvore no teste depois da poda
1 - Com todos os atributos	(a) (b) <- classificado como ----- 1213 48 (a): classe e 224 950 (b): classe p	11,2% - 16	11,2% - 14
2 - Partícula com Menor f_1	(a) (b) <- classificado como ----- 1213 48 (a): classe e 224 950 (b): classe p	11,2% - 14	11,2% - 14
3 - Partícula com Menor f_2	(a) (b) <- classificado como ----- 1261 (a): classe e 24 950 (b): classe p	9,2% - 18	9,2% - 12
4 - Partícula com Média de f_1 e de f_2	(a) (b) <- classificado como ----- 1213 48 (a): classe e 224 950 (b): classe p	11,2% - 15	11,2% - 13

Tabela 7 – Resultados para a base Mushroom.

Na base Mushroom pode-se observar que houve pouca variação no percentual de erro e tamanho de árvore ao comparar as soluções 2, 3 e 4 com a solução 1.

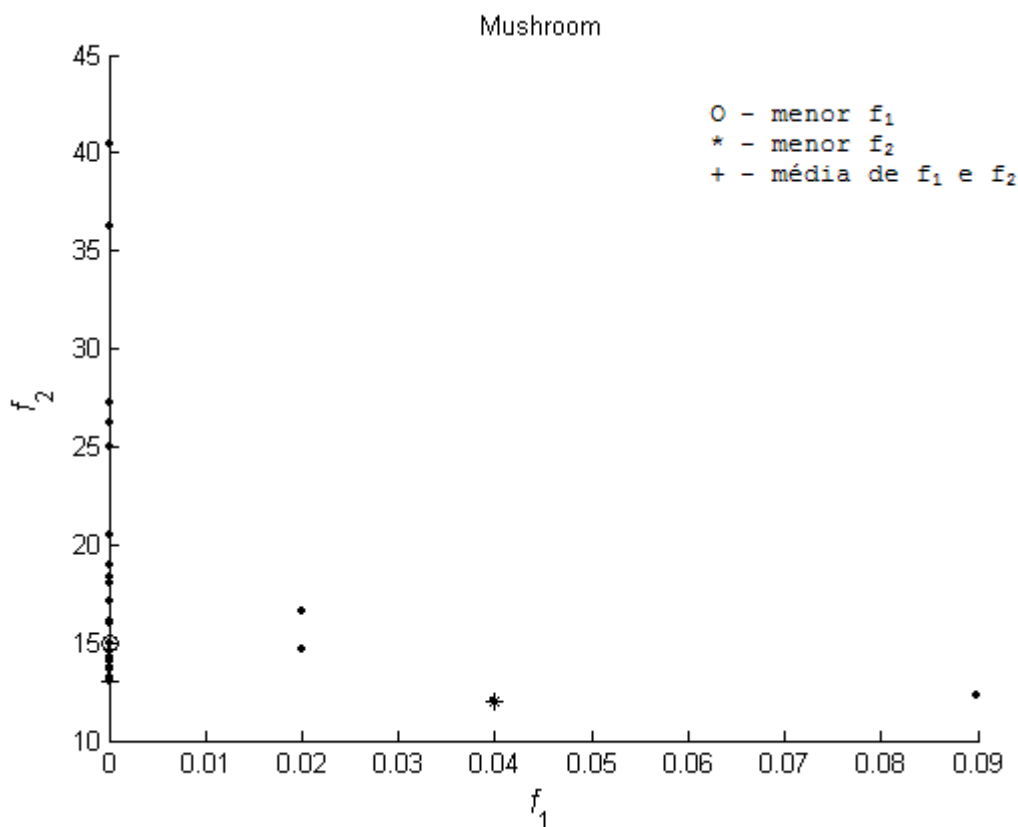


Figura 20 – Conjunto de soluções obtidas para a base Mushroom.

A figura 20 mostra o conjunto de soluções obtidas na base Mushroom, a partícula marcada por um círculo é a partícula que apresentou o menor f_1 . A partícula marcada por um asterisco é a partícula que apresentou o menor f_2 , e a partícula marcada por um sinal de positivo é a partícula que apresentou o média de f_1 e f_2 .

5.6 Base Promoters

O base Promoters originalmente possui 57 atributos previsores e um atributo meta. Todos os atributos são categóricos. O atributo meta pode ser positivo ou negativo. Com a geração das árvores de decisão a partir das partículas selecionadas como relatado no item 5, foram obtidos os resultados mencionados na tabela 8.

Solução	Matriz de confusão gerada no C4.5	%Erro/Tamanho da árvore no teste antes da poda	%Erro/Tamanho da árvore no teste depois da poda
1 - Com todos os atributos	(a) (b) <- classificado como ----- 15 1 (a): classe + 4 12 (b): classe -	15,6% - 18	15,6% - 18
2 - Partícula com Menor f_1	(a) (b) <- classificado como ----- 14 2 (a): classe + 4 12 (b): classe -	18,8% - 16	18,8% - 16
3 - Partícula com Menor f_2	(a) (b) <- classificado como ----- 15 1 (a): classe + 4 12 (b): classe -	15,6% - 17	15,6% - 17
4 - Partícula com Média de f_1 e de f_2	(a) (b) <- classificado como ----- 15 1 (a): classe + 3 13 (b): classe -	12,5% - 15	12,5% - 15

Tabela 8 – Resultados para a base Promoters.

Na base Promoters pode-se observar que houve pouca variação no percentual de erro e tamanho de árvore ao comparar as soluções 2, 3 e 4 com a solução 1.

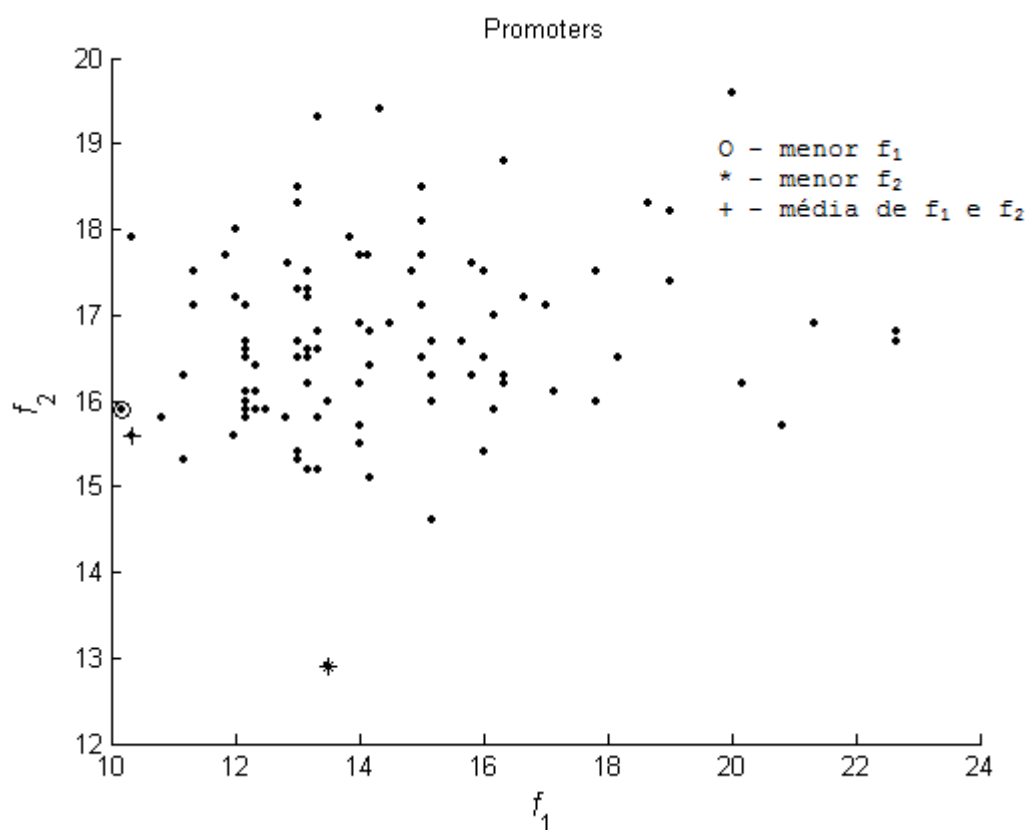


Figura 21 – Conjunto de soluções obtidas para a base Promoters.

A figura 21 mostra o conjunto de soluções obtidas na base Promoters, a partícula marcada por um círculo é a partícula que apresentou o menor f_1 . A partícula marcada por um asterisco é a partícula que apresentou o menor f_2 , e a partícula marcada por um sinal de positivo é a partícula que apresentou o média de f_1 e f_2 .

5.7 Base Sick-Euthyroid

O base Sick-Euthyroid originalmente possui 25 atributos previsores e um atributo meta. Dos atributos previsores 6 são contínuos e 19 são categóricos. O atributo meta pode ser sick-euthyroid ou negativo. Com as geração das árvores de decisão a partir das partículas selecionadas como relatado no item 5, foram obtidos os resultados descritos na tabela 9.

Solução	Matriz de confusão gerada no C4.5	%Erro/Tamanho da árvore no teste antes da poda	%Erro/Tamanho da árvore no teste depois da poda
1 - Com todos os atributos	(a) (b) <- classificado como ----- 74 14 (a): classe sick-euthyroid 9 851 (b): classe negative	3,3% - 87	2,4% - 27
2 - Partícula com Menor f_1	(a) (b) <- classificado como ----- 76 12 (a): classe sick-euthyroid 9 851 (b): classe negative	3,2% - 83	2,2% - 25
3 - Partícula com Menor f_2	(a) (b) <- classificado como ----- 88 (a): classe sick-euthyroid 860 (b): classe negative	9,6% - 99	9,3% - 1
4 - Partícula com Média de f_1 e de f_2	(a) (b) <- classificado como ----- 78 10 (a): classe sick-euthyroid 13 847 (b): classe negative	2,7% - 81	2,4% - 7

Tabela 9 – Resultados para a base Sick-Euthyroid.

Na base Sick-Euthyroid pode-se observar que a solução 2 obteve uma redução no percentual de erro e também no tamanho de árvores em relação a solução 1. As soluções 3 e 4 tiveram resultados piores em relação a solução 1.

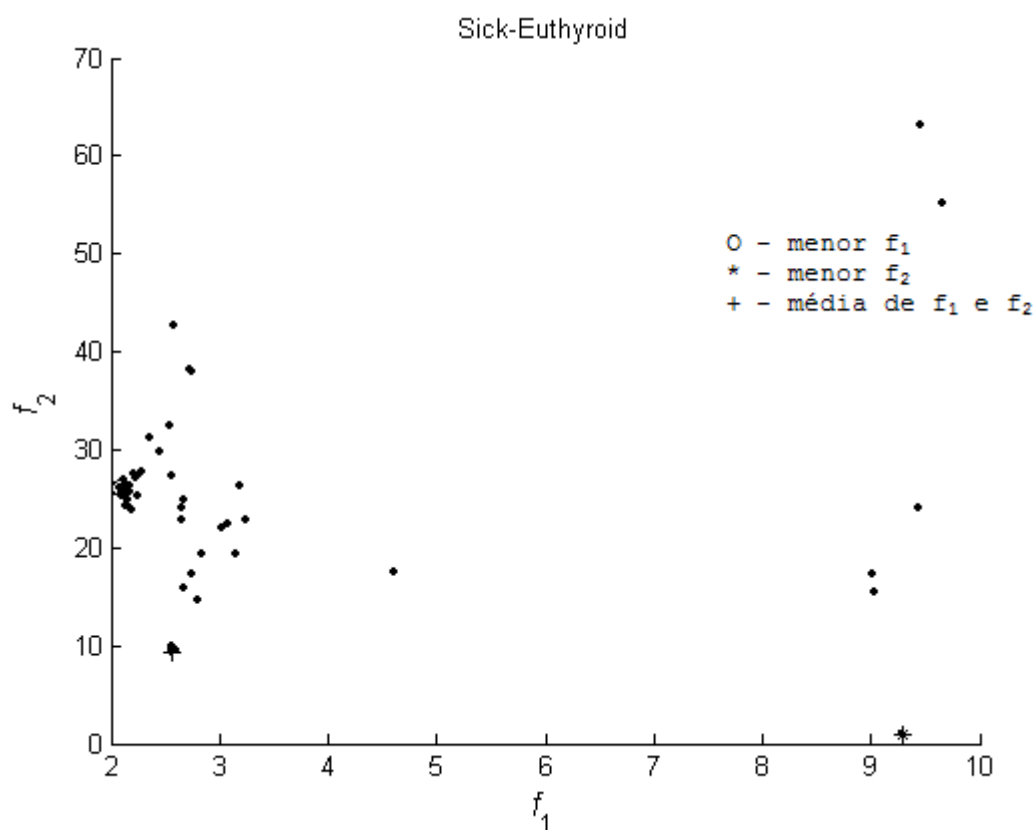


Figura 22 – Conjunto de soluções obtidas para a base Sick-Euthyroid.

A figura 22 mostra o conjunto de soluções obtidas na base Sick-Euthyroid, a partícula marcada por um círculo é a partícula que apresentou o menor f_1 . A partícula marcada por um asterisco é a partícula que apresentou o menor f_2 , e a partícula marcada por um sinal de positivo é a partícula que apresentou o média de f_1 e f_2 .

5.8 Base Vehicle

O base Vehicle originalmente possui 18 atributos previsores e um atributo meta. Todos os atributos previsores são contínuos. O atributo meta pode ser opel, saab, bus ou van. Com a geração das árvores de decisão a partir das partículas selecionadas como relatado no item 5, foram obtidos os resultados descritos na tabela 10.

Solução	Matriz de confusão gerada no C4.5	%Erro/Tamanho da árvore no teste antes da poda	%Erro/Tamanho da árvore no teste depois da poda
1 - Com todos os atributos	(a) (b) (c) (d) <- classificado como ----- 43 15 1 4 (a): classe opel 20 39 5 (b): classe saab 1 3 61 (c): classe bus 2 2 1 55 (d): classe van	21,8% - 163	21,4% - 155
2 - Partícula com Menor f_1	(a) (b) (c) (d) <- classificado como ----- 32 30 1 (a): classe opel 23 34 3 4 (b): classe saab 2 62 1 (c): classe bus 1 6 53 (d): classe van	26,6% - 209	28,2% - 165
3 - Partícula com Menor f_2	(a) (b) (c) (d) <- classificado como ----- 30 28 1 4 (a): classe opel 26 30 2 6 (b): classe saab 2 63 (c): classe bus 2 8 1 49 (d): classe van	31,3% - 201	31,7% - 171
4 - Partícula com Média de f_1 e de f_2	(a) (b) (c) (d) <- classificado como ----- 30 28 1 4 (a): classe opel 26 30 2 6 (b): classe saab 2 63 (c): classe bus 2 8 1 49 (d): classe van	31,3% - 201	31,7% - 171

Tabela 10– Resultados para a base Vehicle.

Na base Vehicle pode-se observar que as soluções 2, 3 e 4 não tiveram bons resultados comparadas a solução 1.

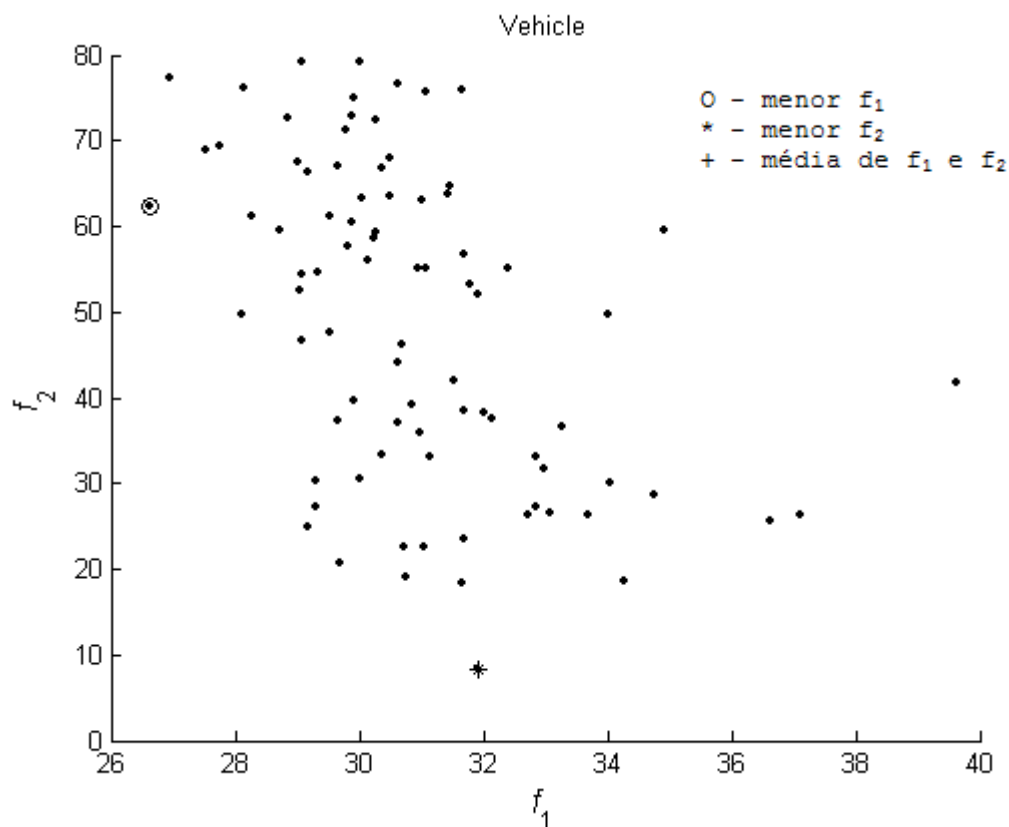


Figura 23 – Conjunto de soluções obtidas para a base Vehicle.

A figura 23 mostra o conjunto de soluções obtidas na base Vehicle, a partícula marcada por um círculo é a partícula que apresentou o menor f_1 . A partícula marcada por um asterisco é a partícula que apresentou o menor f_2 , e a partícula marcada por um sinal de positivo é a partícula que apresentou o média de f_1 e f_2 .

5.9 Base Votes

O base Votes originalmente possui 16 atributos previsores e um atributo meta. Todos os atributos previsores são categóricos. O atributo meta pode ser democrata ou republicano. Com a geração das árvores de decisão a partir das partículas selecionadas como relatado no item 5, foram obtidos os resultados descritos na tabela 11.

Solução	Matriz de confusão gerada no C4.5	%Erro/Tamanho da árvore no teste antes da poda	%Erro/Tamanho da árvore no teste depois da poda
1 - Com todos os atributos	(a) (b) <- classificado como ----- 79 1 (a): classe democrat 2 48 (b): classe republican	1,5% - 37	2,3% - 11
2 - Partícula com Menor f_1	(a) (b) <- classificado como ----- 79 1 (a): classe democrat 2 48 (b): classe republican	2,3% - 11	2,3% - 11
3 - Partícula com Menor f_2	(a) (b) <- classificado como ----- 77 3 (a): classe democrat 2 48 (b): classe republican	6,2% - 23	3,8% - 3
4 - Partícula com Média de f_1 e de f_2	(a) (b) <- classificado como ----- 77 3 (a): classe democrat 2 48 (b): classe republican	6,2% - 23	3,8% - 3

Tabela 11– Resultados para a base Votes.

Na base Votes pode-se observar que apenas a solução 2 obteve resultado igual a solução 1, as demais obtiveram resultados piores se comparados a solução 1.

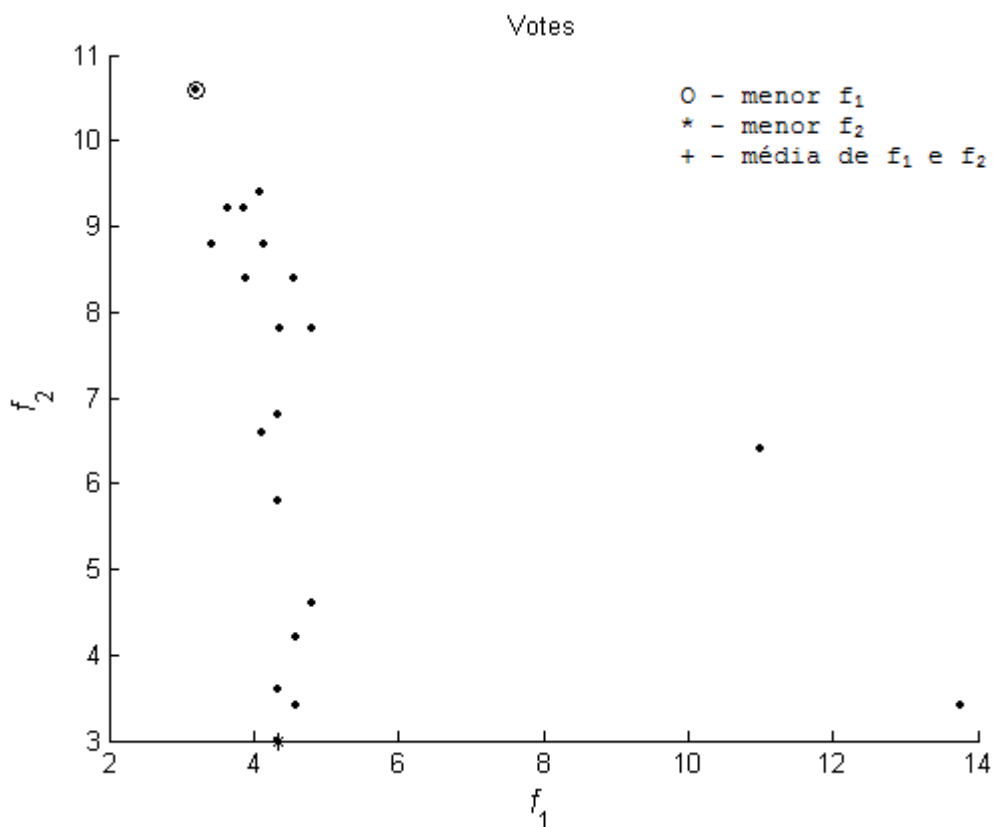


Figura 24 – Conjunto de soluções obtidas para a base Votes.

A figura 24 mostra o conjunto de soluções obtidas na base Votes, a partícula marcada por um círculo é a partícula que apresentou o menor f_1 . A partícula marcada por um asterisco é a partícula que apresentou o menor f_2 , e a partícula marcada por um sinal de positivo é a partícula que apresentou o média de f_1 e f_2 .

5.10 Base Wine

O base Wine originalmente possui 13 atributos previsores e um atributo meta. Todos os atributos previsores são contínuos. O atributo meta pode ser 1, 2 ou 3. Com a geração das árvores de decisão a partir das partículas

selecionadas como relatado no item 5, foram obtidos os resultados descritos na tabela 12.

Solução	Matriz de confusão gerada no C4.5	%Erro/Tamanho da árvore no teste antes da poda	%Erro/Tamanho da árvore no teste depois da poda
1 - Com todos os atributos	(a) (b) (c) ----- 16 2 1 17 3 2 12	<- classificado como	15,1% - 11
2 - Partícula com Menor f_1	(a) (b) (c) ----- 18 3 18 14	(a): classe 1 (b): classe 2 (c): classe 3	5,7% - 9
3 - Partícula com Menor f_2	(a) (b) (c) ----- 18 1 20 14	(a): classe 1 (b): classe 2 (c): classe 3	1,9% - 9
4 - Partícula com Média de f_1 e de f_2	(a) (b) (c) ----- 18 1 20 14	(a): classe 1 (b): classe 2 (c): classe 3	1,9% - 9

Tabela 12– Resultados para a base Wine.

Na base Wine pode-se observar houve uma grande redução no percentual de erro, comparando as soluções 2, 3 e 4 à solução 1. O tamanho de árvore também teve redução.

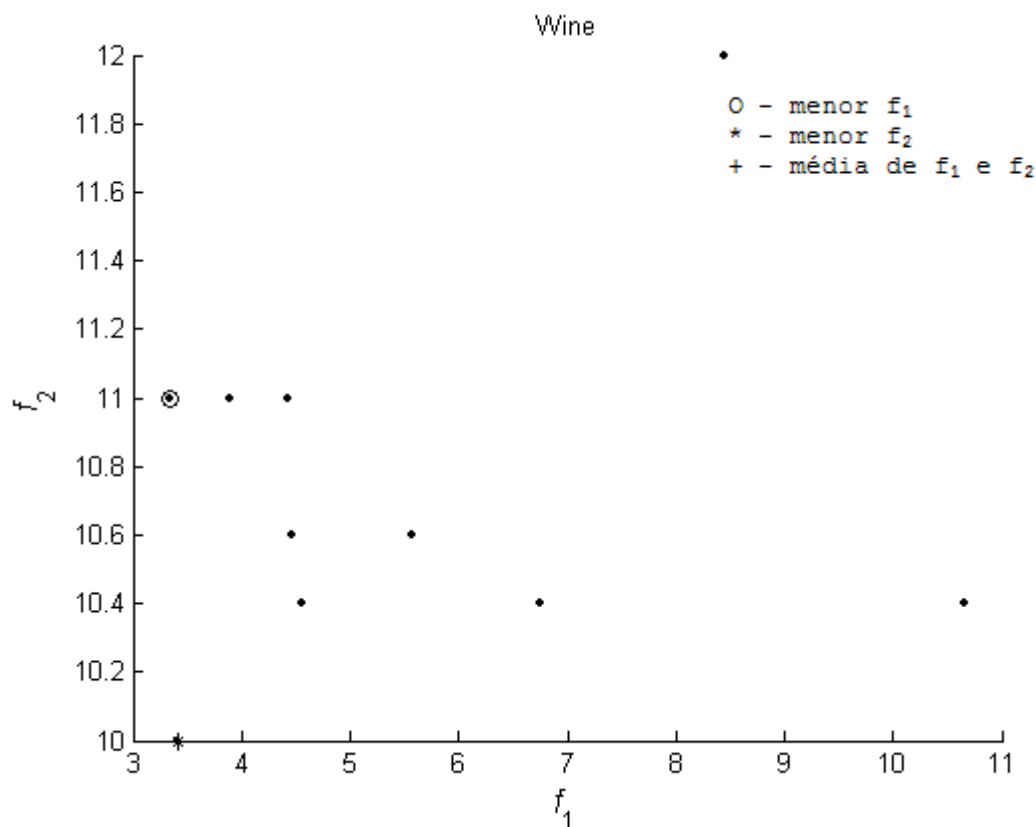


Figura 25 – Conjunto de soluções obtidas para a base Wine.

A figura 25 mostra o conjunto de soluções obtidas na base Wine, a partícula marcada por um círculo é a partícula que apresentou o menor f_1 . A partícula marcada por um asterisco é a partícula que apresentou o menor f_2 , e a partícula marcada por um sinal de positivo é a partícula que apresentou o média de f_1 e f_2 .

5.11 Resumo dos Resultados Obtidos Considerando Erro e Tamanho das Árvores

As tabelas 13 e 14 mostram um resumo dos resultados obtidos de todas as bases testadas.

Bases de dados	Arvore 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

Tabela 13 – Resumo dos resultados obtidos.

Bases de dados	Arvore de decisão gerada	f_1	F_2	Taxa de Erro	Tamanho da árvore	Quantidade de atributos
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 14 – Resumo dos resultados obtidos.

Analisando todos os resultados para as bases testadas 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.

5.12 Análise Considerando Métodos de Desempenho

Existem dois métodos para análise de desempenho de algoritmos evolutivos multiobjetivo: o método analítico e empírico. O método analítico que utiliza a prova de teoremas baseada em um modelo matemático de cálculo. E o método empírico que utiliza de experimentos computacionais.

As técnicas de avaliação de desempenho podem também ser classificadas em duas categorias principais, denominadas de abordagem *a posteriori* e abordagem *a priori*.

Em relação a abordagem *a priori*, esta pode ser classificada em duas categorias, a baseada em solução e a baseada em comparação relativa.

Para a abordagem *a posteriori*, a frente real de Pareto é necessária e utilizada como referência para as medidas de desempenho. No entanto, a desvantagem desta abordagem é que a obtenção da frente real de Pareto, muitas vezes, não é avaliável ou mesmo impossível de ser calculada. No entanto, Ang *et al.* (2001) propuseram a utilização de uma aproximação da frente de Pareto quando a frente real de Pareto não esta disponível.

Segundo Zitzler *et al.* (2000), as três metas fundamentais das métricas propostas na atualidade para medir desempenho de métodos de otimização multiobjetivo são:

- (i) minimizar a distância da frente de Pareto produzida pelo algoritmo de otimização;
- (ii) maximizar a distribuição das soluções obtidas, de forma a obter uma distribuição uniforme das soluções no espaço de objetivos; e
- (iii) maximizar a quantidade de elementos do conjunto de ótimos de Pareto gerados.

Na literatura, diversas métricas de desempenho têm sido propostas, tais como: distribuição (espaçamento) das soluções encontradas, distância das soluções à frente real de Pareto (distância geracional), índice de diversidade das soluções, medidas de distribuição χ -quadrado, tamanho do espaço de objetivos coberto, entre outras (Deb, 2001; Coello Coello *et al.*, 2002; Knowles *et al.*, 2006).

Das bases testadas em relação a f_1 e f_2 neste trabalho, por desconhecimento da frente real de Pareto são adotados dois índices de desempenho, uma métrica de diversidade (espalhamento das soluções) e uma métrica de convergência (distância das soluções a origem dos eixos cartesianos para o problema de minimização de f_1 e f_2 (0,0)). Estas métricas são detalhas a seguir.

Métrica 1 – Distribuição (espaçamento) das soluções encontradas (*Spacing, S*): Esta medida foi proposta em Scott (1995) e calcula o desvio padrão entre as distâncias de soluções consecutivas obtidas pelo algoritmo de otimização. Esta medida é regida pela equação:

$$S = \sqrt{\frac{1}{|Q|} \sum_{i=1}^{|Q|} (\bar{d} - d_i)^2} \quad (6)$$

onde

$$d_i = \min_{k \in Q \text{ e } k \neq i} \sum_{m=1}^M |f_m^i - f_m^k| \quad (7)$$

$$\bar{d} = \frac{\sum_{i=1}^{|Q|} d_i}{|Q|} \quad (8)$$

A variável Q armazena o número de soluções obtidas pelo método de otimização. A variável d_i é a distância entre cada solução encontrada e é calculada pelo mínimo somatório do valor absoluto da diferença dos valores das funções objetivo entre a solução i e qualquer outra solução em Q . E \bar{d} representa a média aritmética de todas as distâncias d_i . Nesta medida, quanto menor o desvio S , melhor distribuídas estão as soluções encontradas (Ticona, 2003).

Métrica 2 – Distância média entre o conjunto Q e a origem dos eixos cartesianos. Esta métrica objetiva avaliar que algoritmo obteve uma melhor convergência do melhor indivíduo da população de soluções não-dominadas. Consiste de uma variante (para o caso da frente de Pareto ser desconhecida) da distância geracional, esta proposta por Veldhuzien (1999).

Para $p=2$, a variável d_i é a distância Euclidiana no espaço de objetivos entre a solução $i \in Q$ e a origem dos eixos n -dimensional. Especificamente, esta métrica para o caso bi-dimensional adotado é dada por

$$DO = \text{média} \left(\sqrt{\sum_{m=1}^M (f_m^i)^2} \right) \quad (9)$$

onde f_m^i é o valor m -ésimo objetivo da i -ésima solução encontrada pelo método de otimização multiobjetivo.

Em síntese, foram adotadas as duas métricas mencionadas para avaliar o método de otimização proposto, sendo elas: o espaçamento das soluções encontradas (métrica 1) e a distância média a origem dos eixos cartesianos (métrica 2). Os resultados são mencionados na tabela 14.

Base de Dados	Espaçamento das soluções	Distância média a origem dos eixos cartesianos	Menor valor de f_1	Menor valor de f_2	Menor média de f_1 e f_2
Crx	7,98	1,43	13,04	1,00	0,18
Dermatology	1,00	0,81	2,46	11,30	0,42
Glass	5,82	0,62	1,91	3,20	0,50
Ionosphere	0,40	1,10	5,35	16,40	0,59
Mushroom	0,60	0,77	0,00	12,00	0,16
Promoters	0,44	1,10	10,17	12,90	0,62
Sick-Euthyroid	6,58	1,13	2,09	1,00	0,20
Vehicle	3,05	2,89	26,63	8,40	0,45
Votes	0,92	0,46	3,20	3,00	0,29
Wine	0,38	0,53	3,35	10,00	0,57

Tabela 15 – Espaçamento e distância euclidiana média a origem dos eixos cartesianos.

Considerando a métrica do espaçamento pode-se considerar que o melhor resultado obtido foi na base Wine, pois esta apresentou o menor valor, o valor de 0,38. E a base CRX apresentou o pior resultado, pois apresentou o maior valor, o valor de 7,98.

Considerando a métrica de distância média entre o conjunto Q e a origem dos eixos cartesianos, verifica-se que a base que apresentou o melhor resultado foi a base Votes, apresentando o valor de 0,46 e a base que apresentou o pior resultado foi base Vehicle com o valor 2,89.

5.13 Resultados Obtidos Com 5 Experimentos

Para as bases Sick-Euthyroid e Vehicle foram executados 5 experimentos, ou seja o algoritmo foi processado por 500 gerações, com uma população de 100 indivíduos por 5 vezes. Estas bases foram selecionadas por terem sido consideradas bases promissoras para se obter o conjunto de Pareto. A seguir são descritos os resultados obtidos.

A figura 26 mostra o conjunto de Pareto obtido com o processamento da base Sick-Euthyroid. Com relação as medidas de desempenho foi obtido o valor de 6,074016 para espaçamento e de 0.064990 para a distância médias à origem dos eixos cartesianos.

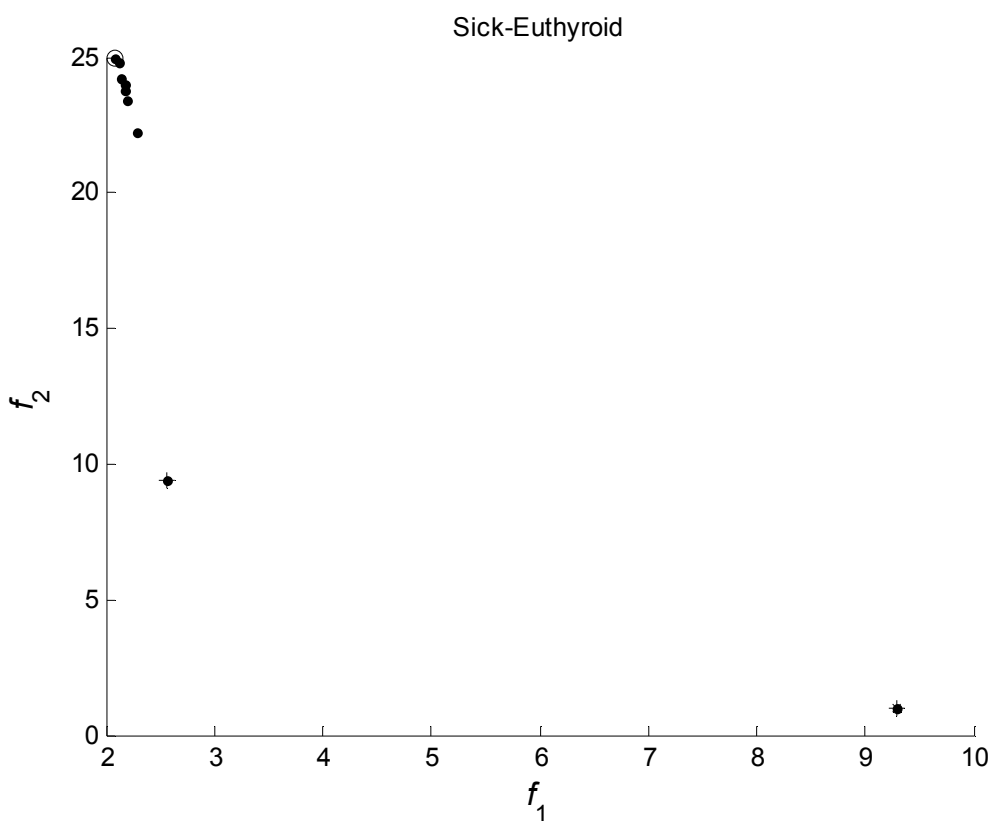


Figura 26 – Conjunto de Pareto para a base Sick-Euthyroid.

A figura 27 mostra o conjunto de Pareto obtido com o processamento da base Vehicle. Com relação as medidas de desempenho foi

obtido o valor de 3.909649 para espaçamento e de 0.188496 para a distância médias à origem dos eixos cartesianos.

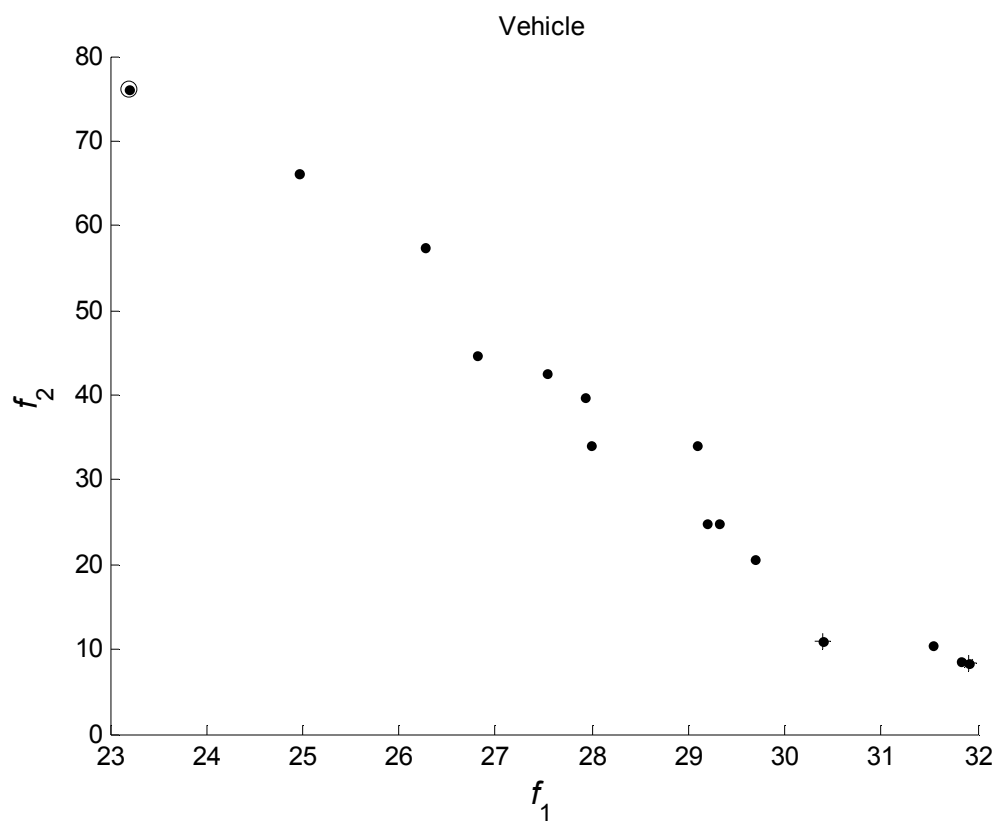


Figura 27 – Conjunto de Pareto para a base Vehicle.

6 CONCLUSÃO E RECOMENDAÇÕES PARA TRABALHOS FUTUROS

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 (Quilan, 1993).

O C4.5 (Quilan, 1993) 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 6 das 10 bases da UCI (Murphy e Aha, 1994) utilizadas nos experimentos. Em 2 bases de dados, os resultados foram semelhantes a solução padrão e em 2 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.

Como proposta de trabalhos futuros, uma análise das características das bases de dados em que o método proposto usando nuvem de partículas combinado ao algoritmo C4.5 (Quilan, 1993) apresentou melhores resultados deve ser realizada. Neste contexto, buscando a descoberta de padrões que identifiquem as bases em que o método pode ser aplicado com sucesso. Outra proposta é a de fazer variações do método proposto, como aumentar o número de objetivos a serem otimizados, variar as taxas de aprendizado e de inércia do algoritmo de nuvem de partículas.

Outra opção é adotar métricas de avaliação para o algoritmo de nuvem de partículas multiobjetivo com a realização de diversos experimentos, aplicá-las para uma comparação mais formal do algoritmo proposto com diferentes variações no projeto do algoritmo de otimização.

7 REFERÊNCIAS BIBLIOGRÁFICAS

ALVES, R. T.; SILVA, E. A. **Aquisição de Conhecimento Através de Algoritmos Genéticos**. Monografia de conclusão de graduação em Tecnologia em Processamento de Dados, Universidade Tuiuti do Paraná, UTP, Curitiba, PR, 2002.

ANGELINE, P. J. **Evolutionary Optimization Versus Particle Swarm Optimization: Philosophy and Performance Differences**. In V.W. Porto, N. Saravanan, D. Waagen, A.E. Eiben (editors), Evolutionary Programming VII. 7th International Conference, EP 98, p. 601-610. Springer. Lecture Notes in Computer Science v. 1447, San Diego, CA, USA, 1998.

BÄCK, T. **Selective Pressure in Evolutionary Algorithms: A Characterization of Selection Mechanisms**. Proceedings of the 1^o. IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence, p. 57-62, Florida, USA, 1994.

BRANKE, J.; MOSTAGHIM, S. **About Selecting the Personal Best in Multi-objective Particle Swarm Optimization**. Proceedings of 9th International Conference on Parallel Problem Solving from Nature, Reykjavik, Iceland, 2006.

BALTAR, A. M.; FONTANE, D. G. **A Generalized Multiobjective Particle Swarm Optimization Solver for Spreadsheet Models: Application to Water Quality**. Conference Proceedings AGU Hydrology Days, Colorado State University, Fort Collins, CO, USA, 2006.

BARTZ-BEIELSTEIN, T.; MEHNEN, J.; NAUJOKS, B.; SCHMITT, K.; ZIBOLD, D. **KEA - A Software Package for Development, Analysis and Application of Multiple Objective Evolutionary Algorithms**. University of Dortmund Reihe Computational Intelligence Collaborative Research Center 531, Dortmund, Germany, 2004.

BARTZ-BEIELSTEIN, T.; LIMBOURG, P.; PARSOPOULOS, K. E.; VRAHATIS, M. N.; MEHNEN, J.; SCHMITT, K. **Particle Swarm Optimizers for Pareto Optimization with Enhanced Archiving Techniques**. Proceedings of Congress on Evolutionary Computation (CEC'2003), v. 3, p. 1780–1787, Canberra, Australia, 2003.

BAUMGARTNER, U. ; MAGELE, C.; PREIS, K.; RENHART, W. **Particle Swarm Optimisation for Pareto Optimal Solutions in Electromagnetic Shape Design**. IEEE Proceedings on Science, Measurement and Technology, v. 151, n. 6, p. 499-502, 2004.

BAUMGARTNER, U.; MAGELE, C.; RENHART, W. **Pareto Optimality and Particle Swarm Optimization**. IEEE Transactions on Magnetics, v. 40, n. 2, p. 1172-1175, 2004.

BERSON, A.; SMITH, S. J. **Data Warehousing, Data Mining, and OLAP**, USA, McGraw-Hill, 1997.

BLICKLE, T. Tournament selection. In: BÄCK, T., FOGEL, D. B. E MICHALEWICZ, T. (eds), **Evolutionary Computation**, v. 1, 2000. Institute of Physics Publishing, p. 172-180, 2000.

BONISSONE, P. P. ; CHEN, Y.-T.; GOEBEL, K.; KHEDKAR, P. S. **Hybrid Soft Computing Systems: Industrial and Comercial Applications**, Proceedings of the IEEE, v. 87 n. 9, 1641-1667, 1999.

CAGNINA, L.; ESQUIVEL, S. **A Particle Swarm Optimizer for Multi-Objective Optimization**. Journal of Computer Science & Technology, v. 5, n. 4, p. 1-7, 2005.

CARRIJO, Ivaltemir Barros. **Extração de Regras Operacionais Ótimas de Sistemas de Distribuição de Água através de Algoritmos Genéticos**

Multiobjetivo e Aprendizado de Máquina. Tese de Doutorado, Universidade de São Paulo, SP, 2004.

CARVALHO, D. R.. **Um Método Híbrido Árvore de Decisão / Algoritmo Genético para Data Mining.** Tese de Doutorado, PPGIA, Pontifícia Universidade Católica do Paraná - PUCPR, Curitiba, PR, 2002.

CARVALHO, D. R.. **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, 2005.

CASTRO, R. E. **Otimização de Estruturas com Multi-Objetivos Via Algoritmos Genéticos.** Tese de Doutorado, COPPE, Universidade Federal do Rio de Janeiro, UFRJ, Rio de Janeiro, RJ, 2001.

CHOW, C.; TSUI, H. **Autonomous Agent Response Learning by a Multi-species Particle Swarm Optimization.** Proceedings of Congress on Evolutionary Computation (CEC'2004), v. 1, p. 778-785, Portland, Oregon, USA, June 2004.

COELLO, C. A. C.; LECHUGA, M. S. **MOPSO: A Proposal for Multiple Objective Particle Swarm Optimization.** Proceedings of Congress on Evolutionary Computation (CEC'2002), v. 2, p. 1051-1056, Piscataway, NJ, USA, 2002.

COELLO, C. A. C.; PULIDO, G. T.; LECHUGA, M. S. **Handling Multiple Objectives with Particle Swarm Optimization.** IEEE Transactions on Evolutionary Computation, v. 8, n. 3, p. 256-279, 2004.

COELHO, L. S.; MARIANI, V. C. **Concepção Híbrida de Otimização por Nuvem de Partículas Aplicada ao Problema de Weber,** INFOCOMP Journal of Computer Science, v. 4, n. 3, p. 24-31, 2005.

DEB, K.; PRATAP, A.; AGARWAL, S.; MEYARIVAN, T. **A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II**. IEEE Transactions on Evolutionary Computation, v. 6, n. 2, p. 182-197, 2002.

DEHURI, S.; JAGADEV, A. K.; GHOSH, A.; MALL, R. **Multi-objective Genetic Algorithm for Association Rule Mining Using a Homogeneous Dedicated Cluster of Workstations**. American Journal of Applied Sciences, v. 3, p. 2086-2095, 2006.

EBERHART, R. C.; SHI, Y., KENNEDY, J. **Swarm Intelligence**. Morgan Kaufmann Publishers, USA, 2001.

EBERHART, R. C.; SHI, Y., **Particle Swarm Optimization: Developments, Applications and Resources**. Proceedings of Congress on Evolutionary Computation, Seoul, Korea, v. 1, p. 81-86, 2001.

FAYYAD, U. M.; PIATETSKY-SHAPIRO, G.; SMYTH, P.; UTHURUSAMY, R. **Advances in Knowledge Discovery and Data Mining**. American Association for Artificial Intelligence, Menlo Park, CA, USA, 1996.

FIELDSEND, J. E.; SINGH, S. **A Multiobjective Algorithm Based Upon Particle Swarm Optimisation, an Efficient Data Structure and Turbulence**. In Proceedings of the 2002 U.K. Workshop on Computational Intelligence, p. 37–44, Birmingham, UK, 2002.

FRANK, E.; WITTEN, I. H. **Generating Accurate Rule Sets Without Global Optimization**. Proceedings of the 15th International Conference Machine Learning (ICML'98), p. 144-151, CA, USA, 1998.

FREITAS, A. A. **A Survey of Evolutionary Algorithms for Data Mining and Knowledge Discovery**, To appear in: A. Ghosh and S. Tsutsui. (Eds.) Advances in Evolutionary Computation, Springer-Verlag, p. 819-845, 2002.

FREITAS, A. A. **Data Mining and Knowledge Discovery with Evolutionary Algorithms**. Natural Computing Series, Springer-Verlag, Berlin, Germany, 2002.

GOLDBERG, D. E. **Genetic Algorithms in Search, Optimization and Machine Learning**. New York, USA, 1989.

GUO, Q.; YU, H.; XU, A. **A Hybrid PSO-GD Based Intelligent Method for Machine Diagnosis**. Digital Signal Processing, v. 16, n. 4, p. 402-418, 2006.

GURECK, E. L. **Data Mining – Aquisição de Conhecimento em Banco de Dados**, 2001. Monografia de conclusão de graduação de Tecnólogo em Processamento de Dados, Universidade Tuiuti do Paraná - UTP, Curitiba, PR, 2001.

HO, S. L.; SHIYOU, Y.; GUANGZHENG, N.; LO, E. W. C; WONG, H. C. **A Particle Swarm Optimization Based Method for Multiobjective Design Optimizations**. IEEE Transactions on Magnetics, v. 41, n. 5, p. 1756-1759, 2005.

HU, X.; EBERHART, R. **Multiobjective Optimization Using Dynamic Neighborhood Particle Swarm Optimization**. Proceedings of Congress on Evolutionary Computation (CEC'2002), v. 2, p. 1677-1681, Piscataway, NJ, USA, 2002.

HU, X.; EBERHART, R.; SHI, Y. **Particle Swarm with Extended Memory for Multiobjective Optimization**. Proceedings of the IEEE Swarm Intelligence Symposium, p. 193-197, Indianapolis, IN, USA, 2003.

ISHIDA, C. Y; POZO, A. T. R.; CARVALHO, A. B.; GOLDBARG, E. F. G; GOLDBARG, M. C. **Exploring Multi-Objective PSO and GRASP-PR for Rule Induction**. Proceedings of Evolutionary Computation in Combinatorial Optimization, Berlin/Heidelberg: Springer, v. 4972, p. 73-84, 2008.

JANSON, S.; MERKLE, D. **A New Multiobjective Particle Swarm Optimization Algorithm Using Clustering Applied to Automated Docking**. In María J. Blesa, Christian Blum, Andrea Roli, Michael Sampels, editors, Hybrid Metaheuristics, Second International Workshop, HM 2005, p. 128-142, Barcelona, Spain, 2005. Springer. Lecture Notes in Computer Science, v. 3636.

JOURDAN, L.; DHAENENS, C.; TALB, E. **A Genetic Algorithm for Feature Selection in Data-Mining for Genetics**. Proceedings of 4th Metaheuristics International Conference MIC'04, Porto, Portugal, p. 29-33, 2001.

KENNEDY, J.; EBERHART, R. C. **Particle Swarm Optimization**. Proceedings of the IEEE International Conference on Neural Networks IV, Perth, Australia, p. 1942-1948, 1995.

KENNEDY, J.; EBERHART, R. C. **A New Optimizer Using Particle Swarm Theory**. Micro Machine and Human Science, 1995. Proceedings of the Sixth International Symposium MHS '95, p. 39-43, Nagoya, Japan, 1995.

KENNEDY, J.; EBERHART, R. C.; SHI, Y. **Swarm Intelligence**. Morgan Kaufmann Publishers, San Francisco, CA, USA, 2001.

KHABZAOU, M.; DHAENENS, C.; TALBI, E. **Parallel Genetic Algorithms for multi-objective rule mining**. Proceeding of 6th Metaheuristics International Conference MIC2005, Vienna, Austria, p. 571-576, 2005.

KIRA, K.; RENDELL, L. A. **The Feature Selection Problem : Traditional Methods and a New Algorithm**. Proceedings of 10th Conference on Artificial Intelligence, p. 129-136, Menlo Park, CA, USA, 1992.

KOHAVI, R. **The Power of Decision Tables**. Proceedings of 8th Conference on Machine Learning, p.174-189, Springer-Verlag, London, UK, 1995.

KUKKONEN, S.; DEB, K. **Improved Pruning of Non-Dominated Solutions Based on Crowding Distance for Bi-Objective Optimization Problems.** Proceedings of IEEE Congress on Evolutionary Computation, p. 1179-1186, 2006.

LARSEN, O. **Construção de Atributos X-OF-N usando Algoritmos Genéticos.** Monografia de conclusão de graduação em Tecnologia em Processamento de Dados. Universidade Tuiuti do Paraná – UTP, Curitiba, PR, 2002.

LI, X. **A Non-dominated Sorting Particle Swarm Optimizer for Multiobjective Optimization.** In Erick Cantú-Paz et al., editor, Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2003), p. 37-48. Springer. Lecture Notes in Computer Science, v. 2723, 2003.

LI, X. **Better Spread and Convergence: Particle Swarm Multiobjective Optimization Using the Maximin Fitness Function.** Lecture Notes in Computer Science. Genetic and Evolutionary Computation – GECCO 2004, Springer, Berlin, Germany, 2004.

LOCATELLI, L. K.; NETO, W. A. **Seleção de Atributos a Partir de Algoritmos Genéticos.** Monografia de conclusão de graduação, Bacharel em Ciência da Computação. Universidade Tuiuti do Paraná – UTP, Curitiba, PR, 2004.

MARATH, A.; MAYER, S. **Data Mining para Avaliar Perfil de Candidatos ao Vestibular.** Monografia de conclusão de graduação em Tecnologia em Processamento de Dados – UTP, Curitiba, PR, 2001.

MINAEI-BIDGOLI, B.; PUNCH, W. F. **Using Genetic Algorithms for Data Mining Optimization in an Educational Web-based System.** Lecture Notes in Computer Science, Genetic and Evolutionary Computation – GECCO, Springer, Berlin, Germany, 2003.

MOORE, J.; CHAPMAN, R. **Application of Particle Swarm to Multiobjective Optimization**. Technical Report Department of Computer Science and Software Engineering, Auburn University, 1999.

MOSTAGHIM, S.; TEICH, J. **The Role of e-dominance in Multi Objective Particle Swarm Optimization Methods**. Proceedings of Congress on Evolutionary Computation (CEC'2003), v. 3, p. 1764-1771, Canberra, Australia, 2003a.

MOSTAGHIM, S.; TEICH, J. **Strategies for Finding Good Local Guides in Multi-objective Particle Swarm Optimization (MOPSO)**. Proceedings of the IEEE Swarm Intelligence Symposium, p. 26-33, Indianapolis, IN, USA, 2003b.

MOSTAGHIM, S.; TEICH, J. **Covering Pareto-optimal Fronts by Subswarms in Multi-objective Particle Swarm Optimization**. Proceedings of Congress on Evolutionary Computation (CEC'2004), v. 2, p. 1404-1411, Portland, OR, USA, 2004.

MURPHY, P. M.; AHA, D. W. **UCI Repository of Machine Learning databases**. [<http://www.ics.uci.edu/~mlearn/MLRepository.html>]. Irvine, CA: University of California, Department of information and Computer Science, 1994.

NOVAES, U. R. **Agrupamento de Dados Através de Algoritmos Swarm**. Tese de Doutorado, Programa de Pós-Graduação de Engenharia, Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, 2002.

PAPPA, G. L. **Seleção de Atributos Utilizando Algoritmos Genéticos Multiobjetivos**. Dissertação de Mestrado, Programa de Pós-Graduação em Informática Aplicada, Pontifícia Universidade Católica do Paraná, Curitiba, PR, 2002.

PARSOPOULOS, K. E.; VRAHATIS, M. N. **Recent Approaches to Global Optimization Problems Through Particle Swarm Optimization**, Natural Computing, v.1, p. 235-306, 2002a.

PARSOPOULOS, K. E.; VRAHATIS, M. N. **Particle Swarm Optimization Method in Multiobjective Problems**. Proceedings of the 2002 ACM Symposium on Applied Computing (SAC'2002), p. 603-607, Madrid, Spain, 2002b.

PARSOPOULOS, K. E.; TASOULIS, D. K.; VRAHATIS, M. N. **Multiobjective Optimization Using Parallel Vector Evaluated Particle Swarm Optimization**. In Proceedings of the IASTED International Conference on Artificial Intelligence and Applications (AIA 2004), v. 2, p. 823–828, Innsbruck, Austria, 2004.

PULIDO, G. T.; COELLO, C. A. C. **Using Clustering Techniques to Improve the Performance of a Particle Swarm Optimizer**. In Kalyanmoy Deb *et al.*, editor, Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2004), p. 225–237, Seattle, Washington, USA. Springer-Verlag, Lecture Notes in Computer Science, v. 3102, 2004.

PULIDO, G.T.; COELLO, C. A. C.; SANTANA-QUINTERO, L. V. **EMOPSO: A Multi-Objective Particle Swarm Optimizer with Emphasis on Efficiency**. Shigeru Obayashi, Kalyanmoy Deb, Carlo Poloni, Tomoyuki Hiroyasu and Tadahiko Murata (editors), Proceedings of 4th International Conference on Evolutionary Multi-Criterion Optimization, EMO 2007, p. 272--285, Springer. Lecture Notes in Computer Science, v. 4403, Matshushima, Japan, 2007.

QUILAN, J. R. **C4.5: Programs for Machine Learning**, Morgan Kaufmann Publisher, San Francisco, CA, USA, 1993.

RAQUEL, C. R.; NAVAL, P. P. **An Effective Use of Crowding Distance in Multiobjective Particle Swarm Optimization**. Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2005), p. 257-264, Washington, DC, USA, 2005.

RAY, T.; LIEW, K.M. **A Swarm Metaphor for Multiobjective Design Optimization.** Engineering Optimization, v. 34, n. 2, p.141-153, 2002.

REYES-SIERRA, M.; COELLO, C. A. C. **Improving PSO-based Multi-objective Optimization Using Crowding, Mutation and e-dominance.** Proceedings of Third International Conference on Evolutionary Multi-Criterion Optimization, EMO 2005., p. 505–519, Guanajuato, México, 2005. LNCS 3410, Springer-Verlag, 2005.

REYES-SIERRA, M.; COELLO, C. A. C. **Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art.** International Journal of Computational Intelligence Research. v. 2, n. 3, p. 287-308, 2006a.

REYES-SIERRA, M.; COELLO, C. A. C. **Dynamic Fitness Inheritance Proportion for Multi-objective Particle Swarm Optimization.** Proceedings of the 8th annual Conference on Genetic and evolutionary computation. Seattle, Washington, USA, 2006b.

REYES-SIERRA, M.; COELLO, C. A. C. **On-line Adaptation in Multi-Objective Particle Swarm Optimization.** Proceedings of Swarm Intelligence Symposium (SIS'06), p. 61-68, Indianapolis, IN, USA, 2006c.

REYES-SIERRA, M.; COELLO, C. A. C. **Improving PSO-based Multi-objective Optimization Using Crowding, Mutation and e-dominance.** In Third International Conference on Evolutionary Multi-Criterion Optimization, EMO 2005, p. 505-519, Guanajuato, México, 2005. LNCS 3410, Springer-Verlag, 2005.

REZENDE, S. O. **Sistemas Inteligentes – Fundamentos e Aplicações.** 1º. Edição. Editora Manole. Barueri, SP, 2003.

RUSSELL, C. E.; SHI, Y. **Comparison Between Genetic Algorithms and Particle Swarm Optimization.** In V. W. Porto, N. Saravanan, D. Waagen, A. E. Eibe, editors,

Proceedings of the 7th Annual Conference on Evolutionary Programming, p. 611-619, Springer-Verlag, Berlin, Germany, 1998.

SALAZAR-LECHUGA, M.; ROWE, J. E. **Particle Swarm Optimization and Fitness Sharing to solve Multi-Objective Optimization Problems.** Proceeding of IEEE Congress on Evolutionary Computation (CEC'2005), p. 1204-1211, v. 2, Edinburgh, Scotland, 2005.

SANTANA-QUINTERO, L. V.; RAMÍREZ-SANTIAGO, N.; COELLO, C. A. C.; LUQUE, J. M.;HERNÁNDEZ-DÍAZ, A. G. **A New Proposal for Multi-objective Optimization Using Differential Evolution and Rough Sets Theory.** Genetic And Evolutionary Computation Conference. Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation. Seattle, Washington, USA, p. 675-682, 2006.

SANTOS, R. D.; NAVAL, P. C. **Induction of Multiple Decision Trees using Multiobjective Particle Swarm Optimization.** Proceedings of 3rd International Conference on Fuzzy Systems and Knowledge Discovery, Xian, China, 2006.

SCHAFFER, J. D. **Multiple Objective Optimization with Vector Evaluated Genetic Algorithms.** Proceedings of International Conference on Genetic Algorithms and Their Applications, p. 93-100, 1985.

SHI, Y.; EBERHART, R. C. **A Modified Particle Swarm Optimizer.** Proceedings of the IEEE International Conference on Evolutionary Computation, San Diego, CA, USA, p. 69-73, 1998.

SHI, Y.; EBERHART, R. C. **Empirical Study of Particle Swarm Optimization.** Proceedings of the Congress on Evolutionary Computation, Washington, DC, USA, p. 1945-1950, 1999.

SOUSA, T.; SILVA, A.; NEVES, A. **A Particle Swarm Data Miner**. Proceedings of the 11th Portuguese Conference on Artificial Intelligence, Workshop on Artificial Life and Evolutionary Algorithms (ALEA), EPIA'03, Beja, Portugal, 2003.

SOUSA, T.; SILVA, A.; NEVES, A. **Swarm Optimization as a New Tool for Data Mining**. Proceedings of the 17th International Parallel & Distributed Processing Symposium - Workshop on Nature Inspired Distributed Computing (NIDISC) 2003, IPDPS, Nice, France, 2003.

SOUSA, T.; SILVA, A.; NEVES, A. **Particle Swarm based Data Mining Algorithms for Classification Tasks**. Parallel Computing, v. 30, n. 5-6, p. 767-783, 2004.

SOUSA, T.; SILVA, A.; NEVES, A. **Bio-Inspired Data Mining in Handbook of Bioinspired Algorithms and Applications**, Computer and Information Science, p. 469-489, Stephan Olariu and Albert Zomaya, 2006.

SRINIVASAN, D.; SEOW, T. H. **Particle Swarm Inspired Evolutionary Algorithm (PS-EA) for Multiobjective Optimization Problem**. Proceedings of Congress on Evolutionary Computation (CEC'2003), v. 3, p. 2292-2297, Canberra, Australia, 2003.

TAN, K. C.; YU, Q.; ANG, J. H. **A Dual-Objective Evolutionary Algorithm for Rules Extraction in Data Mining**, Computational Optimization and Applications, v. 34, n. 2, p. 273-294, 2006.

TAVARES, E.; SILVA, S. M. C. **Data Mining Baseado no Sistema Imunológico Artificial**. Monografia de conclusão de graduação em Tecnologia em Processamento de Dados. Universidade Tuiuti do Paraná - UTP, Curitiba, PR, 2002.

TICONA, W. G. C. **Aplicação de Algoritmos Genéticos Multi-Objetivo para Alinhamento de Seqüências Biológicas**. Dissertação de Mestrado, Instituto de Ciências Matemáticas e de Computação, ICMS-USP, 2003.

VILLALOBOS-ARIAS, M. A.; PULIDO, G. T.; COELLO, C. A. C. **A Proposal to Use Stripes to Maintain Diversity in a Multi-objective Particle Swarm Optimizer.** In Proceedings of the IEEE Swarm Intelligence Symposium, p. 22-29, Pasadena, CA, USA, 2005.

YUSIONG, J. P. T.; NAVAL, P. C. **Training Neural Networks Using Multiobjective Particle Swarm Optimization.** Proceedings of 2 International Conference Advances in Natural Computation, ICNC 2006, Xi'an, China, September 24-28, 2006. Proceedings, Part I. Lecture Notes in Computer Science 4221 Springer, Berlin, Germany, 2006.

ZANIN, E. **Ferramenta para Pré-processamento na Descoberta de Conhecimento em Bases de Dados.** 2002. Monografia de conclusão de graduação, Bacharel de Ciência da Computação. Universidade Tuiuti do Paraná – UTP, Curitiba, PR, 2002.

ZHAO, B.; CAO, Y. J. **Multiple Objective Particle Swarm Optimization Technique for Economic Load Dispatch.** Journal of Zhejiang University Science, v. 6, n. 5, p. 420-427, 2005.

ZHANG, L. B.; ZHOU, C. G.; LIU, X. H.; MA, Z. Q.; MA, M.; LIANG, Y. C. **Solving Multi Objective Optimization Problems Using Particle Swarm Optimization.** Proceedings of Evolutionary Computation (CEC '03), v. 4, p. 2400-2405, 2003.

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)